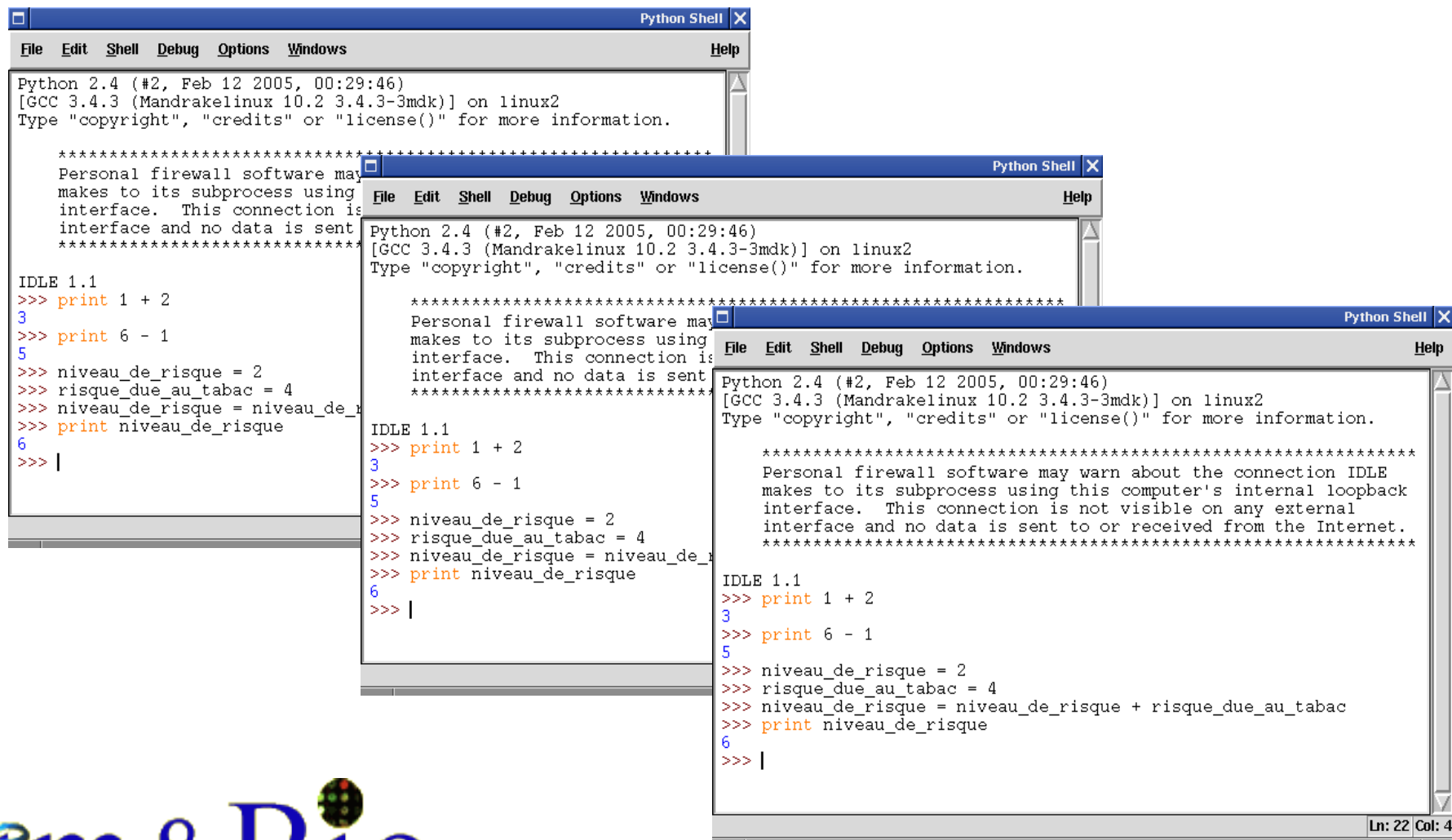


Cours 2 : Python, les bases



The image displays three overlapping screenshots of a Python Shell window, illustrating the execution of a simple Python script. The window title is "Python Shell" and it contains a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help".

The first screenshot shows the initial state of the shell, displaying the Python version (2.4) and the GCC version (3.4.3) on a Linux system. It also shows a warning message from the personal firewall software.

The second screenshot shows the execution of the script, with the output of the print statements visible. The script defines a variable `niveau_de_risque` and prints its value.

The third screenshot shows the execution of the script, with the output of the print statements visible. The script defines a variable `niveau_de_risque` and prints its value.

```
Python 2.4 (#2, Feb 12 2005, 00:29:46)
[GCC 3.4.3 (Mandrakelinux 10.2 3.4.3-3mdk)] on linux2
Type "copyright", "credits" or "license()" for more information.

*****
Personal firewall software may warn about the connection IDLE
makes to its subprocess using this computer's internal loopback
interface. This connection is not visible on any external
interface and no data is sent to or received from the Internet.
*****

IDLE 1.1
>>> print 1 + 2
3
>>> print 6 - 1
5
>>> niveau_de_risque = 2
>>> risque_due_au_tabac = 4
>>> niveau_de_risque = niveau_de_risque + risque_due_au_tabac
>>> print niveau_de_risque
6
>>> |
```

Langage de programmation

- Les êtres humains parlent français, anglais,...
- L'ordinateur parle en binaire
- => nécessité de trouver un langage commun !
 - De nombreux langages de programmation
 - Les principes de base sont communs à tous les langages
 - L'important est de connaître ces principes de base

Python ?

