



**NTiC**

# Support de cours

**T**echnologies de

**l'I**nformatique et de la

**C**ommunication

Niveau : 4<sup>ème</sup> Sciences de l'informatique

*Rédigé par*

**JAZI Marwane**



# Sommaire

## Chapitre 1 : Outils de collaboration



## Chapitre 2 : Production électronique avancée

- Animations



- Pages Web Statiques



- Pages Web Dynamiques



## Chapitre 3 : Publication électronique



# Chapitre 1 :

## Outils de collaboration

### Objectif

- ❖ Exploiter des outils de collaboration



### I-Présentation :

Alors que les lycées et les centres de formation continuent d'investir dans des infrastructures liées aux domaines des nouvelles technologies, il devient essentiel d'exploiter ces dernières pour une formation efficace. Grâce aux outils de collaboration et de travail collaboratif, les organismes pédagogiques sont capables d'optimiser l'efficacité de leurs ressources informatiques tout en améliorant la qualité de la formation et de l'enseignement.

Les outils de collaboration offrent aux enseignants et aux formateurs la possibilité d'interagir facilement avec les élèves individuellement, en groupe ou par classe.

Ce chapitre vous montre comment exploiter les TIC à travers des applications dans un réseau local ou à travers l'utilisation des outils de collaborations à travers le réseau internet.

### II-Applications dans un réseau local :

Le partage des ressources est considéré comme le principal atout des réseaux informatiques qui peuvent être exploités dans les activités de formations.

Le problème qui se pose alors est de savoir comment créer des environnements d'apprentissage distribués permettant de réaliser à distance des activités de :

- Supervision de la classe et de suivi des travaux des élèves en temps réel
- Contrôle d'accès à certaines ressources (matérielle ou logicielles)
- Distribution et de collecte des documents
- Collaboration entre apprenants ou avec le formateur
- D'évaluation

#### Exemples :

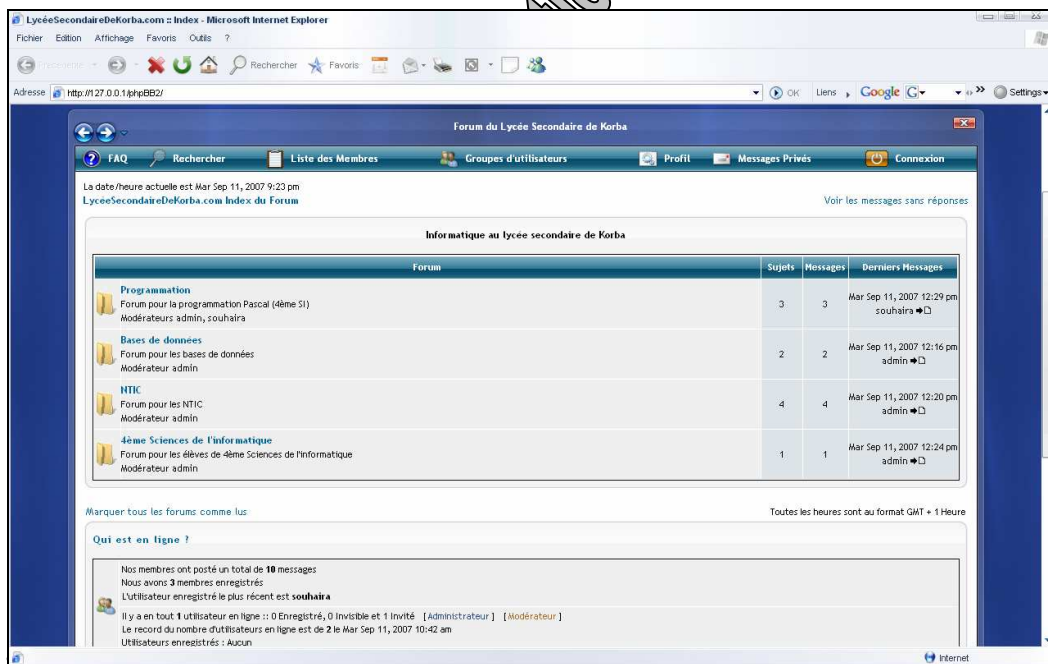
NetSupport School, Netop, VNC, SynchronEyes..



### III-Les forums :

#### 1-Définition :

Les forums se sont des *espaces de discussion* sur Internet, fonctionnant de manière asynchrone, comme une boîte aux lettres publique, dans laquelle chacun est libre de consulter les messages et d'y répondre. Un *forum* ou *newsgroup (groupe de discussion)* s'établit généralement autour d'un *sujet* donné. Dans certains cas, une personne appelée *modérateur* se charge, entre autres, d'animer les débats et d'éliminer les messages non conformes à la charte du forum.



## 2-Principe d'un forum :

- ✓ L'administrateur du forum, appelé aussi **modérateur**, propose un thème de discussion et veille à sa bonne exploitation.
- ✓ Les internautes intéressés par thème en cours de discussion, peuvent :
  - **consulter** le contenu du forum
  - **envoyer** des informations,
  - **poser** des questions
  - **répondre** aux questions
  - ...
- ✓ Pour la plupart des forums de discussion **l'inscription** est obligatoire pour y participer, même parfois attendre l'accord du modérateur ou l'administrateur.
- ✓ Suite au non respect des **règles d'utilisation** du forum, toutes participations peuvent être supprimées et même l'utilisateur peut être exclu.

### Remarque :

Le service associé aux forums est le **USENET**, le protocole utilisé en plus du **TCP/IP** est **NNTP**.

## IV-La visio-conférence :

### 1-Définition :

La visioconférence permet à des personnes éloignées de communiquer entre elles, en temps réel. Les systèmes utilisent des liaisons numériques de bas débit (64 Kbits/s à 2 Mbits/s) pour transmettre les images et le son entre deux lieux de réunion.

Si les dispositifs sont reliés à des systèmes informatiques, on peut aussi accompagner la conférence de séquences de travail coopératif (messagerie électronique en direct, partage d'applications informatiques, transfert de fichiers synchrone).





## 2-Avantages :

La visioconférence offre plusieurs avantages, parmi lesquels on peut citer :

- ✓ **Avantages matériels :**
  - Réduction des coûts : Diminution des frais de voyage et d'hébergement.
  - Meilleure productivité : Moins de temps passé dans les transports et hôtels et plus à travailler.
- ✓ **Avantages immatériels :**
  - Communication améliorée : Les personnes se rencontrent plus fréquemment.
  - Prise de décision plus rapide : Accès plus facile aux personnes clé.
  - Epanouissement des employés : Moins de stress, de temps et d'énergie perdus dans les voyages et hôtels.
  - Plus écologique : Pas d'émissions polluantes émanant des voitures et des avions.
- ✓ **Avantages stratégiques :**
  - Optimisation de la prise de décision : Les décisions sont non seulement plus rapides mais aussi prises avec l'accord de toutes les personnes impliquées.
  - Gestion de crise plus rapide : Les responsables peuvent se réunir, où qu'ils soient dans le monde.
  - Commercialisation plus rapide : Les nouveaux produits arrivent sur le marché plus vite grâce à l'accélération des prises de décision.

## 3-Modes d'utilisation :

Il y a trois principaux modes d'utilisation la visioconférence :

**Point à point :** La visioconférence en point à point, c'est lorsqu'un site se connecte avec un seul autre site.



- ✓ **La visioconférence en multipoint :** Les visioconférences multipoints permettent d'interconnecter au moins trois sites différents. Ce mode nécessite un pont qui gère le multipoint.



- ✓ **La visioconférence en mode broadcast :** Les visioconférences point à multipoint sont parfois appelées " mode broadcast " (mode diffusé). Ce réglage permet à un site privilégié de diffuser un message vers les autres sites, qui peuvent interagir et poser des questions. Si ces systèmes intègrent un pont, la communication est immédiate, sans réservation de pont multipoint. La réunion peut durer aussi longtemps que nécessaire et n'est pas restreinte à une tranche horaire. Les visioconférences point à multipoint sont, par exemple, idéales pour les stages d'entreprise ou lorsque le PDG désire s'adresser à certains employés.



#### 4-Différents types :

Il n'existe pas deux entreprises travaillant, pensant ou communiquant de la même manière. C'est pourquoi les fournisseurs de systèmes de visioconférence offrent un choix de solutions et de services de visioconférence. En combinant différents types de systèmes, une société peut se construire un parc de visioconférence sur mesure.

Les types de visioconférence définissent les protocoles et standards que les constructeurs et opérateurs doivent respecter afin de pouvoir interfonctionner.

##### a) *La visioconférence par réseau numérique à intégration de services (RNIS) :*

La visioconférence sur réseau RNIS (protocole H320) a été jusqu'à récemment la solution de choix pour les visioconférences dans le monde professionnel. Pour des raisons de coût et de flexibilité, une migration se produit vers le monde Internet.

##### b) *La visioconférence par Internet :*

Les systèmes de visioconférence sur Internet sont généralement prévus pour des personnes travaillant dans un bureau ou à leur domicile. Bien que ces systèmes supportent les visioconférences multipoint (c'est à dire des conférences liant plus de deux sites), la plupart d'entre eux sont utilisés pour des conférences point à point (c'est à dire une personne en conférence avec une autre).

Sur Internet (les protocoles H.323 et plus récemment SIP), la norme de visioconférence H323 a été conçue par l'UIT-T (monde des télécoms) alors que SIP a été conçu par l'IETF (monde de l'Internet). Jusqu'à aujourd'hui, H.323 était la norme majoritairement utilisée dans le domaine professionnel pour la visioconférence sur IP. Le protocole SIP est actuellement en train de prendre le relais, les opérateurs et fournisseurs de matériel ayant intégré l'offre.

##### c) *Les visioconférences hybrides :*

La visioconférence hybride est basé sur un micro-ordinateur équipé d'une carte visioconférence qui peut assurer l'interaction entre plusieurs sites de visioconférence et d'autres micro-ordinateurs équipés de microphones et webcams et cela en utilisant des réseaux numériques différents.

##### d) *La visioconférence par ATM :*

Réseau ATM (Asynchronous Transfer Mode) c'est un réseau large bande à intégration de services. C'est le réseau idéal pour la visioconférence. C'est un réseau de haut débit avec possibilité de réserver une certaine bande passante pour garantir le flux des données. La fluidité est garantie ainsi qu'une haute définition des images. La difficulté d'utilisation réside dans sa disponibilité internationale. Seul les universités et les centres de recherches ont accès à ce type de réseau commun.

##### e) *La visioconférence par satellite :*

La visioconférence par satellite permet d'assurer l'interaction entre plusieurs sites de visioconférence avec une qualité excellente, mais très coûteuse.

#### 5-Outils matériels et logiciels :

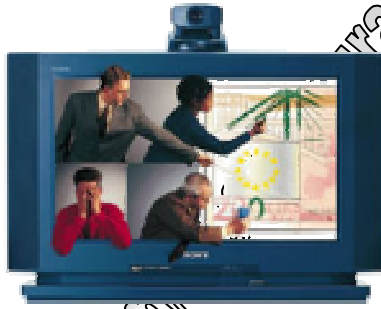
##### a) *Outils matériels :*

- *Cas d'une connexion utilisant les réseaux RNIS :*

Les systèmes de visioconférence comprennent tous les composants *hardware* et *software* (*Codec par exemple*) regroupés dans une *unité* que l'on pose sur une télévision. L'installation se fait très facilement (*Plug & Play*) et l'unité est portable. Vous n'avez besoin que d'un *téléviseur* et d'une *ligne RNIS* pour vos visioconférences. Les systèmes de visioconférence compacts



modernes peuvent être complétés par de nombreux périphériques tels qu'une *caméra document*, un *magnétoscope* ou encore un *PC*.



- *Cas d'une connexion par Internet :*  
Un kit de visioconférence sur PC comprend :
  - Une connexion Internet
  - Une Caméra ou webcam
  - Un casque ou un microphone et des haut-parleurs



#### b) *Outils logiciels :*

##### • *Les logiciels de visioconférence ouverts*

En dehors des logiciels associés à un service, il existe des logiciels de visioconférence "*ouverts*", compatibles avec les standards ouverts du marché (H.323 et SIP) et qui peuvent être utilisés dans le cadre de services tiers.

On peut citer dans l'ordre alphabétique :

- eConf (supporte H323, SIP, 3G-324M, MGCP)
- Ekiga (logiciel libre anciennement GnomeMeeting)
- eyeBeam
- Mirial, de Dylogic
- PVX de Polycom
- ReachVisio, de IP Media (supporte H323)
- Tandberg
- VT Phone, de Dilithium
- WengoPhone de Wengo
- XMeeting / OhPhoneX

##### ❖ *Les logiciels de visioconférence associés à un service*

Plusieurs logiciels de messagerie instantanée offrent la possibilité d'effectuer une communication audio/vidéo. De même, certaines sociétés offrent un service de visioconférence payant, associé à un logiciel spécifique. Généralement, ces logiciels ne peuvent communiquer qu'avec les utilisateurs du même service et ils ne sont pas ouverts sur l'extérieur.

On peut citer dans l'ordre alphabétique :

- AOL Instant Messenger
- Best Of Chat
- Camfrog
- Gizmo
- iChat
- meeting3D, Réunion en ligne avec la vidéo
- MSN Messenger
- Skype
- Yahoo! Messenger

**Chapitre 2 :*****Production électronique avancée*****Objectifs**

- ❖ Réaliser des animations
- ❖ Créer des sites Web statiques et dynamiques
- ❖ Échanger des données à distance à travers une page Web

**Partie A : Animations****Partie B : Pages Web Statiques****Partie C : Pages Web Dynamiques**



## Chapitre 2 : Partie A :

# Animations



macromedia  
**FLASH**<sup>TM</sup> MX  
2004

### I-Introduction :

Flash est logiciel hybride qui allie de manière exceptionnelle les avantages des **graphiques vectoriels** et ceux des **graphiques bitmap** :

- Un **graphique vectoriel** décrit les objets graphiques et les formes comme un assemblage de droites et de courbes dans un système de coordonnées bidimensionnelles.
- Un **graphique bitmap** est défini par une mosaïque de points d'image (pixels), d'une taille particulière.

Les graphiques réalisés dans flash sont surtout de type vectoriels. Ils se composent de contours externes remplis avec une couleur.

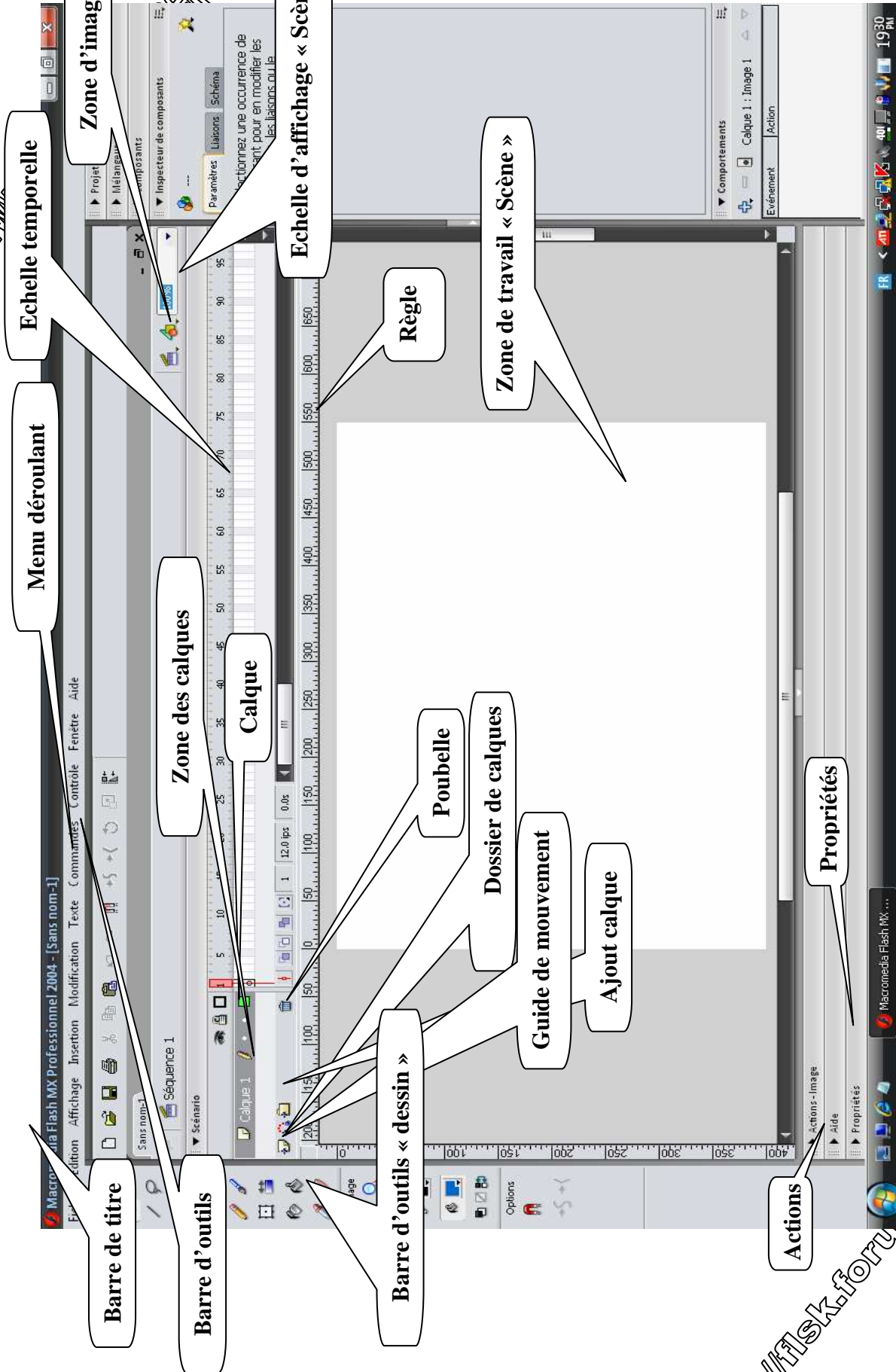
Les avantages sont multiples : faciles à créer, contours et couleurs aisément modifiables, redimensionnement aisé sans perte de qualité.

Le logiciel Flash se caractérise donc par le **taux de compression élevé** de ses fichiers et par une excellente qualité de l'affichage à l'écran.

Il met aussi à la disposition des créateurs une vaste palette de possibilités **de développement interactif**.

II-Présentation de l'interface de Flash :

*Information*



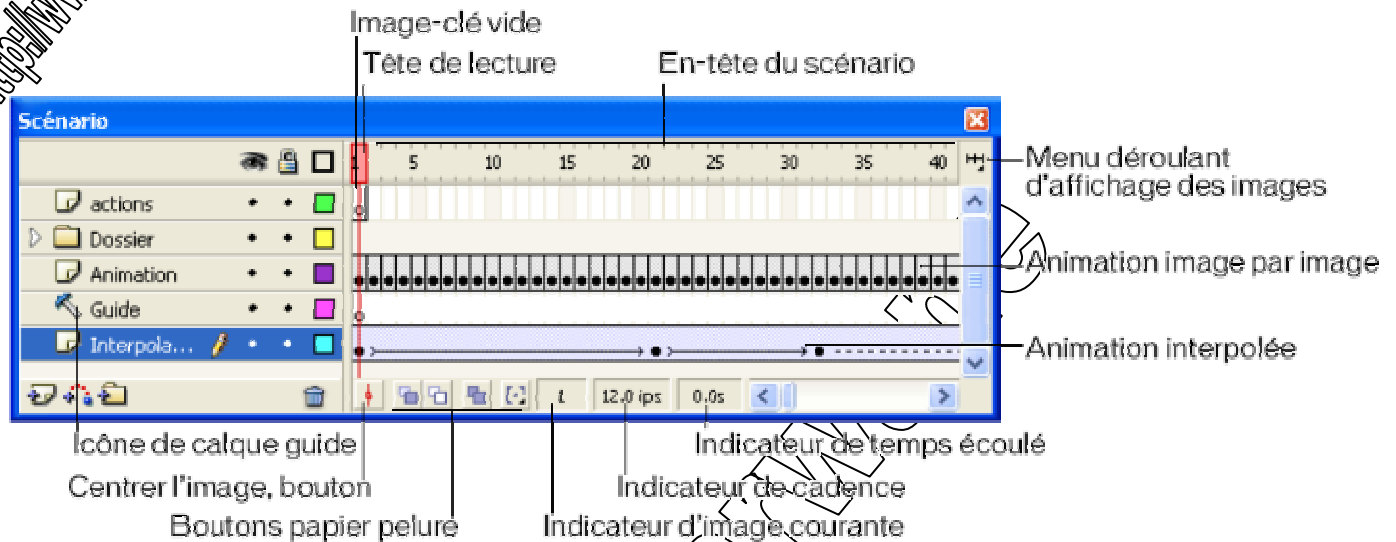
*http://fsk.forumgo.com*

### 1-La scène :

Comme les films, les animations Flash divisent les périodes de temps en images. La scène est la zone dans laquelle vous composez le contenu des images individuelles de l'animation, soit en y dessinant directement les illustrations, soit en organisant les illustrations importées.

### 2-Le scénario :

Le scénario organise et contrôle le contenu d'un document au fil du temps dans des calques et des images. Tout comme les films, les documents Flash divisent les périodes de temps en images. Les calques sont semblables à des bandes de film empilées les unes sur les autres, chacune contenant une image différente qui apparaît sur la scène. Les principaux composants du scénario sont les calques, les images et la tête de lecture.



## III-Les éléments d'affichage :

### 1-La règle :

Lorsque les règles sont affichées, elles apparaissent en haut de l'animation ou sur les côtés. Vous pouvez sélectionner l'unité de mesure utilisée dans les règles. Lorsque vous déplacez un élément sur la scène alors que les règles sont affichées, des lignes indiquant les dimensions des éléments s'affichent sur les règles.

**Pour afficher ou masquer les règles :**

- Sélectionnez **Affichage > Règles**.

### 2-La grille :

Lorsque les grilles sont affichées dans une animation, elles apparaissent sous la forme de lignes placées derrière les illustrations dans chacune des scènes. Vous pouvez ajuster des objets sur la grille, de même que modifier la taille de la grille ou la couleur de ses lignes.

### 3-Les panneaux :

Pour visualiser, organiser et modifier des éléments dans une animation Flash, vous pouvez utiliser des panneaux flottants contenant les commandes et les options relatives à chaque type d'élément. Les panneaux vous permettent de modifier les symboles, les occurrences, les couleurs, le type, les images et d'autres éléments.

Vous pouvez utiliser les panneaux pour personnaliser l'interface de Flash, en affichant les panneaux dont vous avez besoin pour une tâche spécifique et masquer les autres panneaux.

**Pour ouvrir un panneau :**

- Choisissez **Fenêtre > Panneaux** et sélectionnez le **panneau de votre choix** dans la liste.

## IV-Les éléments d'une animation Flash :

### 1-Les propriétés du document :

À chaque ouverture, Flash crée un nouveau fichier. Vous pouvez créer des animations supplémentaires pendant que vous travaillez. Utilisez la boîte de dialogue Propriétés de l'animation pour définir la taille, la cadence, la couleur d'arrière-plan et les autres propriétés d'une nouvelle animation.

#### *Pour créer une animation et définir ses propriétés :*

- 1- Cliquez sur **Fichier > Nouveau**.
- 2- Cliquez sur **Modification > Animation**. La boîte de dialogue Propriétés de l'animation s'affiche.
- 3- Dans la zone **Cadence**, indiquez le **nombre d'images** qui doivent être affichées par seconde. Pour la majorité des effets animés affichés sur ordinateur, notamment les animations lues à partir d'un site web, une cadence de 8 ips (images par seconde) à 12 ips est suffisante (12 ips est la cadence par défaut).
- 4- Dans la zone **Dimensions**, choisissez l'une des options suivantes :
  - Pour spécifier la taille de la scène en pixels, entrez des valeurs dans les champs Largeur et Hauteur. La taille par défaut de l'animation est 550 pixels par 400 pixels. La taille minimale est 18 pixels par 18 pixels ; la taille maximale est 2880 pixels par 2880 pixels.
  - Pour définir la taille de la scène de telle façon qu'il y ait un espace égal autour du contenu, cliquez sur Ajuster selon : Contenu. Pour réduire la taille de l'animation, alignez tous les éléments dans le coin supérieur gauche de la scène avant d'utiliser l'option Ajuster selon : Contenu.
  - Pour définir la taille de la scène sur la zone d'impression maximale, cliquez sur Ajuster selon : Imprimante. Cette zone est déterminée par la taille du papier moins la marge actuelle sélectionnée dans la zone Marges de la boîte de dialogue Mise en page.
- 5- Pour définir la **couleur d'arrière-plan** de votre animation, choisissez une couleur dans le nuancier Couleur d'arrière-plan.
- 6- Sélectionnez **l'unité de mesure** dans le menu déroulant Unités de la règle que vous pouvez afficher en haut et sur le côté de la fenêtre de l'application (l'option Unités de la règle détermine également les unités utilisées dans le panneau Info).
- 7- Cliquez sur OK.

### 2-Les séquences :

Pour organiser une animation de façon thématique, vous pouvez utiliser des séquences. Par exemple, vous pouvez utiliser des séquences distinctes pour une introduction, un message de chargement ou un générique. Lorsque vous publiez une animation Flash qui contient plus d'une séquence, les séquences du fichier SWF sont lues dans l'ordre dans lequel elles sont répertoriées dans le panneau séquences dans le fichier FLA. Les images du fichier SWF sont numérotées de manière consécutive d'une séquence à une autre. Par exemple, si une animation contient deux séquences de 10 images chacune, les images de la séquence 2 sont numérotées de 11 à 20.

Vous pouvez ajouter, supprimer, dupliquer, renommer et modifier l'ordre des séquences. Pour arrêter ou suspendre une animation après chaque scène, ou pour laisser l'utilisateur naviguer dans l'animation de façon non linéaire, vous devez utiliser des actions.

#### *Pour afficher le panneau Scène :*

Choisissez **Fenêtre > Panneaux > Séquence**.

#### *Pour afficher une scène particulière :*

Sélectionnez **Affichage > Aller à**, puis choisissez le nom de la séquence dans le sous-menu.

#### *Pour ajouter un outil, effectuez l'une des opérations suivantes :*

- Cliquez sur le bouton Ajouter dans le panneau séquence.



- Choisissez Insertion > Séquence.

*Pour supprimer une séquence, effectuez l'une des opérations suivantes :*

- Cliquez sur le bouton **Supprimer** dans le **panneau séquence**.
- Ouvrez la séquence à supprimer, puis choisissez Insertion > Supprimer la séquence.

*Pour changer le nom d'une séquence:*

Double-cliquez sur le **nom de la séquence** dans le **panneau séquence** et entrez le nouveau nom.

*Pour dupliquer une séquences:*

Cliquez sur le bouton Dupliquer dans le panneau séquence.

*Pour modifier l'ordre d'une séquence dans l'animation :*

Faites glisser le nom de la séquence à un autre emplacement dans le panneau séquence.

### 3-Les calques :

Les calques s'apparentent à des feuilles transparentes en acétate empilées les unes sur les autres. Quand vous créez une animation Flash, celle-ci contient un calque. Vous pouvez y ajouter d'autres calques de façon à organiser le contenu artistique et les autres éléments de votre animation. Vous pouvez dessiner et modifier des objets sur un calque sans toucher aux objets des autres calques. Lorsque rien ne repose sur le calque, celui-ci laisse apparaître les calques situés en dessous.

Le nombre de calques que vous pouvez créer n'a de limite que la mémoire de votre ordinateur, et les calques n'augmentent pas la taille de fichier de votre animation publiée. Vous pouvez masquer ou verrouiller des calques ou afficher le contenu d'un calque sous forme de contours. Vous pouvez aussi modifier l'ordre des calques.

En outre, vous pouvez utiliser des calques de guide spéciaux pour faciliter la réalisation de dessins et les modifications, et masquer les calques pour mieux créer des effets spéciaux. Il est conseillé d'utiliser des calques distincts pour les fichiers audio, les actions, les étiquettes d'image et les commentaires d'image. Cela vous permet de retrouver plus rapidement ces éléments lorsque vous devez les modifier.

*Pour créer un calque :*

Effectuez l'une des opérations suivantes :

- Cliquez sur le **bouton Insérer** un calque, en bas du scénario.
- Choisissez **Insertion > Calque**.
- Cliquez avec le bouton droit de la souris sur un nom de calque dans le scénario puis choisissez Insérer un calque dans le menu contextuel.

### 4-Les images :

Dans le scénario, vous travaillez avec des **images** et des **images-clés** en les plaçant dans l'ordre correspondant à celui dans lequel vous souhaitez que les objets apparaissent dans les images. Vous pouvez changer la longueur d'un effet animé interpolé en faisant glisser une image-clé dans le scénario.

Une **image-clé** est une image dans laquelle vous définissez les changements des propriétés d'un objet pour une animation, ou dans laquelle vous incluez du code ActionScript afin de contrôler un aspect du document. Flash peut interpoler, ou remplir automatiquement, les images situées entre les images-clés que vous définissez, de manière à produire des animations fluides. Les images-clés facilitent la création des animations car elles évitent d'avoir à tracer chacune de leurs images. Vous pouvez modifier aisément la longueur d'un effet animé interpolé en faisant glisser une image-clé dans le scénario.

### 5-Utilisation d'images dans le scénario :

Dans le scénario, vous travaillez avec des images et des images-clés. Une image-clé est une image dans laquelle vous définissez une modification d'un effet animé, ou incluez des actions d'image permettant de modifier une animation. Les images-clés sont un élément important d'un effet animé interpolé. Vous pouvez modifier la longueur d'un effet animé interpolé en faisant glisser une image-clé dans le scénario.

Vous pouvez apporter les modifications suivantes sur les images et les images-clés :

- Insérer, sélectionner, supprimer et déplacer des images ou des images-clés.
- Faire glisser des images et des images-clés vers un nouvel emplacement du même calque ou d'un calque différent.
- Copier et coller des images et des images-clés.
- Convertir des images-clés en images.
- Faire glisser un élément depuis la fenêtre Bibliothèque vers la scène de manière à l'ajouter à l'image-clé actuelle.

### V-Le dessin vectoriel :

Les ordinateurs affichent les graphiques au format vectoriel ou bitmap. Il est important de bien comprendre la différence entre ces deux formats pour les utiliser de manière plus efficace. Flash vous permet de créer et d'animer des graphiques vectoriels compacts. Il vous permet également d'importer et de manipuler des graphiques bitmap et vectoriels créés dans d'autres applications.

#### ➤ Graphiques vectoriels

Les graphiques vectoriels décrivent les images à l'aide de lignes et de courbes, appelées *vecteurs*, qui intègrent également les propriétés de couleur et de position.

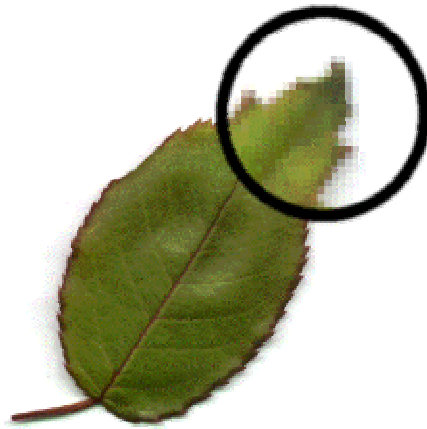
Par exemple, l'image d'une feuille est décrite par les points par lesquels passent des lignes, qui créent ainsi le contour de la feuille. La couleur de la feuille est définie par la couleur du contour et la couleur de la zone qu'il définit.



Lorsque vous éditez un graphique vectoriel, vous modifiez les propriétés des lignes et des courbes qui en décrivent la forme. Vous pouvez déplacer, redimensionner, remodeler et modifier la couleur d'un graphique vectoriel sans modifier la qualité de son apparence. Les graphiques vectoriels ne dépendent pas de la résolution, c'est-à-dire qu'ils peuvent être affichés sur des écrans de résolution différente sans perdre leur qualité.

#### ➤ Graphiques bitmap

Les graphiques bitmap sont les images formées par des points de couleur, appelés *pixels*, disposés suivant une grille. Par exemple, l'image d'une feuille est décrite par l'emplacement et la valeur de couleur spécifiques de chaque pixel de la grille, créant une image assez semblable à une mosaïque.



Lorsque vous éditez un graphique bitmap, vous modifiez des pixels et non des lignes et des courbes. Les graphiques bitmap dépendent de la résolution car les données décrivant l'image sont définies pour une trame de dimension particulière.

L'édition d'un graphique bitmap peut modifier la qualité de son aspect. C'est ainsi que le redimensionnement d'un graphique bitmap peut donner un effet d'escalier aux bords de l'image lorsque les pixels sont redistribués dans la grille. L'affichage d'un graphique bitmap sur un écran d'une résolution inférieure à celle de l'image dégrade également la qualité de son apparence.

## VI-Les formes :

### 1-Dessiner avec l'outil Crayon :

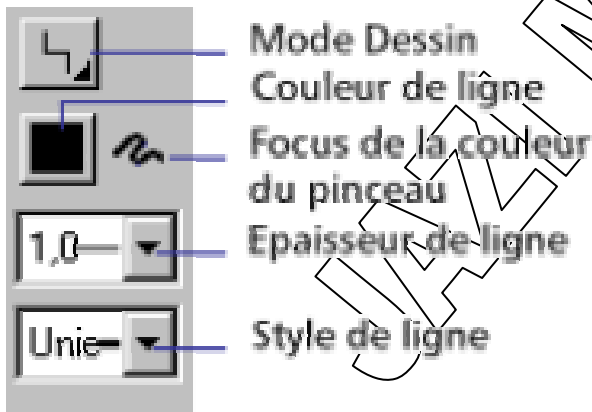
Flash permet de dessiner des lignes et des formes avec l'outil Crayon. Après avoir tracé les lignes, Flash les redresse ou les lisse. La valeur du redressement ou du lissage dépend du mode de dessin sélectionné. Flash segmente également les lignes à leurs angles saillants.



Pour dessiner avec l'outil Crayon :

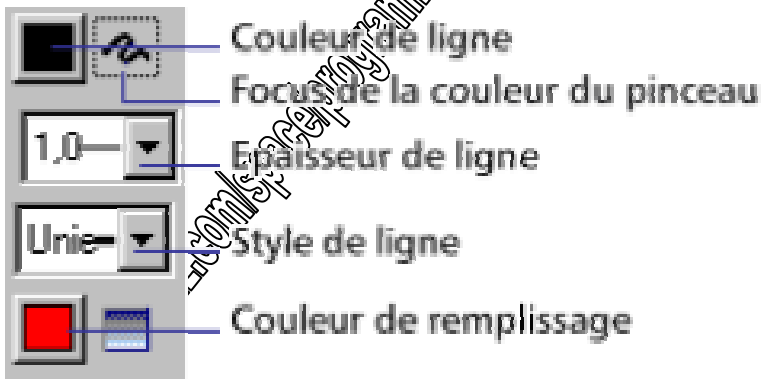
1

Sélectionnez l'outil Crayon.



### 2-Définir les attributs de remplissage et de trait

Les outils Crayon, Ligne, Rectangle, Ovale, Pinceau, Remplissage et Encier disposent des mêmes modificateurs pour les attributs de remplissage et de trait. Lorsque vous passez d'un outil à l'autre, les attributs de remplissage et de trait ne changent pas. Les lignes ne peuvent être remplies que si vous les convertissez en remplissages.



### Pour définir les attributs de remplissage et de trait :

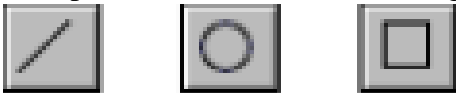
1 Sélectionnez un outil de dessin ou de peinture.

2 Pour définir la couleur de trait ou de remplissage, effectuez l'une des opérations suivantes :

- ▶ Positionnez le pointeur sur le modificateur de couleur de ligne ou de remplissage et faites glisser pour choisir une couleur de la palette.
  - ▶ Choisissez Fenêtre > Couleurs et sélectionnez une couleur dans l'onglet Unies. Si vous utilisez les outils Ovale ou Rectangle, cliquez sur le focus de couleur, situé près des modificateurs de couleur de ligne ou de remplissage de la boîte à outils, pour définir l'attribut approprié.
- 3 Pour définir l'épaisseur et le style de ligne, choisissez les options dans les modificateurs d'épaisseur et de style de ligne

### 3-Tracer des lignes droites, des ovales et des rectangles.

Vous pouvez utiliser les outils Ligne, Ovale et Rectangle pour tracer facilement ces formes. Les outils Ovale et Rectangle vous permettent de tracer et de remplir la forme lors de son dessin. L'outil Rectangle vous permet de dessiner des rectangles avec des coins droits ou arrondis.



### Pour dessiner une ligne droite, un ovale ou un rectangle :

- 1 Sélectionnez l'outil Ligne, Ovale ou Rectangle.
- 2 Choisissez les attributs de trait et de remplissage en suivant la procédure décrite dans. Vous ne pouvez pas définir d'attributs de remplissage pour l'outil Ligne.
- 3 Pour l'outil Rectangle, indiquez des coins droits en cliquant sur le modificateur Rayon de rectangle arrondi et entrez une valeur pour définir son rayon. La valeur zéro crée des coins droits.
- 4 Faites glisser la souris sur la Scène. Si vous utilisez l'outil Rectangle, appuyez sur les touches flèche haut et bas tout en faisant glisser le pointeur pour ajuster le rayon des coins arrondis.

Pour les outils Ovale et Rectangle, faites glisser en appuyant sur la touche Maj. pour tracer des formes en cercles et carrés.

Pour l'outil Ligne, faites glisser en appuyant sur la touche Maj. pour tracer des lignes formant un angle multiple de 45°.

Pour l'outil Ligne, faites glisser en appuyant sur la touche Maj. pour tracer des lignes formant un angle multiple de 45°.



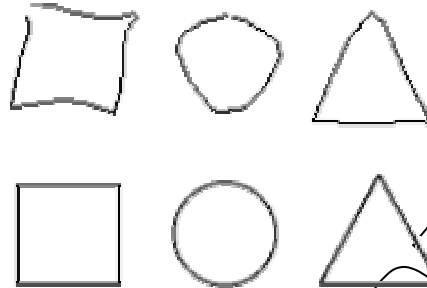
**4-Redresser et lisser les lignes :**

Vous pouvez modifier des lignes et des contours de forme en les redressant ou en les lissant.

**Remarque :** Choisissez *Fichier > Assistant pour ajuster le degré du lissage et du redressement automatiques.*

Utilisez le modificateur Redresser de l'outil Flèche ou choisissez *Modifier > Courbes > Redresser* pour effectuer de petits ajustements de redressement aux lignes et aux courbes que vous avez déjà dessinées. Le redressement n'a pas d'effet sur les segments déjà rectilignes.

Vous pouvez également utiliser la technique de redressement pour que Flash reconnaisse les formes. Flash trace les cercles, ovales, rectangles, carrés, triangles et arcs parfaitement géométriques avec les options de reconnaissance de forme désactivées. Les formes qui se touchent, et qui sont donc connectées à d'autres segments, ne sont pas reconnues.



La reconnaissance de forme transforme les formes du haut en formes du bas.

Le lissage lisse les courbes et réduit les accidents ou autres variations dans la direction générale de la courbe. Le lissage réduit également le nombre de segments d'une courbe. Le lissage est toutefois relatif, et n'a pas d'effets sur les segments droits. Le lissage est particulièrement utile lorsque vous avez des problèmes pour modifier de nouveau la forme d'un certain nombre de très petits segments de ligne. La sélection de tous les segments et leur lissage réduit le nombre de segments, produisant une courbe plus douce dont il est plus facile de modifier la forme.

Une application répétée de lissage ou de redressement lisse ou redresse chaque segment, en fonction de la courbure ou de la rectitude de chaque segment à l'origine.

**Pour lisser la courbe de chaque contour de remplissage sélectionné ou de ligne courbe :**

Sélectionnez l'outil Flèche et cliquez sur le modificateur Lisser, ou choisissez *Modifier > Courbes > Lisser.*

**Pour effectuer des ajustements de redressement plus faibles sur chaque contour de remplissage ou sur une ligne courbe sélectionnés :**

Sélectionnez l'outil Flèche et cliquez sur le modificateur Redresser, ou choisissez *Modifier > Courbes > Redresser.*

**Pour utiliser la reconnaissance de forme :**

Sélectionnez l'outil Flèche et cliquez sur le modificateur Redresser, ou choisissez *Modifier > Courbes > Redresser.*

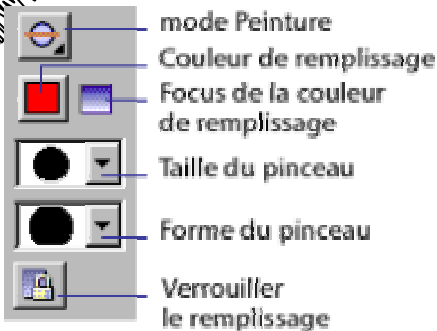
**5-Peindre avec l'outil Pinceau :**

L'outil Pinceau trace des traits qui évoquent ceux d'un pinceau. L'outil Pinceau vous permet de créer des effets spéciaux, notamment des effets de calligraphie et de peinture avec une image bitmap importée. La plupart des tablettes sensibles à la pression vous permettent de faire varier l'épaisseur du trait de pinceau en modifiant la pression du stylet. Pour plus d'informations sur l'utilisation d'une image bitmap importée pour le remplissage d'un trait de pinceau.



### Pour peindre avec l'outil Pinceau :

1 Sélectionnez l'outil Pinceau.



2 Choisissez un mode de peinture :

Image originale, Peint normalement, Peint derrière, Remplissage, Peint la sélection, et Peint à l'intérieur

- ▶ Le mode Peint normalement peint sur les lignes et les remplissages du même calque.
  - ▶ Le mode Peint derrière peint les parties vides de la Scène, sans toucher aux lignes et aux remplissages.
  - ▶ Le mode Remplissage peint les zones remplies et les zones vides, sans toucher aux lignes.
  - ▶ Le mode Peint la sélection peint le remplissage sélectionné.
  - ▶ Le mode Peint à l'intérieur peint le remplissage où commence votre coup de pinceau mais ne peint jamais les lignes. Ceci fonctionne un peu comme un livre à colorier, où vous ne pouvez jamais peindre en dehors des lignes. Si vous commencez à peindre dans une zone vide, ce coup de pinceau n'affectera pas les zones déjà remplies.
- 3 Choisissez une taille de pinceau, une forme de pinceau et une couleur de peinture dans les modificateurs de l'outil Pinceau.
- 4 Si une tablette sensible à la pression est connectée à votre ordinateur, vous pouvez sélectionner le Modificateur de pression pour faire varier l'épaisseur de vos coups de pinceau en fonction de la pression appliquée à votre stylet.
- 5 Faites glisser la souris sur la scène. Faites glisser le pointeur en appuyant sur la touche Maj. pour tracer des coups de pinceau horizontaux et verticaux.

**6-Utiliser l'outil Remplissage :**

L'outil Remplissage remplit de couleur les zones fermées. Il remplit les zones vides et modifie la couleur de la peinture existante. Vous pouvez peindre avec des couleurs unies, dégradées et bitmaps. Vous pouvez également utiliser l'outil Remplissage pour ajuster la dimension, la direction et le centre des remplissages dégradés et bitmaps. Pour plus d'informations sur l'utilisation d'une image de bitmap pour remplir une forme

**Pour utiliser l'outil Remplissage pour remplir une zone :**

- 1 Sélectionnez l'outil Remplissage.
- 2 Choisissez la couleur de remplissage.
- 3 Choisissez une option de taille de l'espace.
  - ▶ Choisissez Ne ferme pas les espaces si vous souhaitez fermer vous-même les espaces. Cela peut être plus rapide pour les dessins complexes.
  - ▶ Choisissez une option de fermeture pour que Flash ferme automatiquement les espaces.
- 4 Cliquez sur une forme ou une zone fermée.

**7-Utiliser l'outil Encrier :**

Vous pouvez tracer des lignes et des formes avec des couleurs unies uniquement, et non pas des dégradés ou des bitmaps. Les attributs de trait comprennent également l'épaisseur de ligne et le style de ligne.

**Pour utiliser l'outil Encrier :**

- 1 Sélectionnez l'outil Encrier.
- 2 Choisissez la couleur de ligne, le style de ligne et l'épaisseur de ligne dans les modificateurs de l'outil.
- 3 Cliquez sur une ligne ou une forme de la Scène.

**8-Utiliser l'outil Pipette :**

Utilisez l'outil Pipette pour copier les attributs de remplissage et de trait d'une forme ou d'une ligne et les appliquer immédiatement à une autre forme ou ligne. L'outil Pipette vous permet également d'échantillonner l'image d'un bitmap pour l'utiliser comme peinture.

Lorsque, avec l'outil Pipette, vous cliquez sur une zone avec la touche Maj. enfoncée, les attributs de remplissage et de trait échantillonnés sont appliqués aux modificateurs pour tous les outils de dessin.

**Pour utiliser l'outil Pipette pour copier et appliquer des attributs de trait ou de remplissage :**

- 1 Sélectionnez l'outil Pipette et cliquez sur une ligne ou une forme.

Lorsque vous cliquez sur une ligne, l'outil se transforme automatiquement en outil Encrier. Lorsque vous cliquez sur un remplissage, l'outil se transforme automatiquement en outil Remplissage et le modificateur Verrouiller le remplissage est activé.

- 2 Cliquez sur une autre ligne ou forme.

**9-Remplir et délimiter des formes :**

Remplir une forme signifie la peindre dans une couleur. Délimiter une forme consiste à tracer son contour dans une couleur. Délimiter une ligne revient à la peindre dans une couleur.

Utilisez l'outil Remplissage pour remplir les formes. Utilisez l'outil Encrier pour délimiter des lignes et des formes. Utilisez l'outil Rasoir pour prélever un échantillon sur une couleur de remplissage ou sur des attributs de trait d'une forme. Appliquez-le immédiatement à une autre forme grâce aux outils Remplissage ou Encrier.

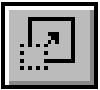
### 10-Redimensionner des objets :

Le redimensionnement d'un objet agrandit ou réduit l'objet horizontalement, verticalement, ou horizontalement et verticalement. Vous pouvez redimensionner un objet en faisant glisser ses poignées ou en saisissant des valeurs dans l'inspecteur Transformer. Le redimensionnement d'une occurrence, d'un groupe ou d'un bloc de caractères s'effectue en relation avec son point d'enregistrement.

#### Pour redimensionner un objet par glissement :

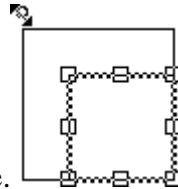
1 Sélectionnez un objet.

2 Sélectionnez l'outil Flèche et cliquez sur le modificateur Redimensionner.



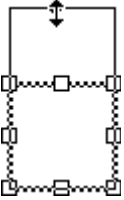
3 Effectuez les opérations suivantes :

- Pour redimensionner l'objet à la fois horizontalement et verticalement, faites glisser l'une des poignées d'angle.



Les proportions se modifient à mesure.

- Pour redimensionner l'objet horizontalement ou verticalement, faites glisser une poignée centrale.



**Remarque :** Lorsque vous augmentez la dimension de plusieurs éléments, ceux qui se trouvent près des bords du cadre de délimitation peuvent se trouver en dehors de la scène. Si cela se produit, choisissez Afficher > Zone de travail pour voir les éléments qui sont au-delà des bords de la scène.

#### Pour redimensionner un objet à l'aide de l'inspecteur Transformer :

- 1 Sélectionnez un objet.
- 2 Choisissez Fenêtre > Inspecteurs > Transformer.
- 3 Saisissez une valeur d'échelle entre 10 et 1000, et cliquez sur Appliquer

### 11-Rotation d'objets :

La rotation d'un objet le fait tourner autour de son point d'enregistrement. Il s'agit par défaut du centre de l'objet, que vous pouvez cependant déplacer. Vous pouvez faire pivoter un objet en le faisant glisser ou en spécifiant un angle dans l'inspecteur Transformer.

#### Pour faire pivoter un objet par glissement :

- 1 Sélectionnez un objet.
- 2 Sélectionnez l'outil Flèche et cliquez sur le modificateur Pivoter.





3 Faites glisser l'une des poignées d'angle.



#### Pour faire pivoter un objet par une valeur :

4 Sélectionnez un objet.

Effectuez l'une des opérations suivantes :

- ▶ Ouvrez l'inspecteur Transformer (Fenêtre > Inspecteurs > Transformer), indiquez un angle de rotation et cliquez sur Appliquer. Les valeurs négatives font pivoter l'objet dans le sens inverse des aiguilles d'une montre, et les valeurs positives dans le sens des aiguilles d'une montre.
- ▶ Pour faire pivoter la sélection de 90° dans le sens des aiguilles d'une montre, choisissez Modifier > Transformer > Pivoter sur la droite.
- ▶ Pour faire pivoter la sélection de 90° dans le sens inverse des aiguilles d'une montre, choisissez Modifier > Transformer > Pivoter sur la gauche

### VII-Les groupes :

Pour manipuler des éléments en tant qu'objet unique, vous devez les grouper. Par exemple, après avoir créé un dessin tel qu'un arbre ou une fleur, vous pouvez grouper les éléments du dessin pour pouvoir sélectionner et déplacer facilement le dessin dans son ensemble. Vous pouvez modifier des groupes sans les dissocier. Vous pouvez aussi sélectionner un objet isolé d'un groupe en vue d'une modification, sans dissocier les objets.

#### Pour créer un groupe :

- 1- Sélectionnez les objets de la scène que vous souhaitez grouper. Vous pouvez sélectionner des formes, d'autres groupes, des symboles, du texte, etc.
- 2- Choisissez **Modification > Grouper** ou appuyez sur Ctrl+G.

#### Pour dissocier des objets

Choisissez **Modification > Dissocier**

#### Pour modifier un groupe ou un objet à l'intérieur d'un groupe :

1. Avec le groupe sélectionné, choisissez **Édition > Modifier** la sélection ou double-cliquez sur le groupe avec l'outil Flèche.

Tous les éléments de la page qui ne font pas partie du groupe sont estompés, ce qui signifie qu'ils sont inaccessibles.

2. Modifiez les éléments du groupe.
3. Choisissez **Édition > Tout modifier** ou double-cliquez sur un espace vierge de la scène avec l'outil Flèche.

Flash restitue au groupe son état d'entité unique et vous pouvez travailler avec d'autres éléments de la scène.

### VIII-Les textes :

Vous pouvez définir les attributs de police et de paragraphe d'un texte. Une police est un ensemble de caractères alphanumériques d'un dessin particulier. Les attributs d'une police sont famille, taille, style,

couleur, inter lettrage, créneau automatique et le décalage de la ligne de base (vous pouvez utiliser des polices intégrées ou des polices de périphérique. Les attributs d'un paragraphe sont alignement, marges, retraits et interlignage.

Pour modifier les attributs de police et de paragraphe, vous utilisez le panneau Caractère et le panneau Paragraphe. Pour apprendre à Flash d'utiliser des polices de périphérique plutôt que des polices intégrées, vous utilisez le panneau Options de texte.

Lors de la création d'un nouveau texte, Flash utilise les attributs du texte en cours. Pour modifier les attributs de police ou de paragraphe du texte existant, vous devez d'abord sélectionner le texte.

### IX-L Importation des médias :

Votre animation Flash peut utiliser des illustrations créées dans d'autres applications. Vous pouvez importer des graphiques vectoriels, des bitmaps et des clips vidéo dans différents formats de fichier.

Flash reconnaît divers formats de fichier vectoriel, bitmap et vidéo. Vous pouvez insérer des illustrations dans Flash en les important ou en les collant. Flash importe des graphiques vectoriels, des bitmaps, des séquences d'images et des clips vidéo.

### X-Les symboles graphiques :

Un *symbole* est un graphique, un bouton ou un clip que vous créez une seule fois et que vous pouvez réutiliser dans votre animation ou dans d'autres animations. Tout symbole que vous créez devient automatiquement un élément de la bibliothèque. Une occurrence est une copie d'un symbole situé sur la scène ou imbriqué dans un autre symbole. Une occurrence peut avoir une couleur, une taille et une fonction différentes de celles de son symbole. La modification d'un symbole met à jour toutes ses occurrences. En revanche, la modification de l'occurrence d'un symbole ne met à jour que cette occurrence.

**Remarque :** *vous pouvez aussi créer des symboles de police dans Flash.*

L'utilisation de symboles dans les animations, réduit sensiblement la taille du fichier ; l'enregistrement de plusieurs occurrences d'un symbole exige, en effet, moins d'espace disque que l'enregistrement d'une description complète de l'élément pour chaque occurrence. Vous pouvez, par exemple, réduire la taille de fichier de vos animations si vous convertissez des graphiques statiques tels que les images d'arrière-plan en symboles réutilisables. L'utilisation de symboles peut également accélérer la lecture des animations dans la mesure où un symbole n'a besoin d'être téléchargé qu'une seule fois sur le navigateur.

### XI-Les animations :

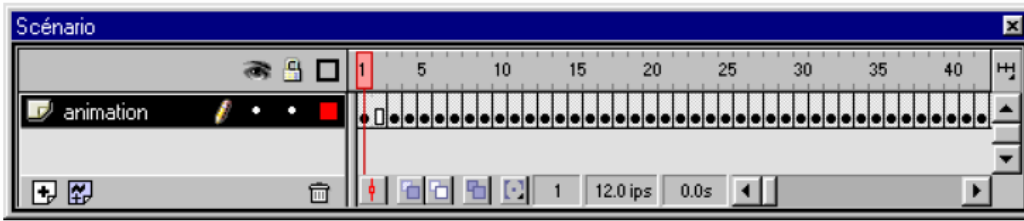
Vous pouvez créer de l'animation en modifiant le contenu d'images successives. Vous pouvez faire déplacer un objet sur la scène, augmenter ou diminuer sa taille, le faire pivoter, modifier sa couleur, effectuer des fondus en entrée ou en sortie, ou modifier sa forme. Les modifications peuvent se faire indépendamment ou avec d'autres modifications. Par exemple, vous pouvez faire pivoter un objet et effectuer un fondu en entrée lorsqu'il se déplace sur la scène.

Il existe deux méthodes de création d'une séquence d'animation dans Flash : **animation image par image** et **animation interpolée**. Dans l'animation image par image, vous créez chaque image. Dans l'animation interpolée, vous créez l'image de début et de fin et laissez Flash créer les autres images. Flash change la taille, la rotation, la couleur ou d'autres attributs de l'objet de façon égale entre l'image de début et l'image de fin pour créer l'apparence de mouvement.

L'animation interpolée est un moyen efficace de créer du mouvement et des modifications progressivement tout en réduisant la taille du fichier. Dans une animation interpolée, Flash stocke uniquement les valeurs pour les modifications entre les images. Dans l'animation image par image, Flash stocke les valeurs de chaque image complète.

### 1-L'animation image par image :

L'animation image par image modifie le contenu de la scène dans chaque image et est plus adaptée aux animations complexes dans lesquelles le contenu change d'une image à l'autre au lieu d'être simplement déplacé. L'animation image par image accroît la taille du fichier plus rapidement qu'une animation interpolée.

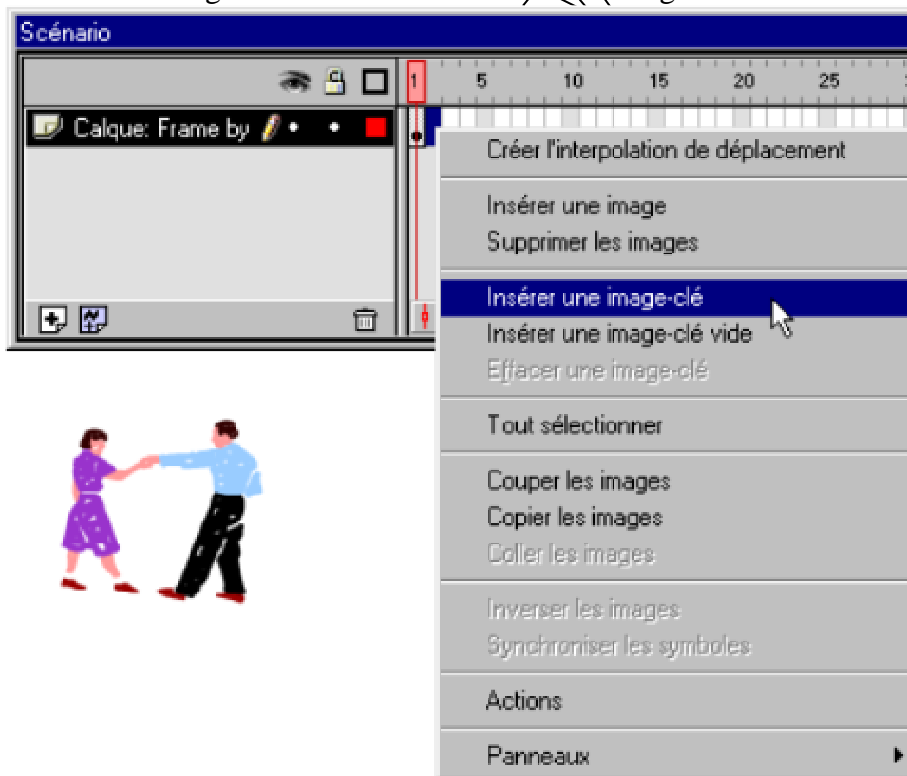


#### Comment créer une animation image par image :

1. Cliquez sur le nom d'un calque pour qu'il devienne le calque courant, puis sélectionnez une image dans le calque dans laquelle vous voulez démarrer l'animation.
2. Si l'image n'est pas encore une image-clé, sélectionnez Insertion > Image-clé pour qu'elle le devienne.
3. Créez le contenu de la première image de la séquence.

Vous pouvez utiliser les outils de dessin, coller des graphiques depuis le Presse papiers ou importer un fichier.

4. Cliquez sur l'image suivante à droite sur la même ligne et sélectionnez Insertion > Image-clé ou cliquez avec le bouton droit (Windows) ou tout en appuyant sur la touche Contrôle (Macintosh) et sélectionnez Insérer une image-clé dans le menu contextuel Image.



Une nouvelle image-clé est ajoutée, dont le contenu est identique à celui de la première image-clé.

5. Modifiez le contenu de cette image sur la scène pour développer la prochaine incrémentation de l'animation.



6. Pour terminer la séquence d'animation image par image, répétez les étapes 4 et 5 jusqu'à ce le mouvement souhaité soit créé.

Pour tester la séquence d'animation, sélectionnez Contrôle > Lire ou cliquez sur le bouton Lire sur le Contrôleur.

Flash peut être utile de visualiser l'animation à mesure que vous la créez.

## 2-L' animation interpolée :

Flash peut créer deux types d'animation **interpolée**. Dans l'**interpolation de mouvement**, vous définissez des propriétés telles que la position, la taille et la rotation d'une occurrence, d'un groupe ou d'un bloc de texte à un point dans le temps, puis vous modifiez ces propriétés à un autre point dans le temps. Dans l'**interpolation de forme**, vous dessinez une forme à un point dans le temps, puis vous modifiez cette forme ou dessinez une autre forme à un autre point dans le temps. Flash interpole ces valeurs ou ces formes pour les autres images, créant ainsi l'animation.

### a-L'interpolation de forme :

En interpolant des formes, vous pouvez créer un effet similaire au morphisme, une forme semblant se transformer progressivement en une autre forme. Flash peut également interpoler l'emplacement, la taille et la couleur des formes.



Interpoler une forme à la fois donne en général les meilleurs résultats. Si vous interpolatez plusieurs formes à la fois, toutes les formes doivent se trouver sur le même calque.

**Remarque :** Flash ne peut pas interpoler la forme de groupes, symboles, blocs de textes ou bitmaps.

Utilisez **Modification > Séparer** pour appliquer l'interpolation de forme à tous ces éléments.

Pour contrôler des modifications de forme plus complexes ou improbables, utilisez les repères de forme qui contrôlent le mouvement des parties de la forme d'origine dans la nouvelle forme.

### Pour interpoler une forme :

1. Cliquez sur le nom d'un calque pour qu'il devienne le calque courant, puis sélectionnez une image-clé vide dans laquelle vous voulez démarrer l'animation.
2. Créez le contenu de la première image de la séquence.  
Utilisez n'importe quel outil de dessin pour créer une forme.
3. Créez une deuxième image-clé après le nombre souhaité d'images après la première image.
4. Créez le contenu de la dernière image de la séquence (vous pouvez interpoler la forme, la couleur ou l'emplacement de l'image créée à l'étape 2).

5. Sélectionnez **Fenêtre > Panneaux > Image**.
  6. Pour l'interpolation, sélectionnez **Forme**.
  7. Cliquez sur la flèche en regard de l'option **Accélération** et faites-la glisser ou entrez une valeur pour ajuster le taux de modification entre des images interpolées :
    - Pour commencer l'interpolation de forme lentement et l'accélérer vers la fin de l'animation, déplacez le curseur vers le bas ou entrez une valeur comprise entre -1 et -100.
    - Pour commencer l'interpolation de forme rapidement et ralentir vers la fin de l'animation, déplacez le curseur vers le haut ou entrez une valeur comprise entre 1 et 100.

Par défaut, le taux de modification entre les images interpolées est constant. L'option **Accélération** permet de donner un aspect naturel à la transformation en ajustant progressivement la cadence de modification.
- Sélectionnez une option pour **Mélange** :
- **Distributif** permet de créer une animation dans laquelle les formes intermédiaires sont plus lisses et plus irrégulières.
  - **Angulaire** permet de créer une animation qui préserve les angles apparents et les lignes droites dans les formes intermédiaires.

**Remarque :** *Angulaire est utile uniquement pour les formes mélangées contenant des angles aigus et des lignes droites. Si vous choisissez de ne pas avoir d'angles, Flash revient à l'option d'interpolation de forme Distributif.*

### ***b-L'interpolation de mouvement :***

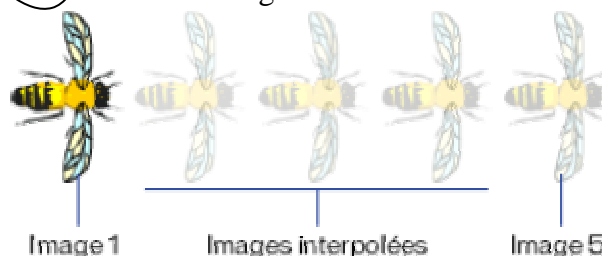
Pour interpoler les modifications des propriétés des instances, groupes et caractères, utilisez l'interpolation de mouvement. Flash peut interpoler la position, la taille, la rotation et l'inclinaison des occurrences, groupes et caractères. En outre, Flash peut interpoler la couleur d'occurrences ou de caractères, en créant des décalages de couleur graduels ou en effectuant un fondu en entrée ou en sortie d'une occurrence. Pour interpoler la couleur de groupes ou de caractères, vous devez les convertir en symboles.

Si vous modifiez le nombre d'images entre deux images-clés, ou déplacez le groupe ou symbole dans n'importe quelle image-clé, Flash interpole à nouveau automatiquement les images.

Vous pouvez créer une interpolation de mouvement à l'aide de l'une des deux méthodes :

- Créez les images-clés de début et de fin de l'animation et utilisez l'option **Interpolation de mouvement** dans le panneau **Propriétés** de l'image.
- Créez la première image-clé de l'animation, puis sélectionnez **Insertion > Créer une interpolation de mouvement** et déplacez l'objet vers le nouvel emplacement sur la scène. Flash crée automatiquement l'image-clé de fin.

Lorsque vous interpoliez la position, vous pouvez faire déplacer l'objet selon une trajectoire non linéaire. Consultez « **Interpolation de mouvement avec un guide** ».



*Les deuxième, troisième et quatrième images de l'abeille résultent de l'interpolation des première et dernière images-clés.*

### **❖ Pour créer une interpolation de mouvement à l'aide de l'option **Interpolation de mouvement** :**

1. Cliquez sur le nom d'un calque pour qu'il devienne le calque courant, puis sélectionnez une image-clé vide dans le calque dans laquelle vous voulez démarrer l'animation.



2. Créez une occurrence d'un groupe ou un bloc de texte sur la scène ou faites glisser une occurrence d'un symbole à partir de la fenêtre Bibliothèque.

*Pour effectuer une interpolation de mouvement d'un objet que vous avez dessiné, vous devez le convertir en symbole.*

3. Créez une deuxième image-clé à l'endroit où vous souhaitez que l'animation se termine.
4. Effectuez une ou des opérations suivantes pour modifier l'occurrence, le groupe ou le bloc de texte de l'image de fin :
  - Déplacez l'élément vers un nouvel emplacement.
  - Modifiez la taille, la rotation ou l'inclinaison de l'élément.
  - Modifiez la couleur de l'élément (occurrence ou bloc de texte uniquement).

*Vous pouvez également interpoler la couleur des éléments autres que des occurrences ou blocs de texte ; utilisez l'option Interpolation de couleur. Consultez « Interpolation de couleur ».*

5. Sélectionnez **Fenêtre > Panneaux > Image**.
6. Pour Interpolation, sélectionnez Mouvement.
7. Si vous avez modifié la taille de l'élément à l'étape 4, sélectionnez Échelle pour interpoler la taille de l'élément sélectionné.
8. Cliquez sur la flèche en regard de l'option Accélération et faites-la glisser ou entrez une valeur pour ajuster la cadence de modification entre des images interpolées :
  - Pour commencer l'interpolation de mouvement lentement et l'accélérer vers la fin de l'animation, déplacez le curseur vers le haut ou entrez une valeur comprise entre -1 et -100.
  - Pour commencer l'interpolation de mouvement rapidement et ralentir vers la fin de l'animation, déplacez le curseur vers le bas ou entrez une valeur comprise entre 1 et 100.

*Par défaut, le taux de modification entre les images interpolées est constant. L'option Accélération permet de donner un aspect naturel à l'accélération ou à la décélération en ajustant progressivement le taux de modification.*

9. Pour faire pivoter l'élément sélectionné au cours de l'interpolation, sélectionnez une option dans le menu Rotation :
  - Sélectionnez Aucune (valeur par défaut) pour ne pas appliquer de rotation.
  - Sélectionnez Auto pour faire pivoter l'objet une fois dans la direction nécessitant le dernier mouvement.
  - Sélectionnez Vers la droite ou Vers la gauche pour faire pivoter les objets comme indiqués, puis entrez le nombre de rotations.

**Remarque :** *il s'agit d'une rotation supplémentaire à toute rotation appliquée à l'image de fin à l'étape 4.*

10. Si vous utilisez une trajectoire de mouvement, sélectionnez Orienter vers la trajectoire pour orienter la ligne de base de l'élément interpolé vers la trajectoire. Consultez Interpolation de mouvement selon une trajectoire.
11. Sélectionnez Synchroniser pour vous assurer que l'occurrence boucle correctement dans l'animation principale.

*Utilisez la commande Synchroniser si le nombre d'images de la séquence d'animation dans le symbole n'est pas un multiple pair du nombre d'images occupées par l'occurrence graphique dans l'animation.*

12. Si vous utilisez une trajectoire de mouvement, sélectionnez Ajuster pour associer l'élément interpolé à la trajectoire par son point d'alignement.

❖ **Pour créer une interpolation de mouvement à l'aide de la commande Interpolation de mouvement :**

1. Sélectionnez une image-clé vide et dessiner un objet sur la scène ou faites glisser une occurrence d'un symbole à partir de la fenêtre Bibliothèque.
2. Sélectionnez **Insertion > Créer une interpolation de mouvement**.

Si vous avez dessiné un objet à l'étape 1, Flash convertit automatiquement l'objet en un symbole et lui affecte le nom Interpolation 1. Si vous avez dessiné plusieurs objets, ceux-ci sont appelés Interpolation 2, Interpolation 3 et ainsi de suite.

3. Cliquez à l'intérieur de l'image dans laquelle vous voulez que l'animation se termine et sélectionnez **Insertion > Image**.
4. Déplacez l'objet, l'occurrence ou le bloc de caractères sur la scène vers l'emplacement souhaité. Ajustez la taille de l'élément si vous voulez interpoler son échelle. Ajustez la rotation de l'élément si vous voulez interpoler sa rotation. Désélectionnez l'objet lorsque vous avez terminé les ajustements. L'image-clé est automatiquement ajoutée à la fin de la plage d'images.



5. Sélectionnez l'image-clé à la fin de l'interpolation de mouvement et sélectionnez **Fenêtre > Panneaux > Image**. L'interpolation de mouvement doit être automatiquement sélectionnée dans le panneau Image.
6. Si vous avez modifié la taille de l'élément à l'étape 4, sélectionnez **Echelle** pour interpoler la taille de l'élément sélectionné.
7. Cliquez sur la flèche en regard de l'option **Accélération** et faites-la glisser ou entrez une valeur pour ajuster le taux de modification entre des images interpolées :
  - Pour commencer l'interpolation de mouvement lentement et l'accélérer vers la fin de l'animation, déplacez le curseur vers le haut ou entrez une valeur comprise entre -1 et -100.
  - Pour commencer l'interpolation de mouvement rapidement et ralentir vers la fin de l'animation, déplacez le curseur vers le bas ou entrez une valeur comprise entre 1 et 100.

*Par défaut, le taux de modification entre les images interpolées est constant.  
L'option **Accélération** permet de donner un aspect naturel à l'accélération ou à la décélération en ajustant progressivement le taux de modification.*
8. Pour faire pivoter l'élément sélectionné au cours de l'interpolation, sélectionnez une option dans le menu **Rotation** :
  - Sélectionnez **Aucune** (valeur par défaut) pour ne pas appliquer de rotation.
  - Sélectionnez **Auto** pour faire pivoter l'objet une fois dans la direction nécessitant le dernier mouvement.
  - Sélectionnez **Vers la droite** ou **Vers la gauche** pour faire pivoter les objets comme indiqué, puis entrez le nombre de rotations.

**Remarque :** il s'agit d'une rotation supplémentaire à toute rotation appliquée à l'image de fin à l'étape 4.

9. Si vous utilisez une trajectoire de mouvement, sélectionnez **Orienter** vers la trajectoire pour orienter la ligne de base de l'élément interpolé vers la trajectoire. Consultez **Interpolation de mouvement selon une trajectoire**.
10. Sélectionnez **Synchroniser** pour vous assurer que l'occurrence boucle correctement dans l'animation principale.
 

*Utilisez la commande **Synchroniser** si le nombre d'images de la séquence d'animation dans le symbole n'est pas un multiple pair du nombre d'images occupées par l'occurrence graphique dans l'animation.*
11. Sélectionnez **Ajuster** pour associer l'élément interpolé à la trajectoire par son point d'alignement.

#### **c-L'interpolation de mouvement avec un guide :**

Les calques de guide de mouvement vous permettent de dessiner des trajectoires le long desquelles des occurrences, des groupes ou des blocs de textes interpolés peuvent être animés. Vous pouvez lier plusieurs

calques à un calque de guide de déplacement pour que plusieurs objets suivent la même trajectoire. Un calque normal lié à un calque de guide de déplacement devient un calque guidé.

**Pour créer une trajectoire de mouvement pour une animation interpolée :**

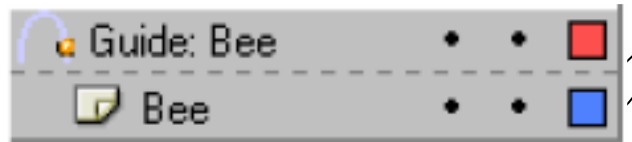
1. Créez une séquence d'animation d'interpolation de mouvement comme décrit à la section « Interpolations de mouvement ».

*Si vous sélectionnez Orienter vers la trajectoire, la ligne de base de l'élément interpolé s'orientera vers la trajectoire de mouvement. Si vous sélectionnez Ajuster, le point d'alignement de l'élément interpolé s'ajustera à la trajectoire de mouvement.*

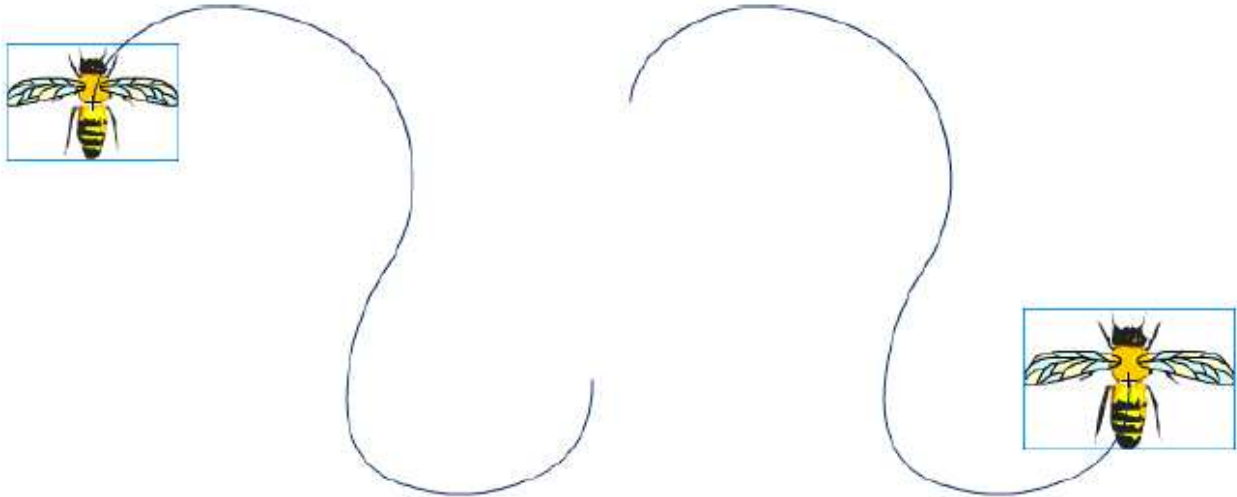
2. Procédez de l'une des façons suivantes :

- Sélectionnez le calque contenant l'animation et sélectionnez **Insertion > Guide de déplacement**.
- Cliquez avec le bouton droit sur le calque contenant l'animation et sélectionnez Ajouter un guide de déplacement.

*Flash crée un calque sur le calque sélectionné avec une icône de guide de déplacement à gauche du nom du calque.*



3. Utilisez l'outil Plume, Crayon, Cercle, Rectangle ou Pinceau pour dessiner la trajectoire souhaitée.



4. Ajustez le centre avec le début de la ligne dans la première image et à la fin de la ligne dans la dernière image.

**Remarque :** faites glisser le symbole par son point d'alignement pour obtenir les meilleurs ajustements.

5. Pour masquer le calque de guide de déplacement et la ligne de sorte que seul le mouvement de l'objet soit visible lorsque vous travaillez, cliquez sur la colonne OEil sur le calque de guide de déplacement.

*Le groupe ou symbole suit la trajectoire lorsque vous exécutez l'animation.*



**Pour lier des calques à un calque de guide de déplacement, exécutez l'une des opérations suivantes :**

- Faites glisser un calque existant sous le calque de guide de déplacement. Le calque est placé intentionnellement sous le calque de guide de déplacement. Tous les objets se trouvant sur ce calque sont automatiquement ajustés à la trajectoire.
- Créez un calque sous le calque de guide de déplacement. Les objets que vous interposez sur ce calque sont automatiquement interpolés selon la trajectoire.
- Sélectionnez un calque sous le calque de guide de déplacement. Sélectionnez **Modification > Calque**, puis sélectionnez Guide dans la boîte de dialogue Propriétés du calque.
- Cliquez sur le calque tout en appuyant sur la touche Alt.

❖ **Pour supprimer le lien entre les calques et le calque de guide de déplacement :**

1. Sélectionnez le calque dont vous voulez supprimer le lien.
2. Procédez de l'une des façons suivantes :
  - Faites glisser le calque au-dessus du calque de guide de déplacement.
  - Sélectionnez **Modification > Calque** et sélectionnez le type de calque Normal dans la boîte de dialogue Propriétés du calque.
  - Cliquez sur le calque tout en appuyant sur la touche Alt.

## XII-Les boutons :

Les boutons sont en réalité des clips interactifs composés de quatre images. Lorsque vous sélectionnez le comportement du bouton pour un symbole, Flash crée un scénario avec quatre images. Les trois premières images affichent les trois différents états du bouton et la quatrième définit la zone active du bouton. En réalité, le scénario n'est pas lu ; il réagit simplement aux mouvements et aux actions du pointeur en se plaçant sur l'image appropriée.

Pour rendre un bouton interactif dans une animation, placez une occurrence du symbole du bouton sur la scène et affectez-lui des actions. Vous devez affecter les actions à l'occurrence du bouton dans l'animation et non aux images dans le scénario du bouton.

### 1-L'insertion d'un bouton :

Chaque image dans le scénario d'un symbole de bouton possède une fonction spécifique :

- La première image, l'état Haut, représente l'apparence normale du bouton quand le pointeur n'est pas dessus.
- La deuxième image, l'état Dessus, représente l'apparence du bouton quand le pointeur se trouve dessus.
- La troisième image, l'état Abaissé, représente l'apparence du bouton quand vous cliquez dessus.
- La quatrième image, l'état Cliqué, définit la zone sensible qui réagit au clic de la souris. Cette zone est invisible dans l'animation.

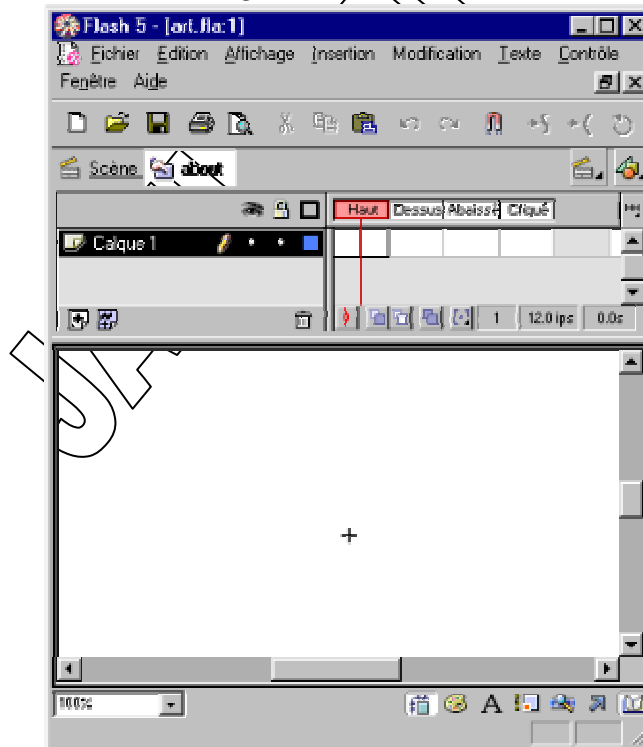
Home Home Home

Contenu type des images Haut, Dessus, Abaissé et Cliqué

Pour une leçon interactive concernant la création de boutons dans Flash, choisissez Aide > Leçons > Boutons.

#### Pour créer un bouton :

1. Choisissez **Édition** > Tout **désélectionner** pour vous assurer que rien n'est sélectionné sur la scène.
2. Choisissez **Insertion** > **Nouveau symbole** ou appuyez sur Ctrl+F8.  
*Pour créer le bouton, convertissez les images du bouton en images-clés.*
3. Dans la boîte de dialogue Propriétés du symbole, entrez le nom du nouveau symbole de bouton et dans Comportement, choisissez Bouton.  
*Flash bascule en mode édition de symboles. L'en-tête du scénario change pour afficher quatre images consécutives intitulées Haut, Dessus, Abaissé et Cliqué.*  
*La première image, Haut, est une image-clé vide.*

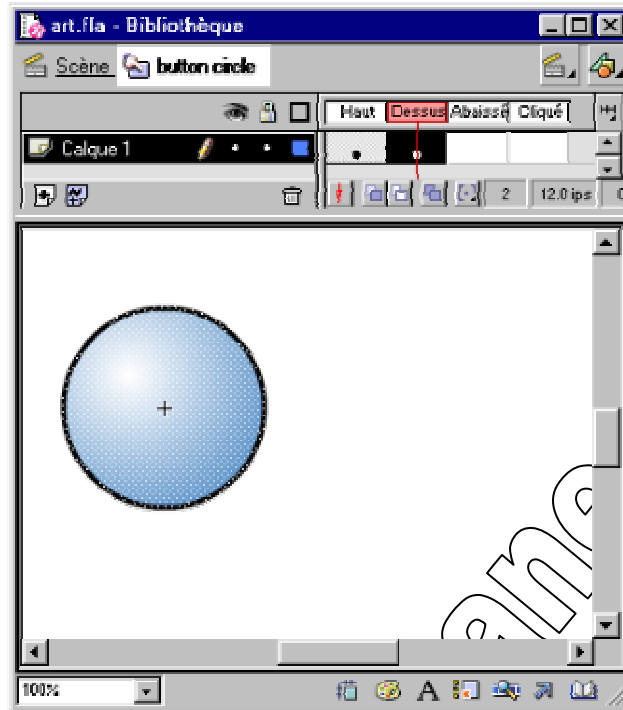


4. Pour créer l'image du bouton dans un état Haut, utilisez les outils de dessin, importez un graphique ou placez l'occurrence d'un autre symbole sur la scène.



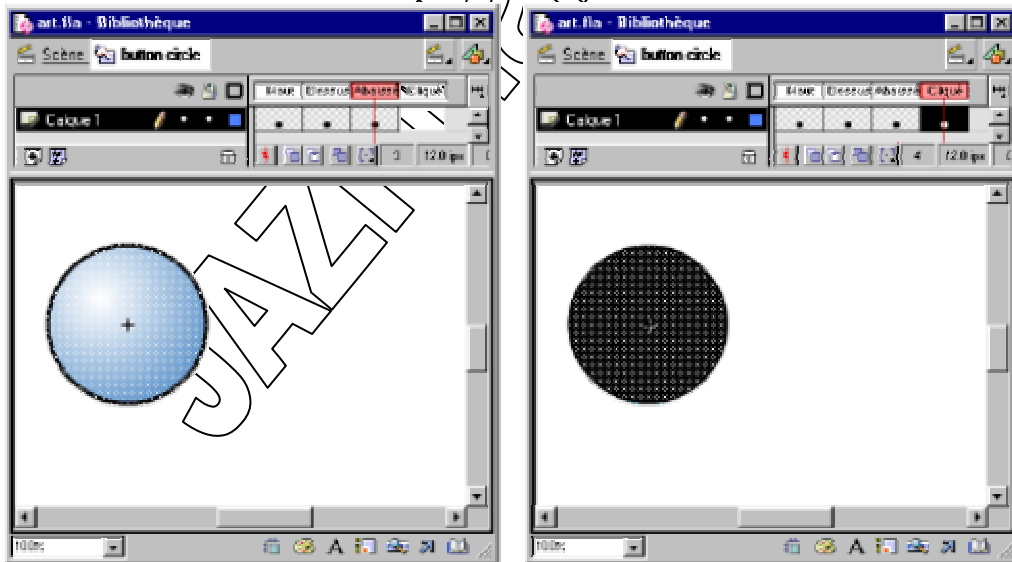
Dans un bouton, vous pouvez utiliser un symbole graphique ou un symbole de clip, mais vous ne pouvez pas utiliser un autre bouton dans un bouton. Si vous souhaitez créer un bouton animé, utilisez un symbole de clip.

5. Cliquez sur la deuxième image, intitulée Dessus, et choisissez **Insertion > Image-clé**.



Flash insère une image-clé qui duplique le contenu de l'image Haut.

6. Remplacez l'image du bouton par l'état Dessus.
7. Répétez les étapes 5 et 6 pour les images Abaissé et Cliqué.



L'image de l'état Cliqué n'est pas visible sur la scène, mais elle définit la zone sensible du bouton qui réagit au clic. Assurez-vous que le graphique de l'image Cliqué est une zone pleine suffisamment large pour englober tous les éléments graphiques des images Haut, Abaissé et Dessus. Elle peut aussi être plus large que le bouton visible. Si vous ne spécifiez pas d'image pour l'état Cliqué, l'image de l'état Haut sert alors d'image pour l'état Cliqué.

Vous pouvez créer un effet de survol décalé en plaçant l'image Cliqué dans un endroit différent des autres images de bouton.

8. Pour associer un son à un état du bouton, sélectionnez l'image correspondant à l'état dans le scénario, choisissez **Modification** > **Image** pour afficher le panneau Image, puis cliquez sur l'onglet Son dans ce même panneau.
9. Lorsque vous avez terminé, choisissez **Édition** > **Modifier l'animation**. Faites glisser le symbole du bouton hors de la fenêtre Bibliothèque pour en créer une occurrence dans l'animation.

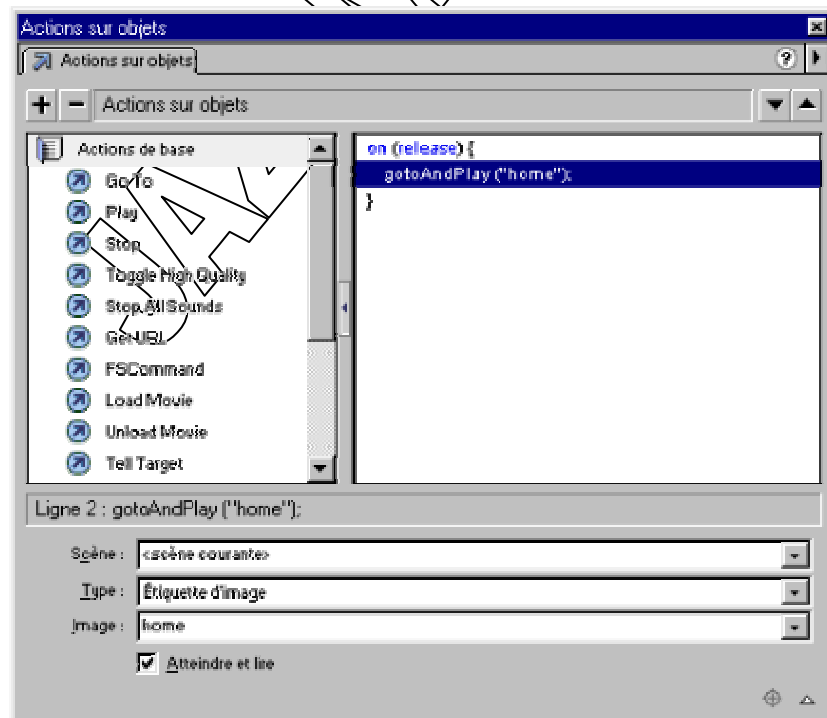
## 2-L'action d'un bouton :

Vous pouvez affecter une action à un bouton ou à un clip afin d'exécuter une action lorsque l'utilisateur clique sur un bouton ou place le pointeur dessus, ou lorsqu'un clip est chargé ou arrive à une image spécifique. Vous attribuez l'action à une occurrence du bouton ou du clip ; les autres occurrences du symbole ne sont pas affectées.

Les instructions suivantes décrivent comment définir des actions pour des objets à l'aide du panneau Actions en mode Normal. Pour plus d'informations sur l'utilisation du panneau Actions en mode Expert, consultez le *Guide de référence Action Script*.

### Pour affecter une action à un bouton ou à un clip :

1. Sélectionnez une occurrence de bouton ou de clip, puis choisissez **Fenêtre** > **Actions**.  
*Si la sélection n'est pas une occurrence de bouton, une occurrence de clip ou une image, ou si la sélection inclut plusieurs objets, le panneau Actions sera estompé.*
2. Dans la liste de zone d'outils, à gauche du panneau, cliquez sur la catégorie Actions de base pour afficher les actions de base.
3. Pour affecter une action, exécutez l'une des opérations suivantes :
  - Double-cliquez sur une action de la catégorie Actions de base dans la liste de zone d'outils, du côté gauche du panneau.
  - Faites glisser une action depuis la catégorie Actions de base, située à gauche, jusqu'à la fenêtre Script, du côté droit du panneau.
  - Cliquez sur le bouton Ajouter (+) et choisissez une action dans le menu contextuel.
  - Utilisez les raccourcis clavier.



Si vous avez sélectionné un clip, Flash insère automatiquement une action *On Clip Event* et l'action sélectionnée dans la fenêtre *Script*. Si vous avez sélectionné un bouton, Flash insère automatiquement le code *On Mouse Event* pour déclencher toute action sélectionnée.

4. Cliquez sur le bouton Paramètres, à l'angle inférieur droit du panneau Actions, pour afficher les champs de paramètres. Sélectionnez une action et entrez de nouvelles valeurs dans les champs de paramètres pour modifier les paramètres des actions existantes.

Les paramètres varient en fonction de l'action que vous choisissez. Par exemple, le paramètre *On* par défaut est *Load*. Pour plus d'informations sur les paramètres des actions les plus souvent choisies.

Répétez les étapes 3 et 4 pour attribuer d'autres actions si nécessaire.

## VII-L'ActionScript :

Flash utilise le langage de script ActionScript pour ajouter des interactions dans une animation. Proche du JavaScript, ActionScript est un langage de programmation orienté objet. Dans les scripts orientés objet, vous organisez les informations sous forme de groupes appelés classes. Vous pouvez créer plusieurs occurrences d'une classe, appelées objets, afin de les utiliser dans vos scripts. Vous pouvez utiliser les classes prédéfinies ActionScript et créer les vôtres.

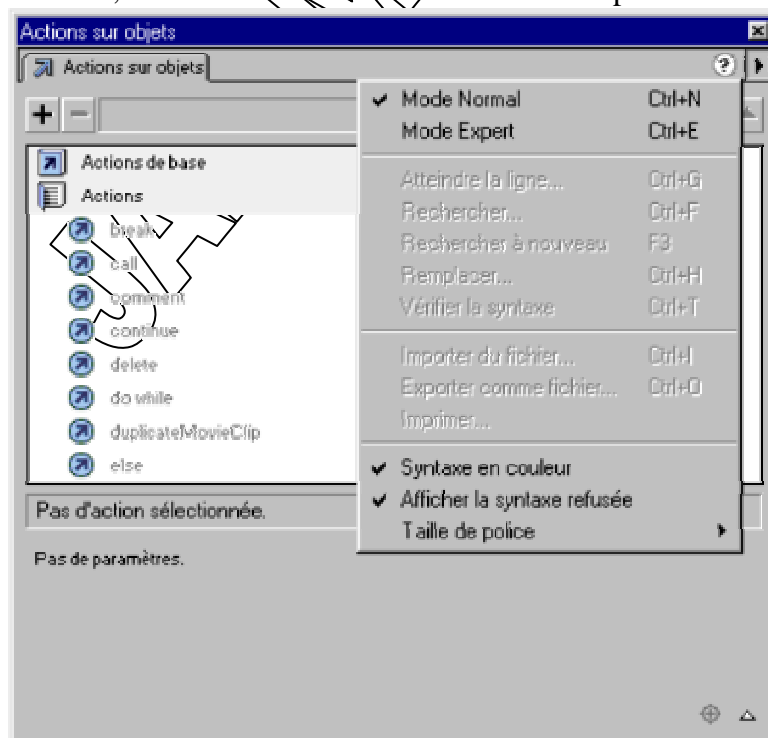
### Pour afficher le panneau Actions :

Choisissez **Fenêtre > Actions**.

Lorsque vous sélectionnez une occurrence d'un bouton, d'un clip ou d'une image le panneau Actions devient actif. Le titre du panneau Actions change pour Actions sur objets si vous sélectionnez un bouton ou un clip, ou Actions d'image si vous sélectionnez une image.

### Pour sélectionner un mode d'édition des actions :

1. Avec le panneau Actions affiché, cliquez sur la flèche située à l'angle supérieur droit du panneau pour afficher le menu contextuel.
2. Dans le menu contextuel, choisissez Mode Normal ou Mode Expert.



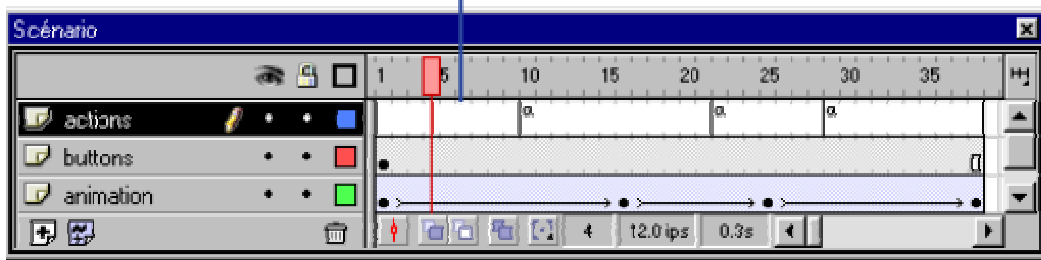
Chaque script conserve son propre mode. Par exemple, si vous écrivez un script d'une occurrence de bouton en mode Normal et un autre en mode Expert et que vous basculez entre les boutons sélectionnés, le mode du panneau bascule lui aussi.

### Attribution d'actions aux images

Pour faire en sorte qu'une animation exécute une action lorsqu'elle arrive à une image-clé, vous attribuez une action d'image à l'image-clé. Par exemple, pour créer une boucle dans une animation, vous pouvez ajouter à l'image 20 une action d'image qui spécifie « aller à l'image 10 et lire ».

Il est préférable de placer toutes les actions d'image dans un calque afin d'en faciliter le suivi. Les images comportant des actions affichent un petit *a* dans le scénario.

Image avec actions



Une fois que vous avez affecté une action, il est recommandé de tester son fonctionnement en utilisant la commande **Contrôle > Tester l'animation**. La plupart des actions ne fonctionneront pas en mode édition. Les instructions suivantes décrivent comment définir des actions d'image à l'aide du panneau Actions en mode Normal. Pour plus d'informations sur l'utilisation du panneau Actions en mode Expert, consultez le *Guide de référence ActionScript*.

#### Pour attribuer une action à une image-clé :

1. Sélectionnez une image-clé dans le scénario, puis choisissez **Fenêtre > Actions**.

Si l'image sélectionnée n'est pas une image-clé, l'action est affectée à l'image-clé précédente. Si la sélection n'est pas une image ou si la sélection contient plusieurs images, le panneau Actions sera estompé. (Pour plus d'informations sur l'affectation d'une action à un bouton ou à un clip, consultez « L'action d'un bouton »).

### XIV-La publication :

Une fois que vous êtes prêt à partager votre animation avec un public, vous devez publier ou exporter le fichier FLA de Flash dans un autre format de lecture.

La fonction Publier de Flash est destinée à la présentation d'animations sur le Web. La commande Publier crée le fichier Flash Player (SWF) ainsi qu'un document HTML qui insère votre fichier Flash Player dans une fenêtre de navigation.

La commande Exporter l'animation vous permet de créer un contenu Flash pouvant être modifié dans d'autres applications et d'exporter une animation directement dans un format unique. Par exemple, vous pouvez exporter une animation entière sous la forme d'un fichier Flash Player, d'une série d'images bitmap, d'un fichier unique d'images et sous la forme d'images mobiles ou immobiles dans divers formats dont GIF, JPEG, PNG, BMP, PICT, QuickTime et AVI.

Avec la commande Publier, vous pouvez effectuer les opérations suivantes :

- Choisir les formats dans lesquels vous souhaitez délivrer le fichier de création et ajuster les paramètres pour un format de fichier particulier. Flash publie automatiquement le fichier de création

dans les formats sélectionnés, crée des fichiers supplémentaires basés sur les paramètres sélectionnés et stocke les paramètres avec le fichier de l'animation en vue d'une réutilisation.

*Les options Tester l'animation correspondent généralement à celles de la publication mais ne sauvegardent pas les paramètres pour une réutilisation.*

- Créer des formats de fichier alternatifs (GIF, JPEG, PNG et QuickTime) et le document HTML nécessaire pour les afficher dans la fenêtre de navigation. Les formats alternatifs permettent à un navigateur d'afficher les effets animés de votre animation et son interactivité pour des utilisateurs n'ayant pas installé le Flash Player.
- Utiliser des modèles Generator pour mettre facilement à jour le contenu d'un site Web (par exemple, des graphiques et du texte), sans avoir à remplacer les fichiers individuellement. Par exemple, dans Flash, vous pouvez utiliser des données Generator comme variables pour fournir aux visiteurs de votre site Web Flash une remontée des informations immédiate ou personnalisée, pour rendre la production de votre site Web Flash plus efficace et pour créer des illustrations (des listes déroulantes par exemple) que vous ne pouvez pas créer seulement avec Flash.

Une alternative à l'utilisation de la commande Publier, si vous maîtrisez le langage HTML, consiste à créer votre propre document HTML avec un éditeur HTML et à inclure les balises nécessaires pour afficher une animation Flash. Si vous possédez Macromedia Dreamweaver, vous pouvez facilement ajouter une animation Flash dans votre site Web. Dreamweaver génère tous les codes HTML nécessaires.

Avant de publier votre animation, il est important d'en tester le fonctionnement avec les commandes Tester l'animation et Tester la scène.



## Chapitre 2 :

### Partie B :

## Pages Web Statiques

### Sous partie B1 :

## Le langage HTML

### I-Introduction au langage HTML :

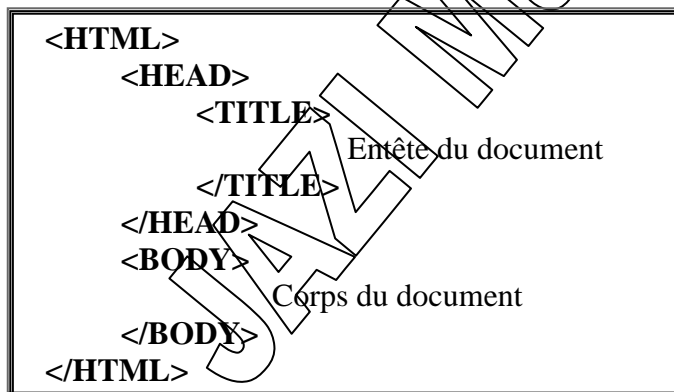
Une page Web est un document contenant aussi bien du simple texte, que des liens vers d'autres pages, des images (fixes et animées), des séquences animées (vidéo) et des fichiers sons. Elle sont écrites avec un langage nommé **HTML**.

**HTML** c'est l'abréviation de : **Hyper Text Markup Langage**, ce que l'on pourrait traduire par «Langage Texte avec Liens» ou «langage Hypertexte». HTML est un langage de conception de pages Web.

### II-Structure d'un document HTML :

Le texte brut d'une page HTML n'est pas facilement lisible, car la présence des balises pollue ce texte. Le navigateur (browser) interprète ces balises et ne les affiche pas, cependant il les utilise pour modifier les apparences du texte, pour afficher des images, des liens hypertextes, des séquences animées et du son.

Structure générale d'une page Web



- ♦ **<HTML>** et **</HTML>** : tous les documents HTML commenceront toujours par la balise **<HTML>** et se termineront par la balise **</HTML>**.
- ♦ **<HEAD>** et **</HEAD>** : on indique toutes les informations qui accompagnent le document (entête de la page) mais n'apparaissent pas explicitement dans la page Web, comme le nom de l'auteur, le sujet de la page, les mots clés, l'éditeur avec lequel vous avez produit le document.
- ♦ **<BODY>** et **</BODY>** : la balise **<BODY>** est la plus importante. C'est elle qui indique au browser que la page HTML commence vraiment, et que les lignes suivantes devront être formatées avant d'être affichées à l'écran. La balise **</BODY>** indique au browser que la partie affichage à l'écran est terminée.

### III-Les balises du langage HTML :

#### 1. Les textes et les paragraphes :

**a- Retour à ligne et paragraphe**

- **Retour à ligne** : `<BR>` : la balise `<BR>` (break) permet de retourner à la ligne suivante. Elle ne possède pas de balise de fin `</BR>`.
- **Paragraphe** : A la différence du passage à la ligne, le changement de paragraphe insère une ligne blanche entre la phrase précédente et celle que vous commencez à taper. Cela se fera au moyen de la balise paragraphe : `<P>`, qui se ferme avec `</P>`

Noter bien :

- ✓ La majorité des browsers tolèrent l'absence de `</P>` et considèrent un nouveau paragraphe lorsqu'ils rencontrent une nouvelle balise `<P>`.

Cette balise peut s'utiliser avec l'attribut **ALIGN** avec les valeurs : **CENTER**, **LEFT**, **RIGHT**.

Exp  
`<BODY>`

Hello world. `<P>` Ceci est ma première page au langage HTML. `<BR>` Ce n'est pas fameux, mais nous allons travailler dessus pour améliorer tout ça. `<BR>`

`</BODY>`

Résultat :

Hello world.  
 Ceci est ma première page au langage HTML.  
 Ce n'est pas fameux, mais nous allons travailler dessus pour

**b- l'apparence gras, italique et soulignée**

- ◆ `<U>...</U>` : ces balises s'utilisent pour le soulignement du texte (de l'anglais Underline).  
`<U>` texte souligné `</U>` donne texte souligné
- ◆ `<B>...</B>` : Ces balises s'utilisent pour mettre le texte en gras (**Bold**).  
`<B>` texte en gras `</B>` donne **texte en gras**
- ◆ `<I>...</I>` : Ces balises s'utilisent pour mettre le texte en italique.  
`<I>` texte en italique `</I>` donne *texte en italique*

**Noter bien :** On peut combiner les balises entre eux.

**Exp 1 :** `<H1><I>` le texte sera de grande taille et en italique `</I> </H1>`

Résultat 1 :

*le texte sera de grande taille et en italique*

**Exp 2 :** `<H4 ALIGN="RIGHT"><U>` le texte sera de taille 4, à droite et souligné `</U></H4>`

Résultat 2 :

le texte sera de taille 4, à droite et souligné

## 2. Les titres :

**<Hn...>** et **</Hn>** : appelé balise header, sont utilisées pour afficher du texte sur l'écran. Sa taille dépend de la valeur de n. **<H1>** ... **</H1>** la plus grande et **<H6>** ... **</H6>** la plus petite.

Exp :	<code>&lt;H1&gt; Titre1 &lt;/H1&gt;</code>	<b>Titre1</b>
	<code>&lt;H2&gt; Titre2 &lt;/H2&gt;</code>	<b>Titre2</b>
	<code>&lt;H3&gt; Titre3 &lt;/H3&gt;</code>	<b>Titre3</b>
	<code>&lt;H4&gt; Titre4 &lt;/H4&gt;</code>	<b>Titre4</b>
	<code>&lt;H5&gt; Titre5 &lt;/H5&gt;</code>	<b>Titre5</b>
	<code>&lt;H6&gt; Titre6 &lt;/H6&gt;</code>	<b>Titre6</b>

### Noter Bien :

- 1- Il est à noter que le texte apparaissant entre ces balises est automatiquement mis en gras.
- 2- De plus lorsque le browser rencontre la balise de fin **</Hn>**, un retour à la ligne est effectué.
- 3- Cette balise peut s'employer avec le seul attribut **ALIGN** dont la valeur peut prendre :
  - ALIGN=CENTER** : le texte sera centré.
  - ALIGN=LEFT** : l'alignement à gauche.
  - ALIGN=RIGHT** : le texte sera aligné sur la marge droite du browser.

## 3. Les images :

Pour inclure une image (quel soit GIF ou JPG) dans une page web, on utilise la balise :

**<IMG SRC="Loupe.gif" BORDER="0" WIDTH="100" HEIGHT="50">** noter bien que cette balise ne possède pas de balise de fin **</IMG>**.

### Attributs :

- **SRC** : source de l'image (non du fichier). Le fichier graphique source doit se trouver dans le même répertoire de la page à créer.

A défaut, il est nécessaire de préciser le chemin permettant d'accéder à ce fichier. Il est à noter que l'on peut utiliser une URL en référence à la place du nom du fichier.

Si par exemple on a besoin d'une image se trouvant dans le serveur Edunet. On mettra : <http://www.edunet.tn/loupe.gif>

- **BORDER** : l'épaisseur de la bordure ou de cadre défini en pixel.
- **WIDTH** : largeur de l'image.
- **HEIGHT** : hauteur de l'image.

Remarquons qu'il est possible de prévoir l'affichage d'un texte pendant le chargement de l'image. Ce texte sera introduit par la classe **ALT**. Il sera visible sous forme d'une Info bulle.

**<IMG .... ALT = "commentaires" >**

### Remarque :

Dans le cas où on veut inclure une image animée (GIF animée), il faut écrire :

**<IMG DYN SRC = "imageanim.gif" BORDER = "0" HEIGHT = "50" WIDTH = "60">**

## 4. Les listes:

### ➤ Liste non numérotée

Elle se compose d'une balise de début de liste ( Unodered list ) **<UL>**, des balises de points de la liste ( liste item ) **<LI>** et d'une balise de fin de liste **</UL>**. Elle s'utilise avec l'attribut **TYPE** qui permet de forcer l'affichage de points, cercles ou carrés à la place de points utilisés par défauts (**CIRCLE**, **DISC**, **SQUARE**).

<p>Exp :</p> <pre> &lt;BODY&gt; &lt;H1&gt; Les mois du printemps&lt;/H1&gt; &lt;UL&gt;&lt;LI&gt;Jan&lt;/LI&gt; &lt;LI&gt;Mai&lt;/LI&gt; &lt;LI&gt;Juin&lt;/LI&gt;&lt;/UL&gt; &lt;P&gt; &lt;H3&gt; Les mois d'automne&lt;/H3&gt; &lt;UL&gt;&lt;LI&gt;Octobre &lt;LI&gt;Novembre &lt;LI&gt;Décembre&lt;/UL&gt; &lt;/BODY &gt;                 </pre>	<p>Résultat :</p> <p style="text-align: center;"><b>Les mois du printemps</b></p> <ul style="list-style-type: none"> <li>• Avril</li> <li>• Mai</li> <li>• Juin</li> </ul> <p style="text-align: center;"><b>Les mois d'automne</b></p> <ul style="list-style-type: none"> <li>• Octobre</li> <li>• Novembre</li> <li>• Décembre</li> </ul>
--	---

➤ **Liste numérotée**

Elle se compose de la balise **ordered list** <OL> et </OL>. La balise <LI> quant à elle servira toujours pour indiquer le numéro de l'item sélectionné. Elle s'utilise avec les attributs.

- **START** : permet de modifier le numéro initial de la liste
- **TYPE** : détermine le type de séquence à utiliser

Pour l'itération de la liste les valeurs peuvent être :

- A** : Classement alphabétique en majuscules
- a** : Classement alphabétique en minuscules
- I** : Classement en chiffres romains en majuscules.
- i** : Classement en chiffres romains en minuscules.
- 1** : Classement en chiffres arabes (par défaut).

<p>Exp :</p> <pre> &lt;BODY&gt; &lt;H2&gt; Liste commençant à partir de 1&lt;/H2&gt; &lt;OL START=1&gt; &lt;LI&gt;Pascal &lt;LI&gt;Cobol &lt;LI&gt;C++&lt;/OL&gt; &lt;P&gt; &lt;H3&gt; Liste commençant à partir de 5 &lt;/H3&gt; &lt;OL TYPE=I START= 5 &gt;&lt;LI&gt;Word &lt;LI&gt;Excell &lt;LI&gt;Power point&lt;/OL&gt; &lt;/BODY &gt;                 </pre>	<p>Résultat :</p> <p><b>Liste commençant à partir de 1</b></p> <ol style="list-style-type: none"> <li>1. Pascal</li> <li>2. Cobol</li> <li>3. C++</li> </ol> <p><b>Liste commençant à partir de 5</b></p> <ol style="list-style-type: none"> <li>V. Word</li> <li>VI. Excel</li> <li>VII. Power point</li> </ol>
---	--

**5. Les tableaux :**

Il est souvent utile de présenter des informations mieux structurées qu'avec des listes. Les tableaux permettent de les afficher en lignes et en colonnes. Les tableaux sont définis comme étant des suites de lignes.

Un tableau doit respecter les quelques règles suivantes :

- Le tableau est encadré par les balises <TABLE> et </TABLE>.
- Le titre du tableau est encadré par <CAPTION> </CAPTION>
- Chaque ligne est encadrée par <TR> </TR> (Table Row, traduisez par *ligne du tableau*).
- Les cellules d'en-tête sont encadrées par <TH> </TH> (pour *Table Header* : En-tête de tableau)
- Les cellules de valeur sont encadrées par <TD> </TD> (Table Data: Donnée de tableau)

Voici un exemple de tableau HTML :

```

<TABLE BORDER="1">
<CAPTION> Voici le titre du tableau </CAPTION>
<TR>

```

```

<TH> Titre A1 </TH>
<TH> Titre A2 </TH>
<TH> Titre A3 </TH>
<TH> Titre A4 </TH>
</TR>
<TR>
<TH> Titre B1 </TH>
<TD> Valeur B2 </TD>
<TD> Valeur B3 </TD>
<TD> Valeur B4 </TD>
</TR>
</TABLE>
    
```

Le code donne le résultat suivant :

Voici le titre du tableau

<b>Titre A1</b>	<b>Titre A2</b>	<b>Titre A3</b>	<b>Titre A4</b>
<b>Titre B1</b>	Valeur B2	Valeur B3	Valeur B4

Les attributs :

Attribut	Balises auxquelles il s'applique	Valeur	Effet Visuel
ALIGN	THEAD TBODY TH TR TD	CENTER LEFT RIGHT JUSTIFY	centré Gauche Droite Justifié
	CAPTION	TOP BOTTOM	Au-dessus En-dessous
VALIGN (alignement vertical)	THEAD TBODY TH TR TD	TOP MIDDLE BOTTOM	En haut Au milieu En bas
BORDER=n	TABLE		Taille de la bordure
CELLPADDING=n	TABLE		Espacement de n pixels entre le contenu des cellules et la bordure
CELLSPACING=n	TABLE		Epaisseur de la grille intérieure
FLOAT	TABLE	RIGHT LEFT	Position du texte qui suivra </TABLE>
COLS=n	TABLE		Nombre de colonnes
FRAME (contrôle les éléments individuels d'encadrement du tableau)	TABLE	NONE TOP BOTTOM TOPBOT SIDES ALL	Aucun au-dessus En bas Tout en haut Sur les cotés Tous
RULES (contrôle les éléments de la grille des cellules)	TABLE	NONE BASIC ROWS COLS ALL	Aucun basique Ligne Colonne Tous
COLSPAN	THEAD TBODY TH TR TD		Débordement des cellules sur les colonnes adjacentes
ROWSPAN	THEAD TBODY TH TR TD		Débordement des cellules sur les lignes adjacentes



## 6. Les liens hypertextes :

Un lien hypertexte constitue une zone active sur laquelle on clique pour se connecter automatiquement sur une autre page. La zone active, appelée communément saut ou hyperlien, peut être un caractère, un groupe de caractères ou même une image. Cette zone active peut être entouré par une bordure lorsqu'il s'agit d'une image, souligné et ou colorée dans le cas d'un groupe de caractères. Le pointeur de la souris prendra l'aspect d'une main quand il est positionné sur le lien.

Ces liens peuvent vous amener vers :

- vers un autre endroit du document
- vers un fichier HTML situé à un emplacement différent sur la machine qui héberge la page
- vers une autre machine
- vers une adresse email

Un lien Hypertexte n'est jamais qu'une balise un peu particulière, la balise anchor <A> mais qui ne sera jamais utilisée seule et dont la balise de fin est </A>.

`<A HREF = " ..... " > Texte </A>`

### Exemples :

```
<a href="http://www.edunet.tn"> Edunet </a>
<a href="file:///lecteur:/répertoire/index.html"> ... </a>
<a href="mailto:adresse_email"> ... </a>
```

### Les signets :

Il est également possible de créer un signet dans une page, c'est-à-dire marquer un endroit précis d'une page pour s'y rendre par un lien hypertexte. Les signets se définissent grâce à l'attribut NAME ou ID.

La syntaxe est la suivante :

```
<balise id="signet"> ... </balise>
```

L'appel d'un signet se fait de la manière suivante :

```
<a href="#signet"> ... </a>
```

Il est ainsi possible de permettre à l'utilisateur de se déplacer au sein d'une même page ou bien d'accéder à une section particulière d'une autre page :

```
<a href="url/nom_du_fichier.html#signet"> ... </a>
```

## 7. Création de formulaire HTML :

Les formulaires interactifs permettent aux auteurs de pages Web de doter leur page web d'éléments interactifs permettant par exemple un dialogue avec les internautes, à la manière des coupons-réponse présents dans certains magazines.

Le lecteur saisit des informations en remplissant des champs ou en cliquant sur des boutons, puis appuie sur un bouton de soumission (submit) pour l'envoyer soit à un URL, c'est-à-dire de façon générale à une adresse e-mail ou à un script de page web dynamique.

### ❖ La balise Form :

Les formulaires sont délimités par la balise <FORM> ... </FORM>, une balise qui permet de regrouper plusieurs éléments de formulaire (boutons, champs de saisie,...) et qui possède les attributs obligatoires suivants :

- **METHOD** indique sous quelle forme seront envoyées les réponses « POST » est la valeur qui correspond à un envoi de données stockées dans le corps de la requête, tandis que « GET » correspond à un envoi des données codées dans l'URL, et séparées de l'adresse du script par un point d'interrogation (pour plus de renseignement sur les méthodes POST et GET, consultez l'article sur le protocole HTTP)
- **ACTION** indique l'adresse d'envoi (script CGI ou adresse email (mailto:adresse\_email@machine))

### ❖ A l'intérieur de la balise FORM...

La balise *FORM* constitue en quelque sorte un conteneur permettant de regrouper des éléments qui vont permettre à l'utilisateur de choisir ou de saisir des données, ensemble de données qui seront envoyées à l'URL indiquée dans l'attribut *ACTION* de la balise *FORM* par la méthode indiquée par l'attribut *METHOD*.

Il est possible d'insérer n'importe quel élément HTML de base dans une balise *FORM* (textes, boutons, tableaux, liens,...) mais il est surtout intéressant d'insérer des éléments interactifs. Ces éléments interactifs sont :

- La balise *INPUT*: un ensemble de boutons et de champs de saisie
- La balise *TEXTAREA*: une zone de saisie
- La balise *SELECT*: une liste à choix multiples

#### La balise INPUT

La balise *INPUT* est la balise essentielle des formulaires, car elle permet de créer un bon nombre d'éléments "interactifs". La syntaxe de cette balise est la suivante :

```
<INPUT type="Nom du champ" value="Valeur par défaut" name="Nom de l'élément">
```

L'attribut *name* est essentiel car il permettra au script CGI de connaître le champ associé à la paire nom/valeur, c'est-à-dire que le nom du champ sera suivi du caractère "=" puis de la valeur entrée par l'utilisateur, ou dans le cas contraire de la valeur par défaut repérée par l'attribut *value*. L'attribut *type* permet de préciser le type d'élément que représente la balise *INPUT*, voici les valeurs que ce champ peut prendre :

- **checkbox**: il s'agit de *cases à cocher* pouvant admettre deux états : *checked* (coché) et *unchecked* (non coché). Lorsque la case est cochée la paire nom/valeur est envoyée au CGI
- **hidden**: il s'agit d'un *champ caché*. Ce champ non visible sur le formulaire permet de préciser un paramètre fixe qui sera envoyé au CGI sous forme de paire nom/valeur
- **file**: il s'agit d'un *champ* permettant à l'utilisateur de préciser l'emplacement d'un fichier qui sera envoyé avec le formulaire. Il faut dans ce cas préciser le type de données pouvant être envoyées grâce à l'attribut *ACCEPT* de la balise *FORM*
- **image**: il s'agit d'un *bouton de soumission personnalisé*, dont l'apparence est l'image située à l'emplacement précisé par son attribut *SRC*
- **password**: il s'agit d'un *champ de saisie*, dans lequel les caractères saisis apparaissent sous forme d'astérisques afin de camoufler la saisie de l'utilisateur
- **radio**: il s'agit d'un *bouton* permettant un choix parmi plusieurs proposés (l'ensemble des boutons radios devant porter le même attribut *name*). La paire nom/valeur du bouton radio sélectionné sera envoyée au CGI. Un attribut *checked* pour un des boutons permet de préciser le bouton sélectionné par défaut
- **reset**: il s'agit d'un *bouton de remise à zéro* permettant uniquement de rétablir l'ensemble des éléments du formulaire à leurs valeurs par défaut
- **submit**: il s'agit du *bouton de soumission* permettant l'envoi du formulaire. Le texte du bouton peut être précisé grâce à l'attribut *value*
- **text**: il s'agit d'un *champ de saisie* permettant la saisie d'une ligne de texte. La taille du champ peut être définie à l'aide de l'attribut *size* et la taille maximale du texte saisi grâce à l'attribut *maxlength*

❖ Attributs de la balise FORM et types d'entrées

Balise	Attribut	Valeur	Résultat	Effet Visuel
<FORM> ... </FORM>	METHOD	POST GET		
	ACTION		envoi à l'adresse indiquée	
	ENCTYPE		spécifie le type de codage utilisé	
<INPUT>	TYPE	submit	effectue l'ACTION dans le balise <FORM>	<input type="submit" value="Envoyer"/>
		text	simple ligne de texte dont la longueur est donnée par l'attribut size	<input type="text"/>
		reset	efface le contenu du formulaire	<input type="reset" value="Rétablir"/>
		radio	bouton radio	<input type="radio"/>
		checkbox	case à cocher	<input type="checkbox"/>
	NAME		Nom	
SIZE		Taille du texte		
<TEXTAREA> ... </TEXTAREA>	NAME		Zone de texte	
	ROWS			
	COLS			
<SELECT> <OPTION> ... </OPTION> </SELECT>	NAME			<input type="text" value="Choix 1"/>
	MULTIPLE		Plusieurs choix possibles	<input type="text" value="Choix 1"/>
<OPTION> ... </OPTION>	SELECTED	Option par défaut		<input type="text" value="Choix 1"/>
	VALUE	Valeur forcée		<input type="text" value="Choix 2"/> <input type="text" value="Choix 3"/>

8. Les cadres :

Grâce à la technologie des frames (en français "cadres") il est désormais possible d'afficher plusieurs pages HTML dans différentes zones (ou cadres).

Afin de créer un site contenant des cadres, il suffit de créer un fichier contenant l'agencement des cadres : ce fichier HTML a pour particularité d'avoir un conteneur <FRAMESET> à la place du jeu de balises <BODY>. C'est cette balise qui définit les cadres par leur dimension en pixels ou en pourcentage (%).

Voyons ceci sur 3 exemples:

- 2 cadres verticaux
- 2 cadres horizontaux
- 2 cadres horizontaux et un vertical

Exemple n°1

```
<FRAMESET COLS="20%,80%">
  <FRAME SRC="frame1.htm" NAME="gauche">
  <FRAME SRC="frame2.htm" NAME="droite">
```

http://fsk.forumgo.com

</FRAMESET>

1	2
---	---

**Exemple n°2**

```
<FRAMESET ROWS="20%,80%">
```

```
<FRAME SRC="frame1.htm" NAME="haut">
```

```
<FRAME SRC="frame2.htm" NAME="bas">
```

```
</FRAMESET>
```

1
2

**Exemple n°3**

```
<FRAMESET COLS="20%,80%">
```

```
<FRAME SRC="frame1.htm" NAME="gauche">
```

```
<FRAMESET ROWS="50%, 50%">
```

```
<FRAME SRC="frame2.htm" NAME="droit_haut">
```

```
<FRAME SRC="frame3.htm" NAME="droit_bas">
```

```
</FRAMESET>
```

1	2
3	

❖ Les attributs de la balise <FRAMESET>

Attribut	Valeur	Action
Rows	pourcentage (entre 1 et 100) valeur en pixels En fixant une seule des valeurs et en donnant à l'autre la valeur *, la valeur s'ajuste automatiquement	Cadre horizontal
Cols	pourcentage (entre 1 et 100) valeur en pixels En fixant une seule des valeurs et en donnant à l'autre la valeur *, la valeur s'ajuste automatiquement	Cadre vertical
frameborder	YES NO	Indique si le cadre a une bordure ou non
border=n	n est une valeur définissant la taille de la bordure	Indique la taille de la bordure
bordercolor	Nom de la couleur Valeur de la couleur en hexadécimal	Indique la couleur de la bordure
Framespacing=n	n est une valeur définissant l'espace entre les cadres	Indique l'espace entre les cadres

❖ Les attributs de la balise <FRAME>

La balise <FRAME> permet de définir un ou plusieurs cadres au sein de la balise <FRAMESET>

Attribut	Valeur	Action
Src	URL	Définit l'emplacement de la page à afficher dans le cadre
Name	"nom"	Définit un nom qui permettra d'afficher un autre document dans le cadre grâce à l'attribut <b>Target</b>
Marginwidth=n	n est un entier spécifiant le nombre de pixels	Taille des marges latérales
Marginheight=n	n est un entier spécifiant le nombre de pixels	Taille des marges du haut et du bas
Frameborder	YES NO	Détermine si les cadres auront ou non une bordure
Border=n	n est un entier spécifiant le nombre de pixels	Taille de l'espace entre les cadres (uniquement pour Netscape)
Noresize	(Aucune)	Interdit à l'utilisateur de redimensionner les cadres (Ce n'est pas la valeur par défaut)
Scrolling	<b>YES</b> <b>NO</b> <b>AUTO</b>	Permet ou non l'affichage d'une barre de défilement ( <b>Auto</b> laisse le navigateur décider de son utilité)



## Chapitre 2 :

### Partie B :

## Pages Web Statiques

### Sous partie B2 :

## Le langage JAVASCRIPT

### I-Introduction au langage Javascript :

#### 1. Pourquoi le script ?

Un script est simple et interprétable par les langages de programmation. Il permet :

- une interactivité aux pages Web et celles-ci deviennent dynamiques
- contrôle des valeurs saisies dans un formulaire
- exécution lors de la survenue d'événement (chargement, utilisation de la souris, etc.)
- scripts du coté clients sont inclus dans les pages Web et exécutable par le navigateur

Les scripts populaires sont: **Javascript**, **VBScript**, **JScript**, etc.

#### 2. Historique du langage Javascript :

**JavaScript** (initialement appelé **LiveScript**) a été développé par Netscape puis a été repris par la firme Sun (qui a aussi développé Java).

Surtout ne pas confondre **Java** et **Javascript/Jscript** bien qu'ils soient tous utilisés pour créer des pages Web évoluées !

**Javascript** est un langage de scripts qui s'incorpore au langage **HTML** et on peut le considérer comme une extension du langage **HTML**. Il est disponible à tout le monde sans licence.

### II-Le formalisme du langage Javascript :

#### 1. Les délimiteurs du JavaScript:

Le code est inséré entre les balises `<script> .. </script>`

Les balises sont insérées dans la partie `<head>` ou `<body>`

```
<html>
<head>
  <title>Le langage JavaScript</title>
</head>
<body>
```

Balises  
HTML

```
<h1>JavaScript Example 1</h1>
<script language ="javascript">
  alert("Bonjour");
</script>
```

Script

```
</body>
</html>
```

- C'est le couple de balises `<Script> ... </Script>` qui permet de repérer l'existence d'un script dans une page Html, mais ... où le placer ? Il peut être placé n'importe où dans la page mais s'il est placé dans l'entête

(<HEAD> ... </HEAD>), ainsi il est exécuté avant même que le navigateur ait commencé à afficher le contenu.

- Lors de l'appel à une fonction dans une page, il est impératif que cette fonction ait été déclarée avant d'être utilisée. La déclaration dans l'entête de la page répond à cette exigence.

#### Remarque :

Lorsqu'un même script doit être utilisé dans des pages Html différentes, afin de faciliter sa maintenance, il est conseillé d'écrire ce script dans un fichier texte séparé (suffixé par exemple .JS pour le Javascript).

L'attribut SRC ajouté à la balise <Script> indiquera le nom du fichier qui contient le script. (Les autres paramètres sont ignorés). La balise </SCRIPT> est alors **absente**.

Exemple :

```
<SCRIPT SRC="MesScripts.js">
```

## 2. Les commentaires :

### Rappel : Les commentaires en Html

```
<!-- commentaire début  
commentaire suite  
commentaire fin  
-->
```

Sur plusieurs lignes, tout ce qui est compris entre <!-- et --> n'est pas interprété par le navigateur. En Html, les commentaires évitent que certains anciens navigateurs affichent les fonctions qui leur sont inconnues comme étant de l'Html. Dans ce cas, ils les ignorent.

### Les commentaires en Javascript

Dans des fonctions Javascript, il existe 2 types de symboles pour désigner du commentaire.

```
// mettez votre commentaire ici
```

sur une même ligne, tout ce qui suit les signes // n'est pas analysé par le compilateur de Javascript

```
/* commentaire début  
commentaire suite  
commentaire fin */
```

Sur plusieurs lignes, tout ce qui est compris entre /\* et \*/ n'est pas analysé par le compilateur. Ceci permet de déterminer des blocs de commentaires.

## III-Les objets Javascript :

### 1. Les objets JavaScript et leurs hiérarchies :

Le Javascript traite les éléments qui s'affichent dans votre navigateur comme des objets, c'est-à-dire des éléments :

- classés selon une hiérarchie pour pouvoir les désigner précisément ;
- auxquels des propriétés sont associées.

#### *a- La hiérarchie des objets d'interface :*

Cette notion semble floue pour l'instant mais voyons voir cela sur un exemple concret : Imaginez un arbre dans un jardin comportant une branche sur laquelle se trouve un nid. On suppose la hiérarchie d'objets est définie comme ceci :

http://www.spa.com/programming

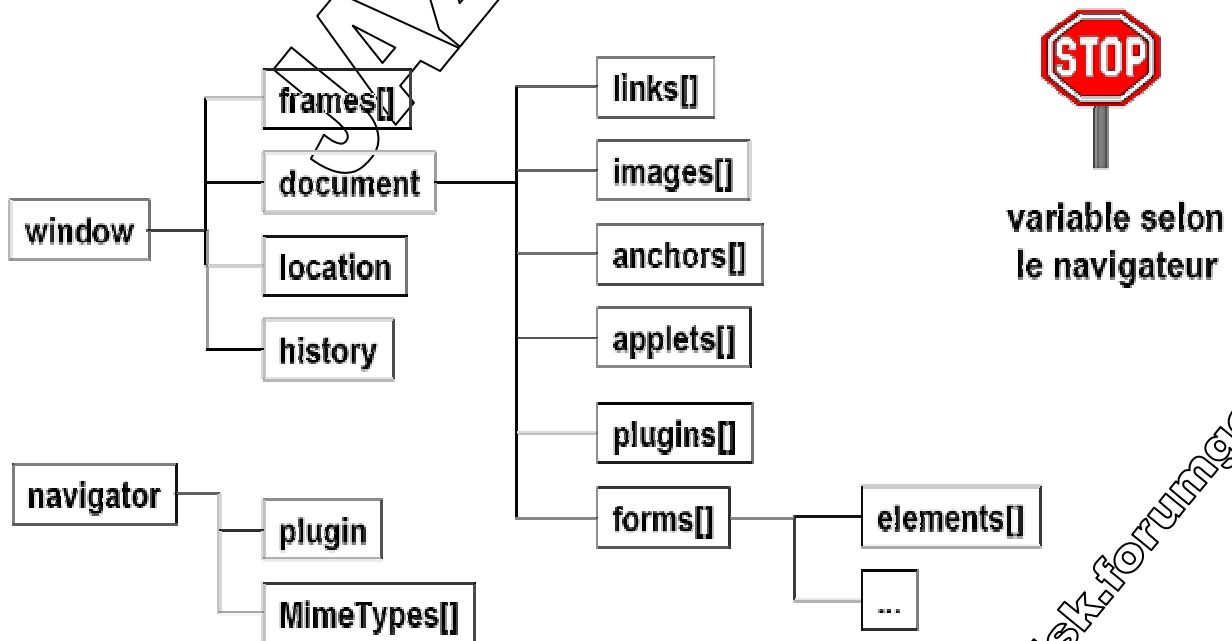
- ▶ jardin
  - arbre
    - branche
      - feuille
      - nid
        - largeur: 20
        - couleur: jaune
        - hauteur: 4
    - tronc
    - racine
  - salade
  - balançoire
    - trapèze
    - corde
    - nid
      - largeur: 15
      - couleur: marron
      - hauteur: 6

Le nid sur l'arbre est donc désigné comme suit :  
**jardin.arbre.branche.nid**

Contrairement au nid situé sur la balançoire :  
**jardin.balançoire.nid**

Pour changer la couleur du nid (dans l'arbre) et le peindre en vert, il suffit d'une commande telle que la suivante :  
**jardin.arbre.branche.nid.couleur= vert;**

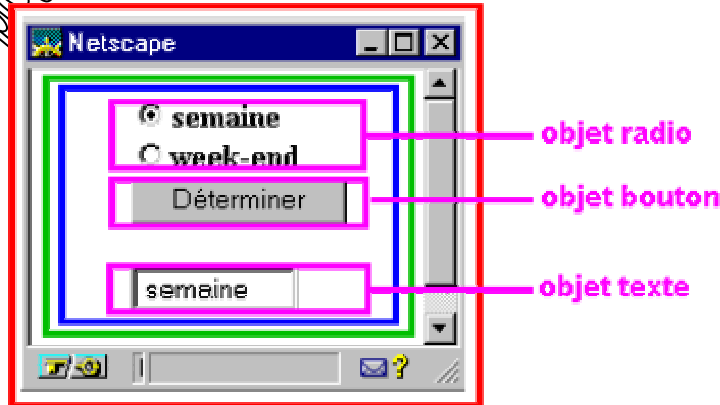
Les objets en Javascript sont représentés de cette manière à ceci près que ce n'est pas un jardin qui est représenté sous forme d'objets mais la fenêtre du navigateur.



variable selon le navigateur

http://flsk.forumgo.com

Javascript divise la page du navigateur en objets, afin de permettre d'accéder à n'importe lequel d'entre-eux et de pouvoir les manipuler par l'intermédiaire de leurs propriétés.



On commence généralement par l'objet le plus grand (celui contenant tous les autres) puis on descend dans la hiérarchie jusqu'à arriver à l'objet voulu.

- L'objet le plus grand est l'objet fenêtre (les objets en javascript ont leur dénomination en anglais, donc dans le cas présent window)
- Dans la fenêtre s'affiche une page, c'est l'objet document
- Cette page peut contenir plusieurs objets, comme des formulaires, des images, etc.

Pour accéder à un objet il est donc nécessaire de parcourir l'arborescence en partant de l'objet le plus grand (l'objet window) vers l'objet à atteindre.

Soient le bouton (appelé checkbox) et le champ de texte suivants :



- Le formulaire (appelé form1 pour le distinguer des autres formulaires de la page) est repéré par le code suivant :  
**window.document.forms["form1"]**
- Le bouton checkbox (baptisé checkbox) est repéré par le code suivant :  
**window.document.forms["form1"].checkbox**
- Le champ de texte (appelé champ\_text) est repéré par le code suivant :  
**window.document.forms["form1"].champ\_text**

Le bouton checkbox a entre autre une propriété checked, qui retourne la valeur 1 si le bouton est coché, 0 dans le cas contraire. Cliquez sur le bouton checkbox pour comprendre ce qu'il est possible de faire grâce à ces propriétés.

#### ***b-Les propriétés des objets :***

Une propriété est un attribut, une caractéristique, une description de l'objet. Par exemple, l'objet volant d'une voiture a comme propriétés qu'il peut être en bois ou en cuir. L'objet livre a comme propriétés son auteur, sa maison d'édition, son titre, son numéro ISBN, etc.

De même les objets Javascript ont des propriétés personnalisées. Dans le cas des boutons radio, une de ses propriétés est, par exemple, sa sélection ou sa non-sélection (checked en anglais).

En Javascript, pour accéder aux propriétés, on utilise la syntaxe :

**nom\_de\_l'objet.nom\_de\_la\_propriété**

Dans le cas du bouton radio "semaine", pour tester la propriété de sélection, on écrira :

**document.form.radio[0].checked**

### c-Les méthodes des objets

Non seulement chaque objet peut avoir des propriétés (valeurs particulières) mais il peut également avoir des fonctions qui lui sont propres, appelées méthodes (ou même comportements). L'appel de la méthode se fait selon la notation

**nom\_de\_l'objet.nom\_de\_la\_méthode()**

#### Exemple :

Il est possible d'écrire du texte dans un document. La méthode s'appelle "write()". Pour afficher du texte dans un document, on écrit :

```
document.write("<CENTER>Ici, le lycée de Korba !</CENTER><HR>Bienvenue !") ;
```

Ce qui donnera dans le navigateur :

Ici, le lycée de Korba !
Bienvenue !

## IV-Les variables :

### 1. Déclaration :

Le Javascript étant très souple au niveau de l'écriture (à double-tranchant car il laisse passer des erreurs...), la déclaration des variables peut se faire de deux façons :

- soit de façon explicite, en faisant précéder la variable du mot clé var qui permet d'indiquer de façon rigoureuse qu'il s'agit d'une variable :

```
var chaine= "bonjour"
```

- soit de façon implicite, en laissant le navigateur déterminer qu'il s'agit d'une déclaration de variable. Pour déclarer implicitement une variable, il suffit d'écrire le nom de la variable suivie du caractère = et de la valeur à affecter :

```
chaine= "bonjour"
```

Même si une déclaration implicite est tout à fait reconnue par le navigateur, il est plus rigoureux de déclarer les variables de façon explicite avec le mot var.

### 2. Portée des variables :

- Variable locale

Une variable déclarée dans une fonction par le mot-clé var aura une portée limitée à cette seule fonction. On ne pourra donc pas utiliser sa valeur ailleurs dans le script. C'est une variable locale à la fonction.

#### Exemple :

```
var lycee ="Lycée de Korba"; // n'est connue que la fonction dans laquelle elle se // trouve
```

- Variable globale

• Si la variable est déclarée sans utiliser le mot var, sa portée sera globale (une fois que la fonction aura été exécutée).

#### Exemple :

```
lycee ="Lycée de Korba"; ; // est connue de toutes les fonctions. Elle pourra être // utilisée dans d'autres fonctions
```

• De même, une variable déclarée tout au début du script, en dehors et avant toute fonction, est toujours globale, qu'elle soit déclarée avec ou sans var.

### 3. Types de données :

En Javascript il n'est pas nécessaire de déclarer le type des variables, contrairement à des langages évolués tels que le langage Pascal ou le C pour lesquels il faut préciser s'il s'agit d'entier (integer), de nombre à virgule flottante (real) ou de caractères (string).



En fait, le JavaScript n'autorise la manipulation que de 4 types de données :

- des nombres: entiers ou à virgules
- des chaînes de caractères (string): une suite de caractères
- des booléens: des variables permettant de vérifier une condition. Les booléens peuvent prendre deux états :
  - true : si le résultat est vrai ;
  - false : lors d'un résultat faux.
- des variables de type null: un mot caractéristique pour indiquer que la variable ne contient aucune donnée.

Déclaration en JavaScript	Type
<pre>var ma = 10.95 ; var indice = -1 ; var i ;</pre>	Numérique décimal Numérique entier Numérique indéfini (type undefined)
<pre>var chaine1 = "Lycée de Korba" ; var chaine2 = "Lycée de \"Korba\" " ; var chaine3= chaine2 ;</pre>	Chaîne de caractères certains caractères spéciaux doivent être précédés de \ (voir Annexe ci-après: Caractères spéciaux)
<pre>var indicateur = false ; var option = true ;</pre>	Booléen (false ou true)
<pre>var maVariable = null ;</pre>	Null – aucune donnée n'est présente (absence de données). Signification différente de 0 (zéro) ou vide ("").

### Caractères spéciaux

Caractère	Code Javascript
Espace arrière	\b
Saut de page	\f
Alinéa (retour ligne)	\n
Retour chariot	\r
Tabulation	\t
Apostrophe	'
Guillemet	"
Antislash	\\

### V-Les opérateurs prédéfinis :

Les opérateurs vont permettre de manipuler les variables (comparaison, affectation, calculs, ...). Il en existe de différentes catégories.

#### 1. Les opérateurs de calcul : \* + - / = %

Ils servent aux calculs : \* (multiplier), + (additionner), - (soustraire), / (diviser), = (affecter), % (prendre le reste de la division = modulo).

**Exemples :** a \* 5.2 ;                      b / (c + 4) ;                      10 % 2 ;

#### 2. Les opérateurs de comparaison : ==, <, >, ...

Ils sont utilisés dans les instructions de conditions (if .... { .. }).

**Exemples :** if (x == 0) signifie si x = 0  
 if (x != 0) signifie si x différent de 0

#### 3. Les opérateurs associatifs : +=, -=, \*=, /=

Ils permettent à la fois le calcul et l'affectation de la valeur calculée à une variable.

**Exemples :** x += y équivalent de x = x + y

#### 4. Les opérateurs logiques : &&, || et !

Ils servent à comparer 2 expressions.

**Exemples :**

```
if ( x > 10 ) && ( y < b ) { ... } //si x > 10 et y < b alors ...
if ( x < 0 ) || ( z == 1 ) { ... } //si x < 0 ou z = 1 alors ...
if ( x != z ) { ... } //si x égale non z 1 alors ...
```

#### 5. Les opérateurs d'incrémentatation :

Ils servent à augmenter (ou diminuer) de 1 la valeur d'une variable. Ils sont très utilisés dans les structures de boucles.

**Exemples :**

incrémentatation d'une valeur :

Dans ce premier exemple la variable a prend la valeur de la variable b, puis b est incrémenté de 1

```
a = b++;
```

si à l'état initial b vaut 10, alors à l'état final a vaut 10 et b vaut 11

Dans ce second exemple la variable b est incrémentée de 1, puis a prend la valeur de la variable b

```
a = ++b;
```

si à l'état initial b vaut 10, alors à l'état final a vaut 11 et b vaut 11

Il est aussi possible d'incrémenter une variable sans l'assigner :

```
a++;
```

- décrémentation d'une valeur :

De même que pour l'incrémentatation, la décrémentation d'une variable donne pour

```
a = b--;
```

avec b valant 5 à l'état initial, alors a vaut 5 et b vaut 4.

```
a = --b;
```

avec b valant 5 à l'état initial, alors a vaut 4 et b vaut 5.

## VI-Les entrées / sorties en JavaScript :

Dans le langage JavaScript les entrées sorties sont effectuées comme suit :

### ➤ L'entrée

Les saisies de données (entrée) vont être réalisées par l'intermédiaire des formulaires ou des méthodes de l'objet **window** suivantes :

- **prompt()** : affiche une fenêtre de dialogue avec une zone texte  
Cette méthode permet à l'utilisateur, de lire une question, spécifiée par la chaîne message, de saisir une valeur et d'appuyer ensuite sur "Ok". On peut spécifier une valeur par défaut à l'aide du deuxième paramètre optionnel initial. La fonction rend la valeur saisie sous la forme d'une chaîne de caractère. Si le programme attend une valeur numérique, il faut la convertir avec des fonctions comme **parseInt()**
- **confirm()** : affiche une fenêtre de dialogue avec boutons 'OK' et 'Cancel'  
Cette méthode permet de poser une question à l'utilisateur, qui répondra en appuyant soit sur "Ok" soit sur "Annuler". Si "Ok" est appuyé, la fonction retourne true, et si "Annuler" est appuyé, elle rend false.

### ➤ La sortie

L'affichage des résultats (sorties) peut s'inscrire dans de nouvelles fenêtres (méthode **alert()** de l'objet **window**) ou dans le document en cours (méthode **write()** de l'objet **document**) ou à l'aide des objets graphiques du formulaire.

- `alert()` : affiche une fenêtre de dialogue  
D'autres méthodes permettent d'ouvrir et de fermer de nouvelles fenêtres ou de changer leur apparence. Une autre méthode très utile est la méthode `setTimeout()`. Elle permet d'exécuter un code JavaScript après un laps de temps donné (indiqué en millisecondes). Par exemple :  
`setTimeout("alert('bonjour');", 500);` fera apparaître la fenêtre de dialogue une demi-seconde après le chargement de la page.
- `write()` : affiche le texte et les balises Html dans la page en cours  
L'utilisation de la méthode `write()` ou `writeln()` de l'objet document, permet d'écrire un message dans la fenêtre du navigateur.

## II- Les structures de contrôle :

### 1. Les structures conditionnelles :

#### a-La structure IF :

##### Syntaxe :

```
If ( condition )
{
    instruction1 si vrai;
}
Else
{
    instruction2 si faux;
}
```

#### Remarque :

Il est tout à fait possible de faire des conditions imbriquées. Pour des facilités de lecture du script, il est conseillé de décaler vers la droite les blocs if internes.

#### b-La structure SWITCH :

##### Syntaxe :

```
switch ( expression )
{
    case v1: instruction_1; break;
    case v2: instruction_2; break;
    ....
    case vn: instruction_n; break;
    default :
        instruction_n+1;
}
```

#### Remarques :

- L'expression peut être une simple variable ou une expression à analyser.
- Plusieurs "case" peuvent être empilés.
- Chaque "case" est terminée par une instruction `break`, pour éviter que le compilateur n'exécute la suite des "case"
- L'instruction "default" est facultative. Elle sert à traiter tous les cas non précisés dans la liste des "case".
- Dans le cas d'une variable alphanumérique, on écrirait : case "L"

## 2. Les structures itératives

### a-La structure FOR

#### Syntaxe :

```
for (expression_initiale ; condition_de_sortie ; expression_de_progression)
{
    instructions;
}
```

### b-La structure DO...WHILE : Faire ... tant que

#### Syntaxe :

```
do
{
    instructions;
}
while (condition)
```

#### Remarque :

Dans une boucle "Faire ... tant que ...", les instructions situées dans la boucle sont exécutées au moins une fois, même si la condition n'est pas remplie.

### c-La structure WHILE : Tant que

#### Syntaxe :

```
while (condition)
{
    instructions;
}
```

#### Remarque :

Dans une boucle "Tant que ...", les instructions situées dans la boucle peuvent ne jamais être exécutées si la condition n'est pas remplie.

## VIII-Les fonctions en JavaScript :

Les fonctions peuvent être assimilées à des boîtes qui regroupent un ensemble d'instructions réutilisables simplement et qui permettent d'obtenir un résultat précis.

- Une fonction peut (mais elle n'est pas obligée) accepter des valeurs (appelées "arguments" ou "paramètres"). Si il y en a plusieurs, elles sont séparées par des **virgules**.
- Elle peut (mais elle n'est pas obligée) renvoyer une **(et une seule !)** valeur issue d'un calcul (instruction **return**)
- Une fonction fait quelque chose d'utile.

#### Syntaxe :

```
function Nom_Fonction (parametres)
{
    Instructions ;
    [return Valeur_Finale; ]
}
```

#### Exemple d'une fonction avec un argument

```
function calcul_surface_cercle (rayon)
{
```

```
/* cette fonction calcule la surface d'un cercle à partir du rayon passé en paramètre */
var surf= Math.PI*rayon * rayon;
return surf; // ou return (surf);
}
```

**Exemple d'une fonction à 2 arguments**

```
function calcul_surface_rectangle (longueur, largeur)
{
// cette fonction calcule la surface d'un rectangle à partir des longueur et largeur passées en
// paramètre */
return longueur*largeur; // calcule la surface et renvoie la valeur
}
```

**IX-La gestion des évènements en JavaScript :**

**1. Généralités :**

C'est la notion d'évènements gérés par du Javascript qui rendent les pages Html animées et attractives. En effet selon l'action de l'internaute, des actions seront déclenchées et la page va s'animer d'une manière ou d'une autre. En bref, c'est la clé de l'interactivité autrement appelée DHTML (Dynamic HTML).

En Html classique, un seul évènement est géré : c'est le clic de la souris sur un lien (balise <a href="lien">... </a>). Javascript va permettre d'en gérer beaucoup d'autres. Les évènements sont associés aux fonctions, aux méthodes et aux formulaires.

**2. Les évènements et les gestionnaires d'évènements**

Les évènements correspondent aux actions réalisées par l'utilisateur. Les fonctions à réaliser selon les évènements seront déclenchées par le gestionnaire d'évènements.

Description	Evénements	S'applique avec	Gestionnaire d'évènement
Quand l'utilisateur clique sur un bouton, un lien ou tout autre élément...	Click	Tout	onClick="fonction()"
Quand l'utilisateur double clique sur un bouton, un lien ou tout autre élément...	DbClick	Tout	onDbClick="fonction()"
Quand la page a fini d'être chargée par le navigateur...	Load	BODY, FRAMESET	onLoad="fonction()"
Quand l'utilisateur quitte la page..	Unload	BODY, FRAMESET	onUnload="fonction()"
Quand le pointeur de la souris survole (sans clic) un lien ou tout autre élément...	MouseOver	Tout	onMouseOver="fonction()"
Quand le pointeur de la souris quitte un lien ou tout autre élément... A partir de Javascript 1.1 (donc pas sous IE 3.0 et Netscape 2).	MouseOut	Tout	onMouseOut="fonction()"
Quand un élément d'un formulaire devient la zone d'entrée active. On dit qu'il reçoit le focus.	Focus	LABEL, INPUT, SELECT, TEXTAREA, BUTTON	onFocus="fonction()"
Quand un élément d'un	Blur	LABEL,	onBlur="fonction()"

<http://www.forumgo.com>



formulaire a perdu le focus (n'est plus l'élément actif). En général, cela se produit quand l'utilisateur clique en dehors de l'élément actif.		<b>INPUT, SELECT, TEXTAREA, BUTTON</b>	
Quand la valeur d'un champ de formulaire est modifiée.	<b>Change</b>	<b>INPUT, TEXTAREA</b>	<b>onChange="fonction()"</b>
Quand l'utilisateur sélectionne tout ou partie d'une valeur d'un champ dans un élément de formulaire.	<b>Select</b>	<b>INPUT, TEXTAREA</b>	<b>onSelect="fonction()"</b>
Quand l'utilisateur clique sur un bouton "Submit" pour envoyer un formulaire au serveur...	<b>Submit</b>	<b>FORM</b>	<b>onSubmit="fonction()"</b>

**Exemples : "Click" :**

```
onClick = "alert ('Vous avez cliqué sur le bouton')"
```

Au clic de l'utilisateur, une boîte d'alerte s'ouvre avec le message indiqué.

**Exemples : "Load - Unload"**

Ces 2 événements sont en général appelés dans les balises <BODY> ou <FRAMESET>.

```
<BODY onLoad = "alert ('Bienvenue !)" onUnload="alert ('A bientôt !)">
```

Quand la page est chargée, l'internaute verra une boîte d'alerte "Bienvenue !" et quand il la quittera, il verra "A bientôt !".

**Exemples : "mouseover - mouseout" :**

```
<IMG SRC="logo.jpg" onMouseover="alert ('Le curseur de la souris est passée sur le logo') ">
```

Cet événement est pratique pour afficher des explications soit dans la barre de statut ou pour afficher un texte qui correspond à la zone survolée...

L'événement onmouseout est généralement associé à un onMouseover. Il se produit lorsque le pointeur quitte la zone "sensible" (lien ou image).

On peut aussi coder en faisant intervenir la fonction sur un lien hypertexte :

```
<A HREF="#" onMouseover="alert ('Le curseur de la souris est passée sur le logo') > lien </A>
```

Dans ce cas, l'internaute peut cliquer sur le lien. Pour gérer correctement ce "clic", il est conseillé de réaliser un lien sur "#" qui aura pour seule conséquence, de réafficher la page en cours.

Autre écriture quasiment équivalente à la précédente :

```
<A HREF="javascript:void(0)" onMouseover="alert ('Le curseur de la souris est passée sur le logo') > lien </A>
```

La différence vient de l'appel à une fonction Javascript qui ne fait .... rien. En effet, l'opérateur void avec le paramètre 0 indique qu'aucune fonction n'est exécutée quand l'internaute clique sur l'image.

C'est grâce à ces 2 événements mouseover et mouseout que l'on peut gérer l'inversion d'images (une image quand le curseur de la souris passe sur le lien, une autre image quand il quitte ce lien).

**Exemples : "focus - blur - change" :**

```
<INPUT type="text" onFocus="calcul1()" onChange="verif ()" onBlur="calcul2()" >
```

La fonction calcul1() est appelée dès que le curseur se trouve dans cette zone de texte. Si la valeur de la zone a été modifiée, la fonction verif() est appelée, puis quand le curseur a quitté cette zone, c'est la fonction calcul2() qui est exécutée.

**Exemples :** "select"

```
<INPUT type="text" onSelect="copie();" >
```

Si tout ou partie de la valeur est sélectionnée (mise en surbrillance) dans une zone Text ou Textarea, la fonction copie() est appelée.

## X-1. formulaires en JavaScript :

**Généralités :**

Les formulaires sont utilisés dans presque tous les sites. Il s'agit d'un ensemble de champs (boutons radio, cases à cocher, lignes de texte, champs de texte...) qui se termine par un bouton de validation. Les formulaires permettent aux visiteurs de participer de façon active au site qu'ils visitent.

### 2. Accéder à un formulaire :

Elle se fait par les balises <FORM Name="nom\_formulaire" ...> et <FORM>. En Javascript, l'attribut NAME a toute son importance pour désigner le chemin complet de ses contrôles.

Le formulaire étant un élément de l'objet document, pour accéder à "nom\_formulaire", il faut écrire :

```
document.forms["nom_formulaire"]
ou
document.forms[0]
ou
document.nom_formulaire
```

**Remarque :** Les attributs ACTION et METHOD sont facultatifs si on ne fait pas appel au serveur.

### 3. Accéder et manipuler à un élément du formulaire :

Pour accéder à une zone de texte qui serait baptisée "zt" dans le formulaire précédent par l'instruction Html suivante :

```
<INPUT TYPE="text" Name="zt">
```

On écrit :

```
document.forms["nom_formulaire"].elements["zt"]
ou
document.forms["nom_formulaire"].elements[0]
ou
document.forms["nom_formulaire"].zt
```

Ensuite, on peut manipuler cet objet.

**Exemple :** placer la valeur "Enitab" dans la zone de texte "zt"

```
document.forms["nom_formulaire"].elements["zt"].value="Enitab"
```

**Autre exemple :** donner le focus à l'élément "zt"

```
document.forms["nom_formulaire"].elements["zt"].focus()
```

**a-Le contrôle "zone de texte" :** <INPUT TYPE="text">

La zone de texte est l'élément d'entrée/sortie par excellence de Javascript. La syntaxe Html est :

```
<INPUT TYPE="text" Name="nom" SIZE=x MAXLENGTH=y>
```

pour un champ de saisie d'une seule ligne, de longueur x et de longueur maximale de y.

L'objet zone de texte possède trois propriétés :

Propriété	Description
name	indique le nom du contrôle par lequel on pourra accéder.
Default Value	indique la valeur par défaut qui sera affichée dans la zone de texte.
value	indique la valeur en cours de la zone de texte. Soit celle tapée par l'utilisateur ou si celui-ci n'a rien tapé, la valeur par défaut.

**b-Les radio-boutons et les cases à cocher : <INPUT TYPE="radio | checkbox">**

Généralement, les radio-boutons sont utilisés pour sélectionner une option et une seule parmi plusieurs. Les cases à cocher servent à indiquer si l'option est choisie ou non. Leurs syntaxes respectives en Html sont :

**<INPUT TYPE="radio" Name="nom" >**

**<INPUT TYPE="checkbox" Name="nom" >**

On ajoute **CHECKED**, cela signifie que le contrôle s'affiche dans l'état "sélectionné".

Propriété	Description
name	indique le nom du contrôle. Dans le cas des radio-boutons, tous les boutons portent le même nom. Chacun est alors repéré par un numéro d'index.
index	l'index ou le rang du bouton radio en commençant par 0.
checked	indique l'état en cours de l'élément radio.
defaultchecked	indique l'état du bouton sélectionné par défaut.
value	indique la valeur de l'élément radio.
type	type du champ text, button, radio, checkbox, submit, reset

**Exemple avec des radio-boutons:**

<HTML>

<HEAD>

<SCRIPT language="javascript">

function choix (nom\_form)

{

if (nom\_form.choix3a[0].checked) { alert("Option choisie : " + nom\_form.choix3a[0].value) };

if (nom\_form.choix3a[1].checked) { alert("Option choisie : " + nom\_form.choix3a[1].value) };

if (nom\_form.choix3a[2].checked) { alert("Option choisie : " + nom\_form.choix3a[2].value) };

}

</SCRIPT>

</HEAD>

<BODY>

<H2>Options de 3<sup>ème</sup> années Choisies : </H2>

<FORM NAME="form3a">

<INPUT TYPE="radio" NAME="choix3a" VALUE="Viti">Viticulture<BR>

<INPUT TYPE="radio" NAME="choix3a" VALUE="Eco">Economie<BR>

<INPUT TYPE="radio" NAME="choix3a" VALUE="Foret">Forêt<BR>

<INPUT TYPE="button" VALUE="Votre choix de 3<sup>ème</sup> année ?" onClick="choix (form3a)">

</FORM>

</BODY>

</HTML>

Dans le formulaire form3a, on a créé 3 radio-boutons auxquels on a donné le même nom. Le bouton de type "Button" déclenche la fonction choix(). Elle teste quel radio-bouton est coché : soit choix3a[0], choix3a[1], choix3a[2]. Si la propriété checked (=sélectionné) est vraie (true), alors un message affiche de choix sélectionné dans une boîte d'alerte.

**c-Liste de sélection ou liste déroulante : <SELECT> <OPTION>**

Ce contrôle permet de proposer diverses options sous la forme d'une liste déroulante. L'utilisateur y fait son choix qui reste alors affiché. La syntaxe Html est :

```
<SELECT Name = "nom" >
  <OPTION value = "valeur1">La 1ère valeur
  <OPTION value = "valeur2">La 2ème valeur
</SELECT>
```

Propriété	Description
name	indique le nom de la liste déroulante.
length	indique le nombre d'éléments de la liste. S'il est indiqué dans le tag <SELECT>, tous les éléments de la liste seront affichés. Si vous ne l'indiquez pas un seul apparaîtra dans la boîte de la liste déroulante.
selectedIndex	indique le rang à partir de 0 de l'élément de la liste qui a été sélectionné par l'utilisateur.
defaultselected	indique l'élément de la liste sélectionné par défaut. C'est lui qui apparaît alors dans la petite boîte.

Exemple : reprend l'exemple donné dans la partie "radio-boutons"

```
<HTML>
<HEAD>
<Script language="javascript">
function choix(nom_form)
{
alert("Elément " + (nom_form.choix3a.selectedIndex + 1) + " choisi : " + "<BR>" + "sa valeur (value=) est "
+ nom_form.choix3a[nom_form.choix3a.selectedIndex].value + "<BR>sa valeur affichée est : " +
nom_form.choix3a[nom_form.choix3a.selectedIndex].text);
}
</SCRIPT>
</HEAD>
<BODY>
<H2>Options de 3ème années choisies.</H2>
<FORM NAME="form3a">
<SELECT NAME="choix3a">
  <OPTION VALUE="Viti"> Viticulture
  <OPTION VALUE="Eco"> Economie
  <OPTION VALUE="Foret"> Forêt
</SELECT>
<INPUT TYPE="button" VALUE=" Votre choix de 3ème année ?" onClick="choix(form3a)"> </FORM>
</BODY>
</HTML>
```

Dans le formulaire form3a, on a créé une liste de sélection (balise <SELECT></SELECT>). Les différents éléments de la liste sont définis par les balises <OPTION>. Le bouton de type "Button" déclenche la fonction choix(). Elle teste quelle option a été sélectionnée. La propriété **selectedIndex** renvoie l'index de l'élément sélectionné, et comme il commence à 0, il ne faut pas oublier d'ajouter 1 pour retrouver le juste rang de l'élément affiché dans la boîte d'alerte. Le message termine par la valeur de l'élément choisi (value). A l'utilisation : si l'utilisateur choisit dans la liste déroulante, l'option "Economie", à l'écran s'affiche la boîte d'alerte suivante :

```
Elément 2 choisi :
sa valeur (value=) est Eco
sa valeur affichée est : Economie
```

**Gestion dynamique d'une liste de sélection**

Accès à la ième option de la liste nommée « liste ».

```
var a ;
```

```
a=liste.options[i].text
```

**Création d'une nouvelle option dans la liste.**

Il faut avant tout créer une nouvelle option :

```
var o ;
```

```
o=new Option(<libellé>,<valeur>);
```

Le paramètre <libellé> détermine le texte qui apparaîtra dans la liste.

Ensuite, il faut ajouter cette nouvelle option en fin de liste (ici la liste de sélection se nomme « liste ») :

```
liste.options[liste.options.length]=o;
```

Suppression de la ième option de la liste :

```
list.options[i]=null
```

Cas particulier, la suppression de toutes les options de la liste :

```
liste.options.length=0 ;
```

**d-Le contrôle textarea : <TEXTAREA> :**

Il s'agit en fait d'une zone de texte de plusieurs lignes. Sa syntaxe Html est :

```
<TEXTAREA NAME="nom" ROWS=x COLS=y> ... </TEXTAREA>
```

où ROWS=x représente le nombre de lignes et COLS=y représente le nombre de colonnes.

Propriété	Description
name	indique le nom du contrôle par lequel on pourra accéder.
defaultvalue	indique la valeur par défaut qui sera affichée dans la zone de texte.
value	indique la valeur en cours de la zone de texte. Soit celle tapée par l'utilisateur ou si celui-ci n'a rien tapé, la valeur par défaut.

A ces propriétés, il faut ajouter onFocus(), onBlur(), onSelect() et onChange().

**e-Les contrôle Reset et Submit : <INPUT TYPE="reset | submit">**

Reset permet d'annuler les modifications apportées à tous les contrôles d'un formulaire et de revenir aux valeurs par défaut. Submit valide le formulaire et envoie les données saisies à l'URL désigné dans l'attribut ACTION de la balise <FORM>. Les syntaxes Html sont respectivement :

```
<INPUT TYPE="reset" NAME="nom" VALUE "texte">
```

```
<INPUT TYPE="submit" NAME="nom" VALUE "texte">
```

où VALUE donne le texte qui sera lisible sur du bouton.

Une seule méthode est associée aux 2 contrôles. C'est la méthode onClick(). Elle peut servir, par exemple, pour faire afficher une autre valeur que celle par défaut.

**f-Le contrôle Hidden (caché) : <INPUT TYPE="hidden">**

Ce contrôle permet d'entrer dans le script des données qui n'apparaîtront pas à l'écran, donc cachés. Sa syntaxe Html est :

```
<INPUT TYPE="hidden" NAME="nom" VALUE "les données cachées">
```



**XI-Annexe :**

**1. Méthodes sur les chaînes de caractères :**

Méthode	Description
<b>Chaine.anchor</b> ("nom_a_donner");	Transforme le texte <i>Chaine</i> en <b>ancrage HTML</b> .
<b>Chaine.big()</b>	Augmente la taille de la police.
<b>Chaine.blink()</b>	Transforme la chaîne en texte clignotant.
<b>Chaine.bold()</b>	Met le texte en <b>gras</b> (balise <B>).
<b>Chaine.charAt</b> (position)	Retourne le caractère situé à la position donnée en paramètre
<b>Chaine.charCodeAt</b> (position)	Renvoie le code <i>Unicode</i> du caractère situé à la position donnée en paramètre
<b>concat</b> (chaîne1, chaîne2[, ...])	Permet de concaténer les chaînes passées en paramètre, c'est-à-dire de les joindre bout à bout.
<b>Chaine.fixed()</b>	Transforme la Chaine en caractères de police fixe (balise <TT>)
<b>Chaine.fontcolor</b> (couleur)	Modifie la couleur du texte (admet comme argument la couleur en hexadécimal ou en valeur littérale)
<b>Chaine.fontSize</b> (Size)	Modifie la taille de la police, en affectant la valeur passée en paramètre
<b>Chaine.fromCharCode</b> (code1[, code2, ...])	Renvoie une chaîne de caractères composée de caractères correspondant au(x) code(s) <i>Unicode</i> donné(s) en paramètre.
<b>Chaine.indexOf</b> (sous-chaîne, position)	Retourne la position d'une sous-chaîne (lettre ou groupe de lettres) dans une chaîne de caractère, en effectuant la recherche de gauche à droite, à partir de la position spécifiée en paramètre.
<b>Chaine italics()</b>	Transforme le texte en <i>italique</i> (balise <I>)
<b>Chaine.lastIndexOf</b> (sous-chaîne, position)	La méthode est similaire à <i>indexOf()</i> , à la différence que la recherche se fait de droite à gauche : Retourne la position d'une sous-chaîne (lettre ou groupe de lettres) dans une chaîne de caractère, en effectuant la recherche de droite à gauche, à partir de la position spécifiée en paramètre.
<b>Chaine.link</b> (URL)	Transforme le texte en <b>hypertexte</b> (balise <A href>)
<b>Chaine.small()</b>	Diminue la taille de la police
<b>Chaine.strike()</b>	Transforme le texte en <del>texte barré</del> (balise <strike>)
<b>Chaine.sub()</b>	Transforme le texte en <sub>indice</sub> (balise <sub>)
<b>Chaine.substr</b> (position1, longueur)	La méthode retourne une sous-chaîne commençant à l'index dont la position est donnée en argument et de la longueur donnée en paramètre.
<b>Chaine.substring</b> (position1, position2)	La méthode retourne la sous-chaîne (lettre ou groupe de lettres) comprise entre les positions 1 et 2 données en paramètre.
<b>Chaine.sup()</b>	Transforme le texte en <sup>exposant</sup> (balise <sup>).
<b>Chaine.toLowerCase()</b>	Convertit tous les caractères d'une chaîne en minuscule.

<b>Chaine.toSource()</b>	Renvoie le code source de création de l'objet.
<b>Chaine.toUpperCase()</b>	Convertit tous les caractères d'une chaîne en majuscule.
<b>Chaine.valueOf()</b>	Renvoie la valeur de l'objet String.

**2. Méthodes sur les dates :**

Méthode	Description	Type de valeurs retournées
<b>getDate()</b>	Permet de récupérer la valeur du jour du mois	L'objet retourné est un entier (entre 1 et 31) qui correspond au jour du mois :
<b>getDay()</b>	Permet de récupérer la valeur du jour de la semaine pour la date spécifiée	L'objet retourné est un entier qui correspond au jour de la semaine :  0: dimanche 1: lundi ...
<b>getFullYear()</b>	Permet de récupérer la valeur de l'année sur 4 chiffres pour la date passée en paramètre	L'objet retourné est un entier qui correspond à l'année (XXXX) : 2008
<b>getHours()</b>	Permet de récupérer la valeur de l'heure	L'objet retourné est un entier (entre 0 et 23) qui correspond à l'objet Date.
<b>getMilliseconds()</b>	Permet de récupérer le nombre de millisecondes	L'objet retourné est un entier (entre 0 et 999) qui correspond aux millisecondes de l'objet passé en paramètre.
<b>getMinutes()</b>	Permet de récupérer la valeur des minutes	L'objet retourné est un entier (entre 0 et 59) qui correspond aux minutes de l'objet Date.
<b>getMonth()</b>	Permet de récupérer le numéro du mois	L'objet retourné est un entier (entre 0 et 11) qui correspond au mois :  0: janvier 1: février ...
<b>getSeconds()</b>	Permet de récupérer le nombre de secondes	L'objet retourné est un entier (entre 0 et 59) qui correspond aux secondes de l'objet passé en paramètre.
<b>getTime()</b>	Permet de récupérer le nombre de millisecondes depuis le 1 <sup>er</sup> janvier 1970	L'objet retourné est un entier. Cette méthode est très utile pour convertir des dates, soustraire ou ajouter deux dates, etc.
<b>getTimezoneOffset()</b>	Retourne la différence entre l'heure locale et l'heure GMT (Greenwich Mean Time)	L'objet retourné est un entier, il représente le nombre de <b>minutes</b> de décalage
<b>getYear()</b>	Permet de récupérer la valeur de l'année sur 2 chiffres pour l'objet Date.	L'objet retourné est un entier qui correspond à l'année (XX) :

## 3. Méthodes sur les tableaux

Méthode	Description
<b>concat</b> (tab1, tab2[, tab3, ...])	Cette méthode permet de concaténer plusieurs tableaux, c'est-à-dire de créer un tableau à partir des différents tableaux passés en paramètre.
<b>join</b> (tableau) ou <b>Tableau.join()</b>	Cette méthode renvoie une chaîne de caractères contenant tous les éléments du tableau.
<b>pop</b> (tableau) ou <b>Tableau.pop()</b>	Cette méthode supprime le dernier élément du tableau et retourne sa valeur.
<b>Tableau.push</b> (valeur1[, valeur2, ...])	Cette méthode ajoute un ou plusieurs éléments au tableau.
<b>Tableau.reverse()</b>	Cette méthode inverse l'ordre des éléments du tableau.
<b>Tableau.shift()</b>	Cette méthode supprime le premier élément du tableau.
<b>Tableau.slice()</b>	Cette méthode renvoie un tableau contenant une partie (extraction) des éléments d'un tableau.
<b>Tableau.splice()</b>	Cette méthode ajoute/retire des éléments d'un tableau.
<b>Tableau.sort()</b>	Cette méthode permet de trier les éléments d'un tableau.
<b>Tableau.unshift</b> (valeur1[, valeur2, ...])	Cette méthode renvoie le code source qui a permis de créer l'objet <i>Array</i> .
<b>Tableau.toString()</b>	Cette méthode renvoie la chaîne de caractères correspond à l'instruction qui a permis de créer l'objet <i>Array</i> .
<b>Tableau.unshift()</b>	Cette méthode permet d'ajouter un ou plusieurs élément au début du tableau.
<b>Tableau.valueOf</b>	Cette méthode retourne tout simplement la valeur de l'objet <i>Array</i> auquel elle fait référence.

## Chapitre 2 :

### Partie C :

# Pages Web Dynamiques

## I-Introduction :

### 1. Qu'est ce qu'un site web dynamique ?

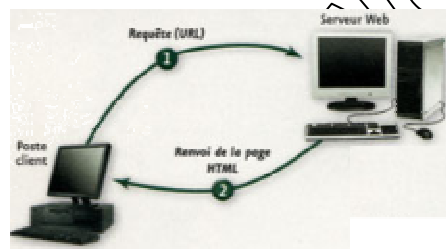
Un site Web dynamique est un site Web dont les pages peuvent être générées dynamiquement, soit à la demande. On parle aussi par opposition de site Web statique.

Le contenu peut être obtenu en combinant l'utilisation d'un langage de scripts ou de programmation et une base de données. Dans le monde des logiciels libres, il s'agit souvent de PHP pour le langage et MySQL pour la base de données.

### 2. Principe de fonctionnement :

#### a. Site Web statique :

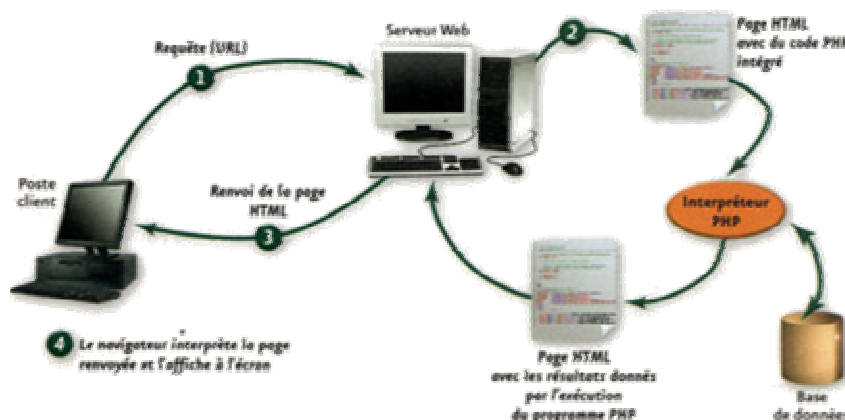
Un site Web statique est un site dont les données sont inchangeables. Les pages sont préparées et enregistrées sur le serveur et accessible seulement en lecture par un client. Leurs contenus ne changent ni en fonction du demandeur ni en fonction d'autres paramètres éventuellement inclus dans la requête adressée au serveur. Toujours le même résultat.



#### b. Site Web dynamique :

Le principe du site Web dynamique est le suivant:

- L'internaute envoie des formulaires au serveur Web.
- Le serveur web compile les pages web et envoie les requêtes au SGBDR.
- Le serveur SGBDR interroge la base à l'aide des requêtes et retourne les résultats de cette interrogation.
- Le serveur SGBDR formule les données et les retourne au serveur Web, qui a son tour prépare la page web et l'envoie au client. Le serveur Web renvoie un fichier ne contenant que du HTML au navigateur qui l'interprète et l'affiche.



PHP n'est pas un langage compilé, c'est un langage interprété par le serveur : le serveur lit le code PHP, le transforme et génère la page HTML. Pour fonctionner, il a donc besoin d'un serveur web. De ce fait une plateforme minimale de base pour l'exécution d'un site web développé en PHP comprend :

- l'interpréteur PHP (serveur PHP)
- un serveur web (Apache, IIS, ...)

Lorsqu'un visiteur demande à consulter une page Web, son navigateur envoie une requête à un serveur HTTP. Si la page contient du code PHP, l'interpréteur PHP du serveur le traite et renvoie du code généré (HTML).

De ce fait le code PHP n'est jamais visible sur la page finale consultée par le client. Ainsi en éditant le source de la page on n'y trouvera que du code HTML.

Avantage du traitement côté serveur

- réduction du temps de téléchargement puisque le client ne reçoit qu'une simple page HTML (=>diminution du trafic réseau)
- absence de problème de compatibilité des navigateurs
- offrir au client des données qui sont dans une base
- le code ne peut être vu par le client.

**II-Le formalisme du langage Php :**

**1. Les délimiteurs du Php :**

Le code est inséré entre les balises <script> ... </script>

Les balises sont insérées dans la partie <head> ou <body>

```

<html>
<head>
  <title>Le langage Php</title>
</head>
<body>
  <h1>Php Example 1</h1>
  <script language ="Php">
    Echo("Bonjour");
  </script>
</body>
</html>
  
```

Diagram illustrating the mapping between HTML tags and PHP code:

- HTML:** <html>, <head>, <title>Le langage Php</title>, </head>, <body>, <h1>Php Example 1</h1>, <script language ="Php">, Echo("Bonjour");, </script>, </body>, </html>
- Script:** <?php, Echo("Bonjour");, <?>

- C'est le couple de balises <Script> ... </Script> qui permet de repérer l'existence d'un script dans une page Html, mais ... où le placer ? Il peut être placé n'importe où dans la page mais s'il est placé dans l'entête (<HEAD> ... </HEAD>), alors il est exécuté avant même que le navigateur ait commencé à afficher le contenu.

- Lors de l'appel à une fonction dans une page, il est impératif que cette fonction ait été déclarée avant d'être utilisée. La déclaration dans l'entête de la page répond à cette exigence.

**Remarque :**

Lorsqu'un même script doit être utilisé dans des pages différentes, afin de faciliter sa maintenance, il est conseillé d'écrire ce script dans un fichier séparé.

PHP possède 2 fonctions qui concernent l'inclusion de fichiers externes :

<pre>&lt;? require ("MesScripts.php"); ?&gt;</pre>	<p>A l'exécution, cette instruction est remplacée par le contenu de "MesScripts.php". Elle ne peut pas être utilisée dans une boucle</p>
<pre>&lt;? for (\$i = 1 ; \$i &lt; 3 ; \$i++) { include ("MesScripts" . \$i . ".php"); } ?&gt;</pre>	<p>Permet d'inclure un fichier ou un autre selon la valeur de \$i. Avec un "require", ce serait toujours le 1er fichier qui serait exécuté.</p>



**N.B :** Les fichiers externes sont recherchés dans le répertoire spécifié dans la directive "include\_path" du fichier php.ini (cf : configuration du Php)

## 2. Les commentaires

### Les commentaires en Php :

Dans des fonctions Php, il existe 2 types de symboles pour désigner du commentaire.

// mettez votre commentaire ici | sur une même ligne, tout ce qui suit les signes // ou # n'est pas analysé par l'interpréteur de Php

/\* commentaire début  
commentaire suite  
commentaire fin \*/

Sur plusieurs lignes, tout ce qui est compris entre /\* et \*/ n'est pas analysé par le compilateur. Ceci permet de déterminer des blocs de commentaires.

## III-Le langage Php :

### I. Présentation :

PHP (abréviation de **P**ersonal Home Page **H**ypertext **P**reprocessor), est un langage de scripts qui s'intègre aux pages Html et qui permet de réaliser des **pages dynamiques**.


- Il s'exécute sur le serveur et permet d'accéder facilement aux bases de données.
- C'est un produit "Open Source" c'est-à-dire que le code est accessible à tout développeur.
- Il est gratuit. Combiné au système d'exploitation Linux, au serveur Apache et à la base de données MySQL (eux-mêmes gratuits), il permet de créer des sites Web à des coûts très réduits.

### 2. Environnement de développement :

PHP travaille sur de nombreuses plates-formes telles que Unix, Linux et Windows sur lesquelles il faut installer un serveur web.

- Un serveur web qui peut être Apache (Unix, Linux, Win NT, Win2K), mais aussi IIS 3/4/5, et même PWS (Personal Web Server).
- Pour écrire des scripts PHP, un simple éditeur de texte tel le Bloc-notes de Windows est suffisant. Toutefois, de nombreux logiciels gratuits (freeware) ou non, en rendent l'écriture plus agréable (aide intégrée, colorisation des mots-clés, ...)
- Pour exécuter les scripts, il faut un navigateur tel que Internet Explorer (IE), Netscape Navigator (NN), Opéra, ...
- Eventuellement, une base de données telle que MySQL.

**EasyPHP** est un utilitaire qui installe et configure automatiquement un environnement de travail complet sur les plates-formes win9x/NT/2000/Me : serveur "Apache, interpréteur PHP, base de données MySQL, ensemble de scripts PHP3 permettant de gérer des bases par le Web. C'est dans cette configuration que seront donnés tous les exemples de ce cours.

Une fois **EasyPHP** démarré, une icône  se place dans la barre des tâches à côté de l'horloge. Un clic droit permet d'accéder à différents menus :

- Fichier Log : renvoie aux erreurs générées par Apache et MySQL
- Configuration : donne accès aux différentes configurations d'EasyPHP
- Web local : ouvre la page <http://localhost/>
- Démarrer/Arrêter : démarre/arrête Apache et MySQL
- Quitter : ferme EasyPHP

### 3. Syntaxe de base du langage Php :

- PHP ressemble aux langages C, C++ et Javascript : à savoir
  - ; à la fin de chaque ligne d'instructions.
  - {...} pour encadrer un bloc d'instructions
  - les opérateurs de comparaison et d'affectation sont les mêmes (&&, ||, ==, ...)



- les symboles des commentaires // et /\* ... \*/
- toute une série de mots réservés qui correspondent à des mots-clés du langage.

- Répertoire par défaut des pages Web
  - Le répertoire mis par défaut à l'installation de EasyPhp est : C:\Program files\EasyPHP\www
- Une page en PHP sera généralement sauvegardée en lui donnant une des extensions suivantes : php, php3, php4. Ce n'est pas une règle absolue, mais ceci correspond à la configuration d'EasyPHP.
- Le script Php s'intègre directement au code HTML et commence par <? (ou <?php) et se termine par

**4. Les structures de données :**

**a. Les constantes :**

➤ Déclaration et test d'existence

Syntaxe en PHP	Type
<pre>define ("Lycee", "Lycée Secondaire de Korba"); define ("PI", 3.14); define ("font", "&lt;FONT size=4 color=#FF0000&gt;&lt;B&gt;")</pre>	Déclaration de Constantes la variable "Ecole" vaut "Enita Bordeaux" La variable "PI" vaut "3.14" La variable "font" contient la chaîne Html
<pre>defined ("Lycee") defined ("Pi")</pre>	Test d'existence renvoi 1 (VRAI) car la constante existe renvoi 0 (FAUX) car la constante n'existe pas (Rappel : Pi est différent de PI)

**Remarque :** Un nom de constante, contrairement à un nom de variable, ne débute pas par un signe \$

**Utilisation :**

Les constantes peuvent être utilisées à la place de leurs valeurs.

Exemple : `echo ("<HR>" . font . ecole);` // le signe . signifie que les variables seront concaténées.

Le résultat sera :

**Lycée Secondaire de Korba** (écrit en en taille 4, en caractères gras rouges et précédé d'une ligne horizontale)

**b. Les variables :**

Chaque variable Php est précédée du signe \$ et reçoit une valeur grâce au symbole d'affectation =. Contrairement à certains langages (Java par exemple), Php n'impose pas de déclaration explicite des variables avant de les utiliser. Toutefois, pour des raisons de cohérence et de clarté, il est une bonne habitude de déclarer les variables et de les faire suivre d'un commentaire qui indiquera leur rôle dans le programme.

Déclarer une variable

Cela signifie réserver et nommer une zone mémoire qui correspond au type de données voulu (entier, alphanumérique, ...).

Exemples : \$ecole \$i \$nbre

En php, il est conseillé de déclarer ses variables ainsi :

```
settype ($i, "integer"); // type entier - indice d'un tableau
settype ($ecole , "string"); // chaîne de caractères- nom école
settype ($nb, "double"); // nbre réel – résultat du calcul
settype ($tb, "array"); // type tableau –liste départements
settype ($obj, "object") // type objet
settype ($cls, "class"); // type classe
settype ($x , "unknown type"); // type inconnu
```

Fonctions utiles pour les variables

<http://fsk.forumgo.com>

Fonctions	Commentaires	Exemple
<b>gettype (\$nom_var)</b>	détermine le type de données de la variable ("integer", "double", "string", "array", "object", "class" ou "unknown type")	\$x = 2.3; echo ( gettype(\$x) ); // double
<b>settype (\$nom_var, "type")</b>	définit explicitement le type de variable	\$x = 2.3; settype (\$x, "integer"); echo \$x; // \$x=2
<b>isset (\$nom_var)</b>	sert à savoir si une variable possède une valeur (true ou false)	\$x = 2.3; \$val = isset (\$x) ; // true
<b>unset (\$nom_var)</b>	détruit une variable	unset (\$x); // isset(\$x) donne false
<b>empty (\$nom_var)</b>	renvoie true si la variable est vide ou vaut 0, sinon renvoie false	\$x = 2.3; \$vide = empty (\$x) ; // false
<b>is_int (\$nom_var)</b> <b>is_integer (\$nom_var)</b> <b>is_long(\$nom_var)</b>	détermine si la variable est un entier	\$x = 2.3; \$entier = is_int (\$x); // false
<b>is_double (\$nom_var)</b> <b>is_float (\$nom_var)</b> <b>is_real (\$nom_var)</b>	détermine si la variable est un double	\$x = 2.3; \$double = is_real (\$x); // true
<b>is_string (\$nom_var)</b>	détermine si la variable est une chaîne	\$x = "2.3"; \$str = is_string (\$x) ; // true
<b>is_array (\$nom_var)</b>	détermine si la variable est un tableau	
<b>is_object (\$nom_var)</b>	détermine si la variable est un objet	
<b>intval (\$nom_var)</b>	définit un entier	\$x="2.3 degrés"; \$nb = intval (\$x); // 2
<b>doubleval (\$nom_var)</b>	définit un double	\$x="2.3 degrés"; \$nb = doubleval (\$x); // 2.3
<b>strval (\$nom_var)</b>	définit une chaîne de caractères	\$x="2.3 degrés"; \$nb = strval (\$x); // 2.3 degrés

❖ **Les entiers :**

Un entier est un nombre de l'ensemble des entiers naturels  $Z : Z = \{ \dots, -2, -1, 0, 1, 2, \dots \}$ . Il est possible de spécifier les nombres entiers (integer) de toutes les manières suivantes : décimale (base 10), hexadécimale (base 16), octale (base 8) éventuellement précédé du signe moins (-).

Pour utiliser la notation octale, vous devez préfixer le nombre avec un zéro; pour utiliser la notation hexadécimale, vous devez préfixer le nombre avec 0x.

```
<?php
$a = 1234; # nombre entier en base 10
$a = -123; # nombre entier négatif
$a = 0123; # nombre entier en base 8, octale (équivalent à 83 en base 10)
$a = 0x12; # nombre entier en base 16, hexadécimale
# (équivalent à 18 en base 10)
?>
```

La taille des entiers dépend de la plate-forme de support, mais la valeur maximale est généralement de milliards et des poussières (c'est un entier signé de 32 bits). PHP ne supporte pas les entiers non signés.

❖ **Les nombres à virgule flottante :**

Les nombres à virgule flottante (connus aussi sous le vocable de "double" ou "float" "nombre réels") peuvent être spécifiés en utilisant la syntaxe suivante:

```
<?php
$a = 1.234;
```

<http://nsk.forumgo.com>

```
$a = 1.2e3;
?>
```

❖ **Booléen :**

C'est le type le plus simple. Un booléen exprime les valeurs de vrai TRUE ou FALSE. Vous pouvez utiliser les constantes 'TRUE' et 'FALSE' pour spécifier une valeur de type booléen. Généralement, vous les créez avec un des opérateurs qui retourne une valeur booléenne, pour le passer à une structure de contrôle.

❖ **Les chaînes de caractères :**

Les chaînes de caractères sont des séquences de caractères. En PHP, un caractère est un octet, et il y en a 256 de possibles. La taille n'est pas un problème majeur pour une chaîne. Elle peut devenir très grande sans problème. Il n'y a pas à s'en faire pour cela.

Les chaînes comprises entre "" sont évalués:

```
$age = 30;
echo "J'ai $age ans"; // Affiche : J'ai 30 ans
echo 'J\'ai $age ans'; // Affiche : J'ai $age ans
```

L'opérateur de concaténation des chaînes est le (.) point.

```
<?php
$a = "Le"; $b = "Langage"; $c = "PHP";
$d = $a.' '.$b.' '.$c;
$e = 'Apprendre';
echo $e.' '.$d;
?>
```

Fonctions utiles pour les chaînes

Fonctions	Commentaires	Exemple
Int strlen(\$chaîne)	Longueur d'une chaîne de caractères	\$longueur_nom=strlen(\$nom);
String strtoupper(\$chaîne) String strtolower(\$chaîne)	Mettre une chaîne de caractère en minuscule/majuscule	\$nom_majuscule=strtoupper(\$nom); \$nom_minuscule=strtolower(\$nom);
Int strcmp(\$chaîne1,\$chaîne2)	Comparer 2 chaînes de caractères (pour déterminer si elles sont égales)	renvoie la valeur 0 si les deux chaînes sont identiques strcmp("Jazi", "JAZI") renvoie < 0  strcmp(\$nom1,\$nom2); identique, sauf que l'on ne tient pas compte de la casse strcmp("Jazi","JAZI") renvoie 0
String substr(\$chaîne, \$position,\$longueur)	Extraire une sous-chaîne de caractère (premier caractère: position 0 !!)	\$nom_prenom="Jazi Marwane"; \$prenom=substr(\$nom_prenom,5,7); \$nom=substr(\$nom_prenom,0,4); si pas d'argument de longueur, extrait la sous-chaîne jusqu'à la fin.
Int strpos(\$chaîne,\$motif)	Trouver la 1ère occurrence d'un motif dans une chaîne	\$nom_prenom="Jazi Marwane"; strpos(\$nom_prenom,"Ma") → renvoie 5 strpos(\$nom_prenom,"tutu") → renvoie FALSE
Int strrpos(\$chaîne,\$motif)	Trouver la Dernière occurrence d'un motif dans une chaîne	\$nom_prenom="Jazi Marwane"; strrpos(\$nom_prenom,"a") → renvoie 9 strrpos(\$nom_prenom,"tutu")

		→ renvoie FALSE												
<code>str_replace(\$ch1,\$ch2,\$ch);</code>	Remplace ch1 par ch2 se trouvant dans ch.	<pre>\$chaine = "Le php est un langage payant"; \$chaine1 = str_replace("payant", "gratuit",\$chaine); print \$chaine1; Affiche à l'écran : Le php est un langage gratuit</pre>												
<code>Split(\$separateur,\$chaine);</code>	Découpe une chaîne à base d'un séparateur	<pre>//la phrase modèle \$chaine = "Le php est un langage gratuit"; //on utilise split() pour découper la phrase \$resultat = split(" ",\$chaine); //on d écoupe la chaîne avec comme mod èle, l'espace  //Attention: le résultat obtenu est sous forme de tableau print \$resultat[1]; Affiche à l'écran php Pour mémoire, le tableau obtenu se présente comme ceci</pre> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Le</td> <td>Php</td> <td>est</td> <td>un</td> <td>langage</td> <td>Gratuit</td> </tr> <tr> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> </tr> </table>	Le	Php	est	un	langage	Gratuit	0	1	2	3	4	5
Le	Php	est	un	langage	Gratuit									
0	1	2	3	4	5									

❖ **Les tableaux :**

Avec PHP, il existe plusieurs façons de déclarer des tableaux unidimensionnels ou multidimensionnels.

**Déclaration d'un tableau à une dimension (Les indices sont numériques)**

<pre>\$T = array ("des", "dpa", "dbe"); ou \$T[] = "des"; \$T[] = "dpa"; \$T[] = "dbe"; ou \$T[0] = "des"; \$T[1] = "dpa"; \$T[2] = "dbe";</pre>	Tableau de données. Dans tous les cas, on obtiendra : T [0] = "des" ; T [1] = "dpa" ; T [2] = "dbe" ;
--	---

**Modification des valeurs implicites d'indices**

Nous avons vu qu'implicitement, Php débute par l'indice 0. Dans certains cas, il peut être pratique de commencer à 1. Php permet de modifier les valeurs des indices. Par exemple :

<code>\$T = array (1 =&gt; "des", "dpa", "dbe") ; // T[2]="dpa"</code>	équivalent à T[1]="des"; T[2]="dpa"; T[3]="dbe";
<code>\$T = array ("des", "dpa", 5 =&gt; "dbe") ; // T[0]="des" T[5]="dbe"</code>	équivalent à T[0]="des"; T[1]="dpa"; T[5]="dbe";

**Déclaration d'un tableau à une dimension (Les indices sont des chaînes de caractères)**

<pre>\$T["des"]="entreprise &amp; syst."; \$T["dpa"]="prod. animale"; \$T["dbe"]="bio expériment."; ou \$T=array("des" =&gt; "entreprise syst." "dpa" =&gt; " prod. animale ", "dbe" =&gt; " bio expériment. ");</pre>	<p>Tableau de données. Il est obligatoire d'affecter une valeur à chaque indice "chaînes de caractères"</p>
--	---

**Déclaration d'un tableau multidimensionnel (Les indices sont numériques)**

<pre>\$T[0][0]="uf info"; \$T[0][1]="uf eco"; \$T[1][0]="uf pv"; \$T[1][1]="uf pa"; ou \$Tb = array (array ("uf info", "uf eco"), array ("uf pv", "uf pa"));</pre>	<p>Tableau multidimensionnel : Chaque élément est lui-même un tableau : array(array(...), array(...), array(...)) Il est ainsi possible de créer des tableaux à 2, 3, 4 ... dimensions</p>
--	--

**Déclaration d'un tableau multidimensionnel (Les indices sont des chaînes de caractères)**

<pre>\$T["des"][0]="uf info"; \$T["des"][1]="uf eco"; \$T["dpa"][0]="uf pv"; \$T["dpa"][1]="uf pa"; ou \$Tb = array ("des" =&gt; array ("uf info", "uf eco"), "dpa" =&gt; array ("uf pv", "uf pa"));</pre>
--

**Fonctions concernant les tableaux**

Cette liste des fonctions concernant les tableaux n'est pas exhaustive.

Fonctions	Commentaires	Exemple
<b>count(\$var)</b>	compte le nombre d'éléments affectés	\$T = array ("des", "dpa"); \$nb = count(\$T); // = 2
<b>current(\$var)</b>	détermine la valeur de l'élément en cours	\$T[5]="formco"; \$val =current(\$T); //="formco"
<b>key(\$var)</b>	détermine l'indice de l'élément en cours	\$T[5]="formco"; \$val = key(\$T) ; // = 5
<b>reset(\$var)</b>	déplace le pointeur sur le 1er élément	\$reset (\$T); // = "des"
<b>list (\$indice, \$val) each (\$var_tableau)</b>	permettent de parcourir les éléments même si les indices ne sont pas consécutifs.	while (list(\$indice, \$val) =each(\$T) ) { echo (" \$indice - \$val"); // "0 - des" puis "1 - dpa" }
<b>next(\$var) prev(\$var)</b>	permettent d'aller respectivement à l'élément suivant et précédent.	fonctions déconseillées
<b>sort(\$var)</b>	trie les éléments selon un ordre numérique ou alphabétique et réaffecte les indices du tableau	\$T=array ("des", "dpa", "dbe"); sort(\$T); while (list(\$indice, \$val) =each(\$T) ) { echo (" \$indice - \$val"); // "0 -



		dbe" puis "1 - des" // puis "2 - dpa" } }
<b>asort(\$var)</b>	trie les éléments mais ne réaffecte pas les indices	\$T=array ("des","dpa","dbe"); asort(\$T); // "2 - dbe" puis "0 - des" // puis "1 - dpa"
<b>rsort(\$var)</b>	même fonction que sort() mais trie les éléments en ordre inverse	
<b>rsort(\$var)</b>	même fonction que asort() mais trie les éléments en ordre inverse	
<b>ksort(\$var)</b>	trie par valeur d'indice	
<b>krsort(\$sort)</b>	même fonction que ksort() mais trie les indices en ordre inverse	
<b>usort(\$var, \$critere), uasort(\$var, \$critere), uksort(\$var, \$critere)</b>	trie selon un critère donné par l'utilisateur	// tri sur l'inverse de la chaîne \$T=array ("des","dpa","dbe"); uasort(\$T, strrev(\$T)); // "1 - dpa" puis "2 - dbe" // puis "0 - des"

**c. Les opérateurs :**

Les opérateurs vont permettre de manipuler les variables (comparaison, affectation, calculs, ...). Il en existe de différentes catégories. Tout comme en mathématique certains signes ont des priorités de calcul par rapport à d'autres, les opérateurs PHP ont aussi des priorités décrites plus tard.

➤ **Les opérateurs de calcul + - \* /**

Ils servent aux calculs : \* (multiplier), + (additionner), - (soustraire), / (diviser), = (affecter) , % (prendre le reste de la division = modulo).

Opérateur	Nom	Signification	Exemple	Résultat
+	plus	addition	x + 3	14
-	moins	soustraction	x - 3	8
*	multiplié par	multiplication	x*2	22
/	divisé par	division	x /2	5.5
%	modulo	reste de la division par	x%5	1
=	a la valeur	affectation	x=5	5

➤ **Les opérateurs de comparaison =, <, ...**

Ils sont utilisés dans les instructions de conditions (if .... then ....).

Opérateur	Description	Exemples	
==	Egal	if (w == 1) then { ... }	Si w égal 1 alors ...
!=	Différent	if (w != 1) then { ... }	Si w différent de 1 alors ...
<	Inférieur	if (w < 1) then { ... }	Si w plus petit que 1 alors ...
<=	Inférieur ou égal	if (w <= 1) then { ... }	Si w plus petit ou égal à 1 alors ...
>	Supérieur	if (w > 1) then { ... }	Si w plus grand que 1 alors ...
>=	Supérieur ou égal	if (w >= 1) then { ... }	Si w plus grand ou égal à 1 alors ...
?:	Opérateur ternaire	x = (w>0) ? 3 : 5	Si w>0 alors x=3, sinon x=5

➤ **Les opérateurs associatifs += -= \*= /=**



Ils permettent à la fois le calcul et l'affectation de la valeur calculée à une variable.

Opérateur	Description	Exemple	Signification	Résultat
=	plus égal	x += y	x = x + y	16
-=	moins égal	x -= y	x = x - y	6
*=	multiplié égal	x *= y	x = x * y	55
/=	divisé égal	x /= y	x = x / y	2.2

➤ **Les opérateurs logiques && (ou and) // (ou or) !**

Ils servent à comparer 2 expressions.

Opérateur	Description	Exemples
	Opposé ("Not" = opérateur booléen)	if (! w) then { ... } Si w non vide (= non nul) alors ...
&&	Et	if (x>=1) && (x<=10) then Si x>=1 ET x<=10 alors ... (si x compris entre 1 et 10 inclus)
	Ou	if (w>=1)    (x==2) Si w>=1 OU x=2 alors ...

➤ **Les opérateurs unaires + et -**

L'opérateur - est courant puisqu'il sert à inverser la valeur donnée :

Exemples :  
- \$x si \$x = 2 alors \$x = -2  
- \$y si \$y = -4 alors \$y = 4

➤ **Les opérateurs d'incrémenter ++ et --**

Ils servent à augmenter (ou diminuer) de 1 la valeur d'une variable. Ils sont très utilisés dans les structures de boucles.

Dans les exemples x vaut 3.

Opérateur	Description	Exemple	Signification	Résultat
x++	Incrémenter (instruction équivalente à x=x+1)	y = x++	3 puis plus 1	4
x--	Décrémenter (instruction équivalente à x=x-1)	y = x--	3 puis moins 1	2

**Remarque :**

On trouve aussi l'écriture suivante : ++x ou --x. L'incrémenter (ou la décrémentation) a lieu avant exécution de l'instruction. Ainsi :

alors que y = x++ + 2 si x=3 alors y = 5 puis x = 4  
alors que y = ++x + 2 si x=3 alors x= 4 puis y = 6

Donc, il convient d'être très vigilant lorsqu'on utilise ces opérateurs combinés à d'autres.

➤ **L'opérateur d'affectation =**

Il sert à donner une valeur à une variable.

Exemples : \$x = 10 x prend la valeur 10  
\$y = "Lycée de Korba" y est initialisé par une chaîne de caractère

➤ **L'opérateur de concaténation de chaînes de caractères.**

Avec PHP, il existe 2 façons de concaténer des chaînes : avec opérateur et sans opérateur.

- **L'opérateur "."** sert à associer des chaînes de caractères pour en former une nouvelle.

Exemples : \$section = "Sciences de l'informatique" ;  
\$chaine1 = "- Lycée de Korba - " . \$section . " - " ;

**Attention :** Le point doit être précédé et suivi d'un espace pour éviter de le confondre avec le point décimal ("2 . 3" donnera "23" – 2 sera considéré comme la valeur décimale 2.3)

- **Sans opérateur**, on peut créer de nouvelles chaînes en intégrant le nom de la variable dans la chaîne. Sans contexte, l'écriture est simplifiée et est donc conseillée.

Exemples : `$section = "Sciences de l'informatique" ;`  
`$chaine1 = "- Lycée de Korba - $section - " ;`

**Attention :** Si l'on veut écrire "DES3", il convient de coder : `${departement}3` afin que PHP comprenne bien que le 3 ne fait pas partie du nom de la variable.

### L'opérateur de suppression d'erreur @

Permet d'éliminer la génération de messages d'erreur provenant de fonctions intégrées.

Exemples : `print ($x / 0) ; // provoque une erreur division par zéro`  
`@ print ($x / 0) ; // aucune erreur est signalée, aucune valeur n'est imprimée.`

### Priorités des opérateurs

Les opérateurs s'effectuent dans l'ordre suivant de priorité (du degré de priorité le plus faible au degré de priorité le plus élevé).

Dans le cas d'opérateurs de priorité égale, de gauche à droite.

Opérateur	Signification
,	virgule ou séparateur de liste
= += -= *= /= %=	affectation
? :	opérateur conditionnel
	ou logique
&&	et logique
== !=	égalité
< <= >= >	relationnel
+ -	addition soustraction
* /	multiplier diviser
! - ++ --	unaire
()	parenthèses

### 5. Les entrées/sorties :

- ❖ Les saisies de données (entrées) font être réalisées par l'intermédiaire des formulaires
- ❖ Les éditions des résultats (sorties) vont s'inscrire dans la page par les fonctions
  - `echo ("message");`
  - `print ("message");`

### 6. Les structures de contrôle :

#### a. Les structures conditionnelles :

- ❖ L'instruction **if** :

Syntaxe :

```

If ( condition )
{
    instruction1 si vrai;
}
Else
{
    instruction2 si faux;
}

```

**Remarque :**

Il est tout à fait possible de faire des conditions imbriquées. Pour des facilités de lecture du script, il est conseillé de décaler vers la droite les blocs if internes.

❖ **Les structures de branchement :**Syntaxe :

```
switch ( expression )
{
    case v1: instruction_1; break;
    case v2: instruction_2; break;
    case vn: instruction_n; break;
    default :
        instruction_n+1;
}
```

**Remarques :**

- L'expression peut être une simple variable ou une expression à analyser.
- Plusieurs "case" peuvent être empilés.
- Chaque "case" est terminée par une instruction break, pour éviter que le compilateur n'exécute la suite des "case"
- L'instruction "default" est facultative. Elle sert à traiter tous les cas non précisés dans la liste des "case".
- Dans le cas d'une variable alphanumérique, on écrirait : case "1"

**b. Les structures itératives :**❖ **La boucle FOR :**Syntaxe :

```
for (expression_initiale ; condition de sortie ; expression de progression)
{
    instructions;
}
```

❖ **La boucle DO...WHILE : Faire ... tant que**Syntaxe :

```
do
{
    instructions;
} while (condition) ;
```

**Remarque :**

Dans une boucle "Faire ... tant que ...", les instructions situées dans la boucle sont exécutées au moins une fois, même si la condition n'est pas remplie.

❖ **La boucle WHILE : Tant que**Syntaxe :

```
while (condition)
{
    instructions;
}
```

**Remarque :**

Dans une boucle "Tant que ...", les instructions situées dans la boucle peuvent ne jamais être exécutées si la condition n'est pas remplie.

## 7. Les fonctions en Php :

- Les fonctions et les procédures regroupent un ensemble d'instructions réutilisables simplement et qui accomplissent un ensemble d'opérations. Elle peuvent (mais ce n'est pas obligé) accepter des valeurs (appelées "arguments" ou "paramètres"). Si il y en a plusieurs, les arguments sont séparés par des virgules.
- Une fonction réalise une succession d'instructions et renvoie une (et une seule !) valeur issue d'un calcul (instruction return)
- Une procédure réalise une succession d'instructions mais ne renvoie pas de valeur en retour.

### Syntaxe

```
function Nom_Fonction (parametres)
{
  Instructions ;
  [return Valeur_Finale; ]
}
```

### Exemple d'une fonction avec un argument

```
function calcul_surface_cercle ($rayon)
{
  # Cette fonction calcule la surface d'un cercle à partir du rayon passé en
  paramètre
  $surf= 3.14 * $rayon * $rayon;
  return $surf; // ou return ($surf);
}
```

### Exemple d'une fonction à 2 arguments

```
function calcul_surface_rectangle ($longueur, $largeur)
{
  /* cette fonction calcule la surface d'un rectangle à partir des longueur et
  largeur passées en paramètre */
  return $longueur*$largeur; // calcule la surface et renvoie la valeur
}
```

### Exemple d'une procédure à 2 arguments

```
function affichage_prix ($s_nom_article, $d_prix)
{
  echo ("<B> $s_nom_article</B>"); // nom article en gras
  echo ("<FONT color=red> $d_prix FF </FONT>"); // prix en FF en rouge
  echo ("<FONT color=green> " . $d_prix*6.556597 . "€</FONT>"); // prix en euros en
  vert
  echo "<HR>"; // ligne horizontale
}
```

### ❖ Transmission des arguments : par valeur et par référence

- Par défaut, "Par valeur" est le type de transmission par défaut, c'est-à-dire que la variable conserve sa valeur initiale après l'appel de la fonction même si celle-ci l'a modifiée.

Exemple : À partir d'une fonction qui calcule le double d'une valeur donnée en paramètre :

```
function double ($n)
{
  $n = $n * 2;
  return $n;
}
```

Si on passe l'argument "par valeur" à cette fonction double(), on obtient :

```
$a=5; // valeur initiale. On va en calculer le double
echo ("double=" . double ($a) . " valeur=" . $a); // double=10 valeur=5
```

- "Par référence" signifie que la valeur initiale peut être modifiée dans la fonction.

Syntaxe : Un symbole & précède le nom de l'argument (variable).

Exemple : Si on passe l'argument "par référence" à la même fonction double(), on obtient :

```
$a=5; // valeur initiale. On va en calculer le double
```

```
echo ("double=" . double (&$a) . " valeur= " . $a) ; // double=10 valeur=10
```

### Remarque :

En Php tous les types de variables peuvent être renvoyés par un module (fonction ou procédure).

### Portée des variables : locale, globale, statique

#### Variable locale :

Une variable déclarée dans une fonction aura une portée limitée à cette seule fonction. On ne pourra donc utiliser sa valeur ailleurs dans le script. C'est une variable locale à la fonction.

Exemple : \$matiere ="informatique"; // n'est connue que la fonction dans laquelle elle se trouve

#### Variable globale :

Si la variable est déclarée tout au début du script, en dehors et avant toute fonction, elle est toujours globale. Afin de distinguer une variable locale qui porterait le même nom qu'une variable globale, il faut la déclarer à l'aide du mot-clé **global**.

Exemple :

```
<?
$matiere ="informatique" // est connue de toutes les fonctions
function signer()
{
    global $ matiere; // $matiere est la variable globale précédente
    $titre="Professeur ";
    $titre = $titre . "d'" . $matiere; // Professeur d'informatique
}
?>
```

### Remarque :


Pour faciliter la gestion des variables, on ne peut que conseiller de les déclarer en début de script (comme dans la plupart des langages de programmation).

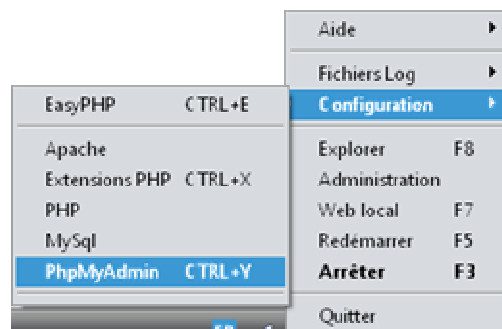
#### Variable statique :

Implicitement, les variables locales d'une fonction sont réinitialisées à chaque appel de fonction. Pour conserver leurs valeurs précédentes, il faut les déclarer par le mot-clé **static**.

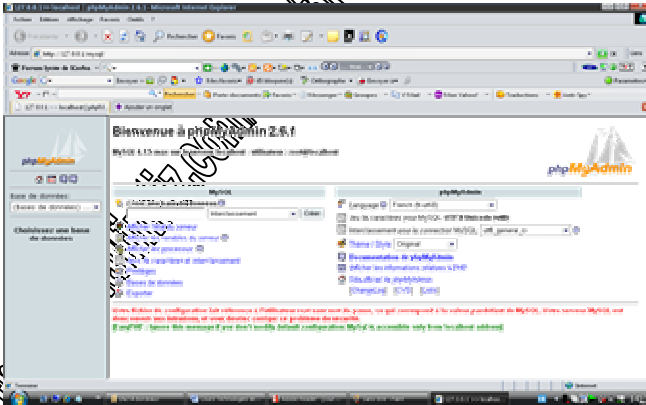
## 8. Utiliser MySQL avec Php :

### a. Création d'une base de données avec PhpMyAdmin :

La création de la base ainsi que celles des tables sera faite à l'aide de PhpMyAdmin. Pour démarrer ce programme, faire clic droit sur , l'icône de EasyPhp, puis choisir "Configuration" et "PhpMyAdmin".



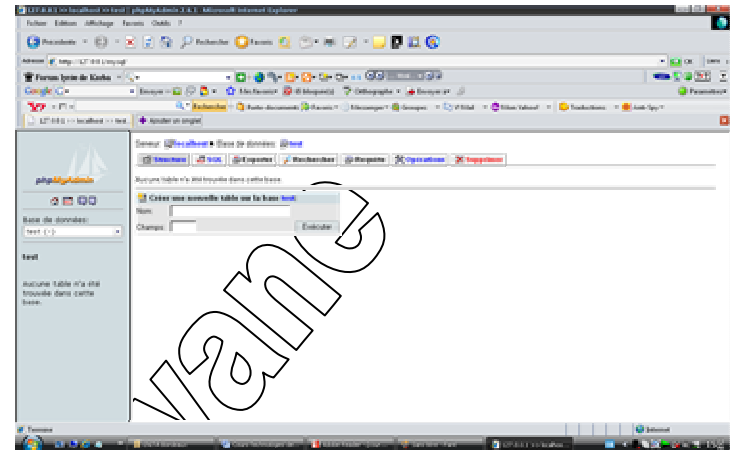
### ❖ Création d'une base MySQL



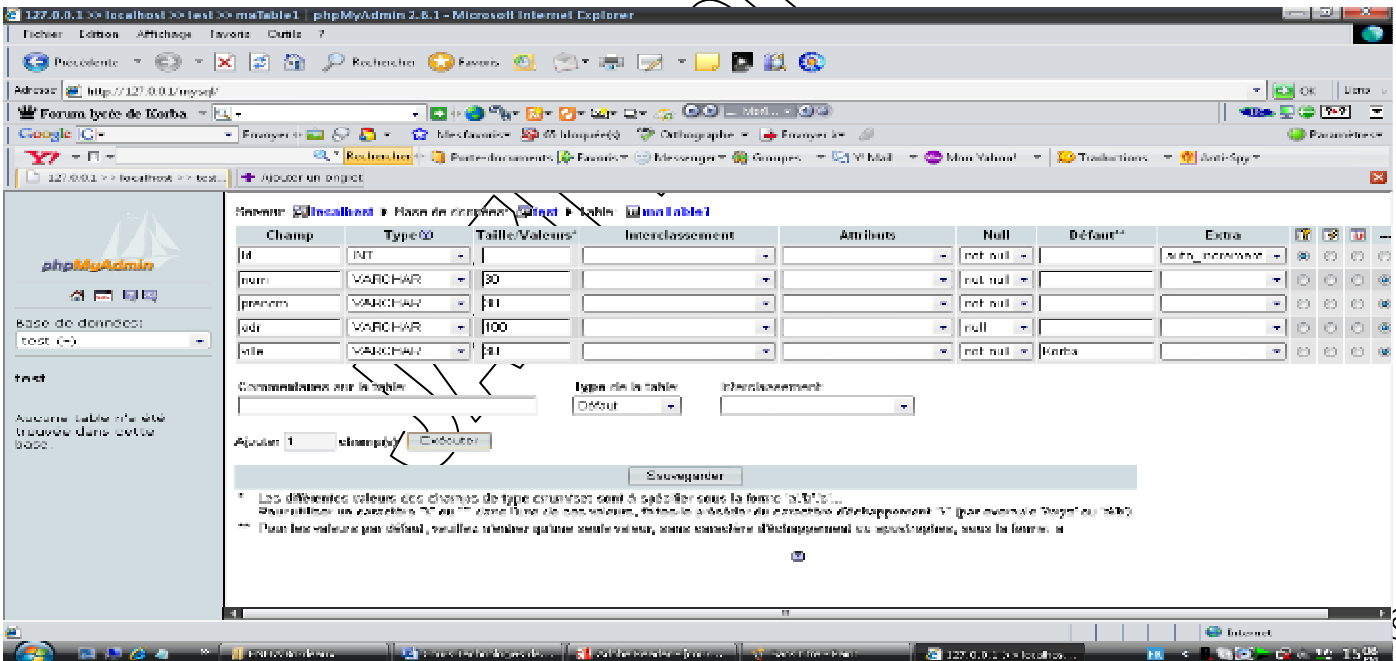
Le navigateur s'ouvre proposant :

- de créer une nouvelle base de données
- ou de choisir parmi les bases de données déjà créées (liste dans le cadre de gauche)

Lorsque la base est créée (ici, elle s'appelle "test"), la création d'une table est possible : Dans la rubrique, "créer une nouvelle table sur la base test", donner le "nom" de la table et le nombre de "champs" qu'elle contient. Cette valeur peut être modifiée ultérieurement. "Exécuter" la commande



Voici l'écran suivant :



On y retrouve les 5 lignes qui vont décrire les champs de la table, elles-mêmes constituées de différentes colonnes.

- Champ = nom du champ
- Type = type de la donnée (numérique, alphanumérique, date, ...)
- Longueur = longueur de la données (adr sur 100 caractères)
- Attributs = type de valeurs (binaire, non signé, ...)
- Null = la valeur peut être nulle ou pas

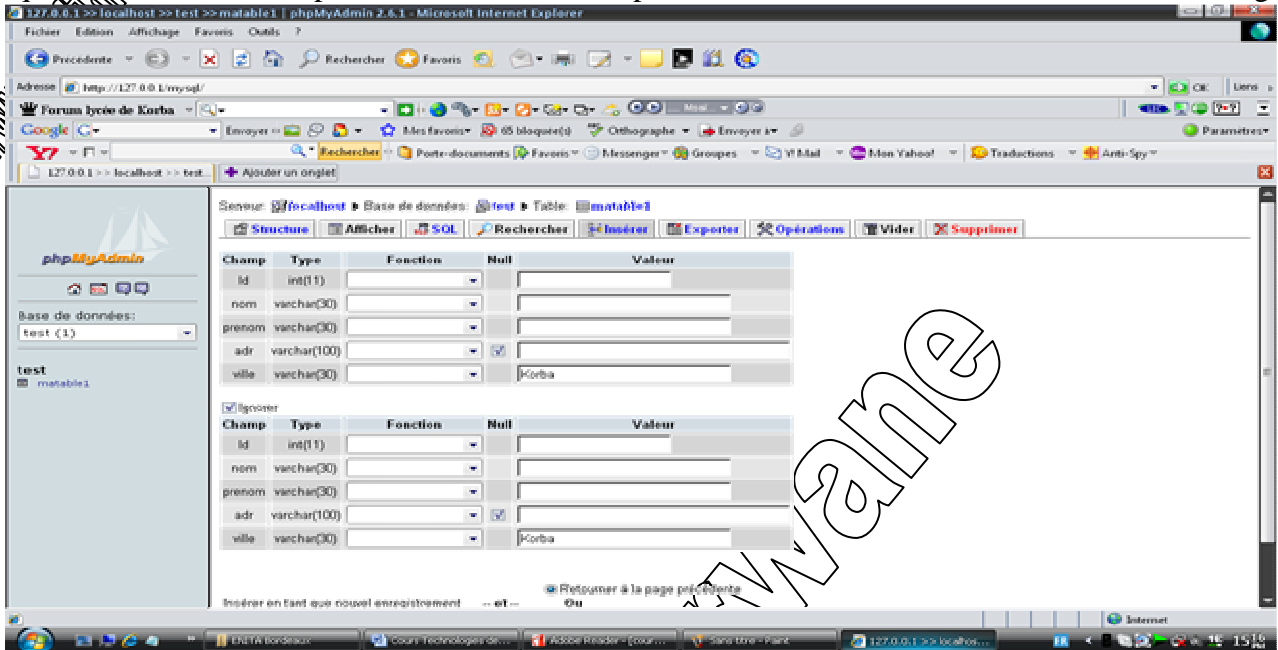
<http://fisk.forumgo.com>



- Défaut = valeur par défaut
- Extra = numérotation automatique (auto\_increment)
- Primaire = clé de la table (automatiquement indexée et unique)
- Index = d'une recherche selon ce champ, permet un accès plus rapide aux enregistrements
- Unique = les doublons de ce champ sont interdit dans l'ensemble de la table

❖ **Enregistrer des données**

Lorsque la table et les champs sont créés, il est alors possible de saisir des données en vue de l'interroger.



b.

**Interrogation d'une base MySQL :**

Pour interroger une base de données, il est impératif de maîtriser le langage SQL. Nous ne le détaillerons pas dans ce manuel.

L'accès aux données se décompose en plusieurs étapes :

- ouvrir la connexion sur le serveur MySql (\*)
- ouvrir une connexion sur la base de données (\*)
- définir la requête à exécuter
- exécuter la requête. On obtient les lignes de résultats dans un recordset. (\*)
- extraire les données de chaque ligne de résultat
- traiter les données (calcul et affichage par exemple) (\*)
- fermer le recordset pour libérer la mémoire
- fermer la connexion

(\*) étapes obligatoires.

❖ **Ouverture de la connexion sur la base de données : mysql\_connect()**

La fonction qui ouvre une connexion à un serveur MySQL, est mysql\_connect() dont la syntaxe est la suivante :

**int mysql\_connect (string hostname [:port] , string username , string password)**

- avec : hostname : nom de l'hôte ou du site (chaîne de caractères)
- username : nom de connexion à la base (=login) (chaîne de caractères)
- password : mot de passe associé au login (chaîne de caractères)

<http://flsk.forumgo.com>

La fonction retourne un **nombre positif** si la connexion a réussi sinon elle retourne **FALSE (=0)**. La connexion sera fermée automatiquement dès que l'exécution du script sera terminée, à moins d'être fermée explicitement avec **mysql\_close()**, ce qui est fortement conseillée.

```
// Connexion locale, login "root" et pas de mot de passe
$maConnexion = mysql_connect ("localhost", "root", "");
if ($maConnexion == FALSE) // si la connexion a échoué
{
    die ("connexion impossible"); // affiche le message puis arrête le script
}
echo ("connexion réussie"); // ou instructions diverses
```

### Sélectionner une base de données MySQL : mysql\_select\_db()

Cette fonction permet d'indiquer dans quelle base se trouvent les données à traiter.

**int mysql\_select\_db (string database, resource link\_identifier )**

avec database : nom de la base de données MySQL qui contient les tables

link\_identifier : identifiant de la connexion ouverte par **mysql\_connect()**. C'est un paramètre facultatif

La fonction retourne **TRUE** en cas de succès, **FALSE** en cas d'échec. Toutes les requêtes exécutées avec **mysql\_query()** seront faites avec la base de données active.

```
$ConnectBase = mysql_select_db ("maBase", $maConnexion);
if ($ConnectBase == FALSE) // si la base est inaccessible
{
    die ("base inaccessible"); // arrêt du script
}
echo ("accès correct à la base");
```

### ❖ Création et exécution de la requête :

La fonction **mysql\_query()** envoie une requête SQL à la base de données active. Sa syntaxe est la suivante : **resource mysql\_query (string query, resource link\_identifier )**

avec : query : nom de la requête ou chaîne de caractères contenant la requête

link\_identifier : identifiant de la connexion ouverte par **mysql\_connect()**. C'est un paramètre facultatif

La fonction retourne **FALSE (=0)** pour indiquer l'échec de la requête, c'est-à-dire qu'elle n'a pas pu être exécutée sur le serveur, le plus souvent dans le cas d'une erreur de syntaxe SQL ou de droits insuffisants sur la base.

**Remarque :** Il est parfaitement possible qu'une requête correcte au niveau de sa syntaxe ne retourne aucune ligne. La fonction retourne donc **TRUE**.

Pour les commandes **DELETE**, **INSERT**, **REPLACE**, ou **UPDATE**, la fonction **mysql\_affected\_rows()** permet de connaître le nombre de lignes affectées. Pour les commandes **SELECT**, **mysql\_query()** retourne un identifiant de résultat que l'on peut passer à **mysql\_result()**. A la fin du traitement, il est conseillé de libérer la mémoire avec **mysql\_free\_result()**.

```
$sql = "Select * from maTable"; // La requête
$result = mysql_db_query ("maBase", $sql); // envoi au serveur
if ($result == FALSE) // si la requête est incorrecte
{
    die ("requête incorrecte : $sql"); // arrêt du script
}
echo ("requête correcte");
```

**Autre commande**

La fonction `mysql_db_query()` permet tout à la fois d'indiquer de quelle base sont extraites les données et envoie une requête SQL à la base de données donnée en paramètre. Sa syntaxe est la suivante :

**resource mysql\_db\_query (string database, string query, resource link\_identifier )**

avec : database : nom de la base de données qui contient les tables

query : nom de la requête ou chaîne de caractères contenant la requête

link\_identifier : identifiant de la connexion ouverte par `mysql_connect()`. C'est un paramètre facultatif

❖ **Fermer la connexion : mysql\_close()**

La fonction `mysql_close()` ferme la connexion MySQL. Elle retourne FALSE si la connexion a échoué et TRUE en cas de succès.

**bool mysql\_close (resource link\_identifier )**

avec : link\_identifier : identifiant de la connexion ouverte par `mysql_connect()`.

**Remarques :**

- cette commande n'est pas obligatoire, car toutes les connexions non persistantes seront automatiquement fermées à la fin du script. Toutefois elle est fortement conseillée.

- `mysql_close()` ne ferme pas les connexions persistantes générées par `mysql_pconnect()`.

```
mysql_close ($maConnexion);
```

❖ **Extraction & traitement des données**

Php possède différentes fonctions qui permettent le retour des résultats issus d'une requête SQL. L'une permettra d'afficher les résultats selon la **position** du champ dans la requête (**indice**), une autre selon le **nom du champ** donné dans la table et une dernière qui permettra tout autant de travailler sur les **indices que sur les noms des champs**. La manière d'afficher le résultat dépendra de la fonction utilisée.

➤ **à partir des numéros de champ :**

La fonction `mysql_fetch_row()` retourne **UNE** ligne d'enregistrements appelée recordset sous la forme d'un tableau. Chaque colonne est enregistrée sous la forme d'un tableau commençant à la position 0.

**array mysql\_fetch\_row (resource result\_identifier )**

avec : resource\_identifier : identifiant retourné par `mysql_query()`.

Cette fonction retourne un **tableau énuméré** qui correspond à la ligne demandée, ou **FALSE**, s'il ne reste plus de ligne. Comme la fonction ne retourne qu'une seule ligne, il faudra boucler sur le nombre d'enregistrements précédemment calculé par la fonction `mysql_num_rows()`.

On affiche les résultats comme étant les éléments d'un tableau.

```
// boucle sur le nbre de lignes de
$result for ($i = 0 ; $i < mysql_num_rows($result) ; $i++)
{
    // extrait de $result la prochaine ligne sous la forme d'un tableau énuméré
    $tbl_ligne = mysql_fetch_row ($result);
    echo ("$tbl_ligne[0] -- $tbl_ligne[1]<BR>"); // affichage
}
```

➤ **à partir des noms de champ :**

La fonction `mysql_fetch_object()` retourne **UNE** ligne d'enregistrements appelée recordset. Cette fonction est identique à `mysql_fetch_array()`, à la différence qu'elle retourne un objet à la place d'un tableau. On peut alors accéder aux valeurs des champs par leur nom, mais plus par leur indice.

**array mysql\_fetch\_object (resource result\_identifier, int result\_type )**

avec : resource\_identifier : identifiant retourné par `mysql_query()`

result\_type : constante facultative qui peut être

**MYSQL\_ASSOC,**  
**MYSQL\_NUM** ou  
**MYSQL\_BOTH**

Cette fonction retourne une seule ligne sous forme d'un objet dont les propriétés correspondent à **une ligne de résultat**, ou **FALSE**, s'il ne reste plus de ligne. Il faudra boucler sur le nombre d'enregistrements précédemment calculé par la fonction **mysql\_num\_rows()** ou tester s'il ne reste plus de ligne.

Pour afficher les résultats, on donne le nom de l'objet suivi du nom du champ dans la base et séparé par ->

```
// boucle tant qu'il y a des lignes de résultat dans
$resultat while (($ligne = mysql_fetch_object($result)) == TRUE)
{
    echo ("$ligne->nom -- $ ligne->prenom<BR>"); // affichage
}
```

### ➤ à partir des numéros ou des noms de champ :

La fonction **mysql\_fetch\_array()** retourne un tableau qui contient la ligne demandée, ou **FALSE** s'il ne reste plus de ligne. Il s'agit une fonction étendue de **mysql\_fetch\_row()** car elle enregistre les données dans un tableau à indices numériques et/ou dans un tableau à indices alphanumériques.

**array mysql\_fetch\_array (resource result\_identifiant, int result\_type )**

avec : resource\_identifiant : identifiant retourné par **mysql\_query()**

result\_type : constante facultative qui déterminent l'accessibilité des données :

**MYSQL\_ASSOC** : dans tableau à indices alphanumériques uniquement

**MYSQL\_NUM** : dans tableau à indices numériques uniquement ou

**MYSQL\_BOTH** : on pourra utiliser l'un ou l'autre des indices

**Remarque :** Si plusieurs colonnes portent le même nom, il faudra les distinguer par un alias, exemple :  
select t1.nom as n1, t2.nom as n2 from t1, t2;

Comme les 2 fonctions précédentes, cette fonction retourne une seule ligne. Il faudra boucler sur le nombre d'enregistrements précédemment calculé par la fonction **mysql\_num\_rows()** ou tester s'il ne reste plus de ligne.

Pour afficher les résultats, on utilise la syntaxe des tableaux à indices alphanumériques, les indices correspondants aux noms des champs dans les tables.

```
while (($ligne = mysql_fetch_array($result)) == TRUE)
{
    echo $row["nom"]; echo $row["prenom"];
}
```

### Quelle fonction choisir ?

Concernant la vitesse, **mysql\_fetch\_object()** est aussi rapide que **mysql\_fetch\_array()** et presque aussi rapide que **mysql\_fetch\_row()** (la différence est insignifiante). Etant donné les possibilités plus grandes apportées par **mysql\_fetch\_array()** sans perte de performance, il est conseillé d'utiliser cette fonction.

### ❖ Libérer la mémoire : **mysql\_free\_result()**

La fonction **mysql\_free\_result()** efface le résultat de la mémoire pour éviter une saturation de la place mémoire lors de l'exécution du script.

**int mysql\_free\_result (resource result\_identifiant)**

avec : resource\_identifiant : identifiant retourné par **mysql\_query()**

```
mysql_free_result ($result);
```

❖ Quelques autres fonctions MySQL

Fonction	Rôle et syntaxe
mysql_get_host_info	Lit les informations sur l'hôte MySQL. string mysql_get_host_info(resource link_identifier)
mysql_get_server_info	Lit les informations sur le serveur MySQL. string mysql_get_server_info (resource link_identifier)
mysql_fetch_lengths	Retourne la taille de chaque colonne d'une ligne de résultat. array mysql_fetch_lengths (resource result_identifier)
mysql_insert_id	Retourne l'identifiant généré par la dernière requête INSERT. int mysql_insert_id (resource link_identifier)
mysql_create_db	Crée une base de données MySQL. int mysql_create_db (string database_name , resource link_identifier)
mysql_change_user	Change le nom de session de l'utilisateur actif. int mysql_change_user (string user , string password , string database , resource link_identifier)
mysql_field_len	Retourne la longueur du champs spécifié. int mysql_field_len (resource result_identifier , int field_offset)
mysql_list_fields	: Liste les champs du résultat MySQL. resource mysql_list_fields (string database_name , string table_name , resource link_identifier)
mysql_list_dbs	Liste les bases de données disponibles sur le serveur MySQL. resource mysql_list_dbs (resource link_identifier)
mysql_num_fields	Retourne le nombre de champs d'un résultat. int mysql_num_fields (resource result_identifier)
mysql_db_name	Lit les noms des bases de données. int mysql_db_name (resource result_identifier , int row , mixed field)
mysql_error	Retourne le texte associé avec l'erreur générée lors de la dernière requête. string mysql_error (resource link_identifier)
mysql_affected_rows	Retourne le nombre de lignes affectées lors de la dernière requête SQL. (utilisée dans le cas des requêtes de manipulation des données (insert, update, delete, ...) int mysql_affected_rows(resource link_identifier)
mysql_field_type	Retourne le type de la colonne spécifiée dans le résultat courant. string mysql_field_type (resource result_identifier , int field_offset)
mysql_field_seek	Déplace le pointeur de résultat. int mysql_field_seek (resource result_identifier , int field_offset)
mysql_list_tables	Liste les tables d'une base de données. resource mysql_list_tables (string database , resource link_identifier)
mysql_tablename	Lit le nom de la table qui contient le champs spécifié. string mysql_tablename (resource result_identifier , int i)
mysql_field_table	: Retourne le nom de la table où se trouve une colonne. string mysql_field_table (resource result_identifier , int field_offset)
mysql_data_seek	Déplace le pointeur interne de résultat vers un numéro d'enregistrement int mysql_data_seek (resource result_identifier , int row_number)
mysql_drop_db	Efface une base de données MySQL. boolean mysql_drop_db (string database_name , resource link_identifier)
mysql_field_name	Retourne le nom d'une colonne. string mysql_field_name (resource result_identifier , int field_index)



# Chapitre 3 : Publication électronique

## Objectif

- ❖ Publier des sites Web dynamiques



## Rappel :

un site Internet ou Intranet d'informations reste avant tout une vitrine, un support. C'est pour cela un important moyen de communication et d'identification. La mise en place d'un site doit être étudiée de façon approfondie et ne peut être abordée comme la rédaction d'un document classique. Il s'agit de rendre cohérent une masse d'information et de la mettre à disposition d'un panel hétérogène d'utilisateurs plus ou moins initiés à la navigation sur un site web. C'est pourquoi un intérêt tout particulier doit lui être porté, aussi bien au niveau du contenu que de la présentation.

## II-Hébergement :

Afin de rendre disponible un site web 24/24H il est nécessaire qu'il soit hébergé sur un serveur relié en permanence à Internet. Techniquement il est possible d'héberger son site Internet soi-même à condition de posséder un débit en sens montant suffisamment important. En effet sur un site Internet les visiteurs font essentiellement du téléchargement en sens descendant (download), ce qui signifie que le serveur fait surtout de l'envoi de pages web (upload).

Pour autant, afin de rendre un service de qualité aux visiteurs, il est conseillé de recourir aux services d'un hébergeur, c'est-à-dire une société proposant l'hébergement (en anglais hosting) du site Internet sur des serveurs spécialisés connectés en permanence à Internet à très haut débit.

On distingue généralement les types d'hébergement suivants :

- ❖ Hébergement gratuit
- ❖ Hébergement professionnel
  - hébergement mutualisé (en anglais mutualized hosting) : il s'agit d'une offre d'hébergement où le serveur héberge un grand nombre de sites ;
  - hébergement dédié (en anglais dedicated hosting) : il s'agit de la location complète d'un serveur ;
  - colocation (en anglais housing) : cette formule consiste à louer une baie d'hébergement pouvant accueillir les serveurs du client.

### 1-Hébergement gratuit :

Les fournisseurs d'accès à Internet proposent généralement dans leur offre une formule d'hébergement de site web avec un espace de stockage assez limité. Il existe ainsi un grand nombre de sites offrant de tels services gratuits, mais il est essentiel de regarder en détails les conditions dans lesquelles le service est rendu.

En effet le service d'hébergement est souvent rendu en contrepartie de publicités affichées sur le site, soit dans un coin, soit sous la forme de pop-ups. D'autre part le temps d'accès à ce genre de service est plutôt moyen et la disponibilité du site web n'est pas garantie. Il est à noter qu'il est relativement rare de pouvoir mettre en place un nom de domaine avec ce type d'hébergement mais il est tout de même possible de mettre en place une redirection du nom de domaine vers l'URL qu'offre l'hébergeur gratuit (du type <http://perso.mon-fournisseur.com/mapageweb>).



### III-Référencement :

#### 1-Qu'est-ce que le référencement ?

On désigne habituellement par le terme « **référencement** » (en anglais *web positioning*) l'ensemble des techniques permettant d'améliorer la visibilité d'un site web :

- **indexation** (en anglais **submission**) consistant à faire connaître le site auprès des outils de recherche grâce aux formulaires que ceux-ci proposent ;
- **positionnement** (en anglais **positioning**) consistant à positionner le site ou certaines pages du site en première page de résultat pour certains mots-clés ;
- **classement** (en anglais **ranking**) dont le but est similaire au positionnement mais pour des expressions plus élaborées ; une partie du travail étant d'identifier ces requêtes.

Plus autant, toute la difficulté de l'exercice ne réside pas tant dans la promotion du site auprès des moteurs de recherche que dans la structuration du contenu pour être bien positionné dans les résultats sur des mots-clés préalablement choisis.

En effet une majorité d'internautes utilise les moteurs de recherche afin de trouver de l'information et interroge à ce titre un moteur de recherche à l'aide de mots clés (en anglais **keywords**). Il est donc essentiel avant tout chose de se préoccuper du contenu que l'on propose afin de répondre au mieux aux attentes des internautes, et d'identifier dans un second temps les mots-clés susceptibles d'être saisis par ces derniers !

#### 2-Améliorer son référencement :

Il existe quelques techniques de conception de site permettant de donner plus d'efficacité au référencement des pages d'un site :

- un contenu original et attractif,
- un titre bien choisi,
- une URL adaptée,
- un corps de texte lisible par les moteurs,
- des balises META décrivant précisément le contenu de la page,
- des liens bien pensés,
- des attributs ALT pour décrire le contenu des images.

##### *a-Contenu de la page web :*

Les moteurs de recherche cherchent avant tout à fournir un service de qualité à leurs utilisateurs en leur donnant les résultats les plus pertinents en fonction de leur recherche ainsi avant même de songer à améliorer le référencement il est indispensable de s'attacher à créer un contenu consistant et original.

Un contenu original ne signifie pas un contenu qui n'est proposé par aucun autre site, cela serait une mission impossible. En revanche il est possible de traiter un sujet et de lui amener de la plus-value en approfondissant certains points, en l'organisant d'une manière originale ou en mettant en relation différentes informations.

D'autre part, toujours dans l'optique de fournir le meilleur contenu aux visiteurs, les moteurs de recherche accordent de l'importance à la mise à jour des informations. Le fait de mettre à jour les pages du site permet donc d'augmenter l'indice accordé par le moteur au site ou en tout cas la fréquence de passage du robot d'indexation.

##### *b-Titre de la page :*

Le titre est l'élément de prédilection pour décrire en peu de mots le contenu de la page, c'est notamment le premier élément que le visiteur va lire dans la page de résultat du moteur de recherche, il est donc essentiel de lui accorder une importance particulière. Le titre d'une page web est décrit dans l'en-tête de la page web entre les balises `<TITLE>` et `</TITLE>`.

Le titre doit décrire le plus précisément possible, en 6 ou 7 mots maximum, le contenu de la page web et sa longueur totale recommandée ne doit pas dépasser la soixantaine de caractères.

Le titre est d'autant plus important qu'il s'agit de l'information qui s'affichera dans les favoris de l'internaute, ainsi que dans l'historique.

#### **c-URL de la page :**

Certains moteurs de recherche accordent une importance capitale aux mots-clés présents dans l'URL, notamment les mots-clés présents dans le nom de domaine. Il est donc conseillé de mettre un nom de fichier adapté, contenant un ou deux mots-clés, pour chacun des fichiers du site plutôt que des noms du genre page1.html, page2.html, etc.

#### **d-Corps de la page :**

Afin de valoriser au maximum le contenu de chaque page il est nécessaire que celui-ci soit transparent (par opposition aux contenus opaques tels que le flash), c'est-à-dire qu'il comporte un maximum de texte, lisible par les moteurs. Le contenu de la page doit être avant tout un contenu de qualité adressé aux visiteurs, mais il est possible de l'améliorer en veillant à ce que différents mots-clés soient présents.

Les cadres (frames) sont fortement déconseillés car ils empêchent parfois l'indexation du site dans de bonnes conditions.

#### **e-Balises META :**

Les META Tags sont des balises non affichées à insérer en début de document HTML afin de décrire finement le document. Parmi ces balises les balises **keywords** et **description** permettent respectivement d'ajouter des mots-clés et une description décrivant la page, sans les afficher aux visiteurs (par exemple des termes au pluriel, voire avec des fautes d'orthographe volontaires). Il est conseillé d'utiliser le codage HTML pour les caractères accentués et de ne pas dépasser une vingtaine de mots clés.

Le META **robots** permet de décrire le comportement du robot vis-à-vis de la page, notamment indiquer si la page doit être indexée ou non et si le robot est autorisé à suivre les liens.

Etant donné l'usage abusif des métas constaté dans un nombre important de sites web, les moteurs utilisent de moins en moins ces informations lors de l'indexation des pages.

#### **f-Liens hypertextes :**

Afin de donner le maximum de visibilité à chacune de vos pages, il est conseillé d'établir des liens internes entre vos pages pour permettre au robots indexeurs (appelés **crawlers**, traduisez *chenilles*) de parcourir la totalité de votre arborescence. Ainsi il peut être intéressant de créer une page présentant l'architecture de votre site et contenant des pointeurs vers chacune de vos pages.

D'autre part il est important d'avoir des liens externes pointant vers son site web car cela augmente d'une part le trafic et la visibilité de votre site, d'autre part car certains moteurs de recherches prennent en compte le nombre et la qualité des liens pointant vers un site pour caractériser son niveau de pertinence (c'est le cas de Google avec son indice appelé *PageRank*).

#### **g-Attributs ALT des images :**

Les images du site sont opaques pour les moteurs de recherche, c'est-à-dire qu'ils ne sont pas capables d'en indexer le contenu, il est donc conseillé de mettre un attribut *ALT* sur chacune des images, permettant d'en décrire le contenu. La balise ALT est également primordiale pour les non-voyants, navigant à l'aide de terminaux en braille.

Voici un exemple de balise ALT :

```

```

#### **h-Un référencement articulé autour de la page :**

L'élément de référence pour les moteurs de recherche est la page web, il faut donc penser lors de la conception du site web, à structurer les pages en prenant en compte les conseils ci-dessus pour chacune des pages.

En effet la plupart des webmasters pensent à indexer correctement la page d'accueil de leur site mais délaissent les autres pages. Or ce sont généralement les autres pages qui contiennent le contenu le plus intéressant. Il est donc opératif de choisir un titre, une URL et des métas (etc.) adaptés **pour chacune des pages du site**.

#### IV-Sécurisation :

Il existe différentes attaques auxquelles un site web peut être vulnérable :

Tout d'abord il y a la fameuse recherche du répertoire d'administration. Et oui combien de gens installent encore dans leur backoffice dans un dossier admin placé à la base de l'arborescence. Mieux encore, la proie favorite des hackers sont les sites plug & play (CMS) tels que phpnuke. Ainsi on connaît (pour les configurations par défaut) l'emplacement des fichiers de configurations et l'accès à la zone d'administration. Reste plus qu'à trouver le login et le password ou encore une faille de sécurité dans le système.

Une autre vulnérabilité est celle des accès aux bases de données. Je ne parlerai pas du problème de phpmyadmin qui revient à la faille précédente (installation sous /phpMyAdmin connu de tous) mais de la mise en dur dans le code des pages php, jsp ou asp, des paramètres de connexion. En effet il existe des méthodes d'attaques sur les serveurs (surtout sur les IIS avec la méthode .httr) pour récupérer le code source des pages, et ainsi avoir accès à des informations sensibles. Par exemple, pour se prémunir, rien de tel que de créer un fichier ini avec ces données. Ce fichier bien sûr doit être placé dans un répertoire non accessible aux internautes (dans le répertoire précédent WWW) et pour les paranoïaques de crypter les informations qu'il contient.

Une façon de sécuriser son site web est d'utiliser les fichiers .htaccess et .htpasswd si vous utilisez un serveur apache.

En effet ces deux fichiers permettent de limiter l'accès à certains répertoires avec un login et password, mais aussi permettent d'empêcher le listing des fichiers et répertoire du site. Ceci compliquera la tâche du hacker dans la recherche de fichiers.

Enfin on ne le dit jamais assez, mais une manière simple de se protéger c'est d'utiliser des passwords qui sont des passwords. C'est à dire une suite non logique d'au moins huit caractères alphanumériques en utilisant les majuscules et les caractères spéciaux.

En suivant ces règles, votre site internet se trouvera sécurisé. Bien sûr sécurisé ne veut pas dire inviolable, surtout qu'ici nous avons parlé que du côté développement du site internet, il reste encore toute la partie faille de sécurité au niveau serveur, mais cela généralement, relève de votre hébergeur.

*J'ai le grand plaisir de dédier ce travail en témoignage d'affection et de reconnaissance à tous ceux qui me sont chers, particulièrement :*

*A mes chers et adorables parents*

Que ce modeste travail soit le témoignage d'un amour profond pour leurs sacrifices, leur grande confiance, leur patience et leur soutien moral qu'ils nous ont consenti et sans lesquels rien n'était possible.

*A ma femme, ma fille et à mes chères soeurs*

Leurs aides si précieuses et leurs encouragements pour tout l'amour qu'ils ont prodigué.

*A mes sincères amis*

Que ce travail soit le témoignage de l'affection Qu'ils nous porte, pour leur grand amour, leur générosité, leur présence chaleureuse et leurs aides.

*Je prie Dieu pour que ceux qui nous ont soutenus de près ou de loin dans la réalisation de ce document de cours puissent trouver dans ce travail les résultats souhaités.*

*JAZI Marwane*

---