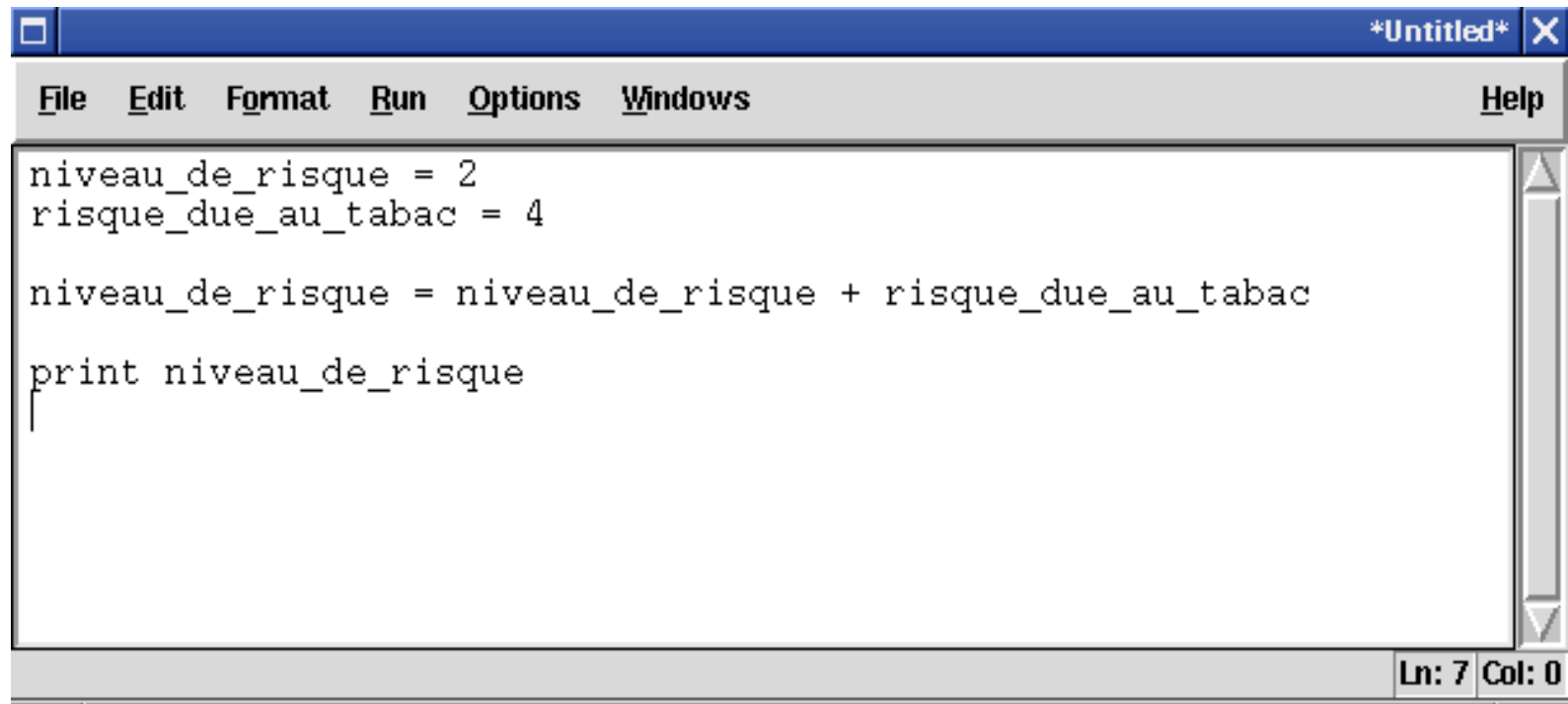
- **Python** est un langage de programmation très facile à apprendre
- Mais ce n'est pas un «**jouet**» : il est vraiment utilisé, y compris par des professionnels
- Langage multi-plateforme (= fonctionne sous Windows, Mac et Unix / Linux)
- **Logiciel libre**
 - Disponible sur <http://python.org>

Python ?

- Programmation **impérative** : le programme donne des ordres à l'ordinateur
 - Écrit cela à l'écran !
 - Calcule ça !
 - Enregistre ça dans ce fichier !
 - Ajoute Mr X dans la base de données des patients !
 - ...
- Le programme se présente sous la forme d'une suite d'ordre, à exécuter les uns après les autres

Comment faire un programme ?

- Deux possibilités
- Les ordres sont placés dans un fichier, puis exécutés
 - Permet d'exécuter plusieurs fois le même programme
 - => pour écrire un logiciel



```
niveau_de_risque = 2
risque_due_au_tabac = 4

niveau_de_risque = niveau_de_risque + risque_due_au_tabac
print niveau_de_risque
```

The screenshot shows a window titled '*Untitled*' with a menu bar containing 'File', 'Edit', 'Format', 'Run', 'Options', 'Windows', and 'Help'. The main text area contains the following Python code:

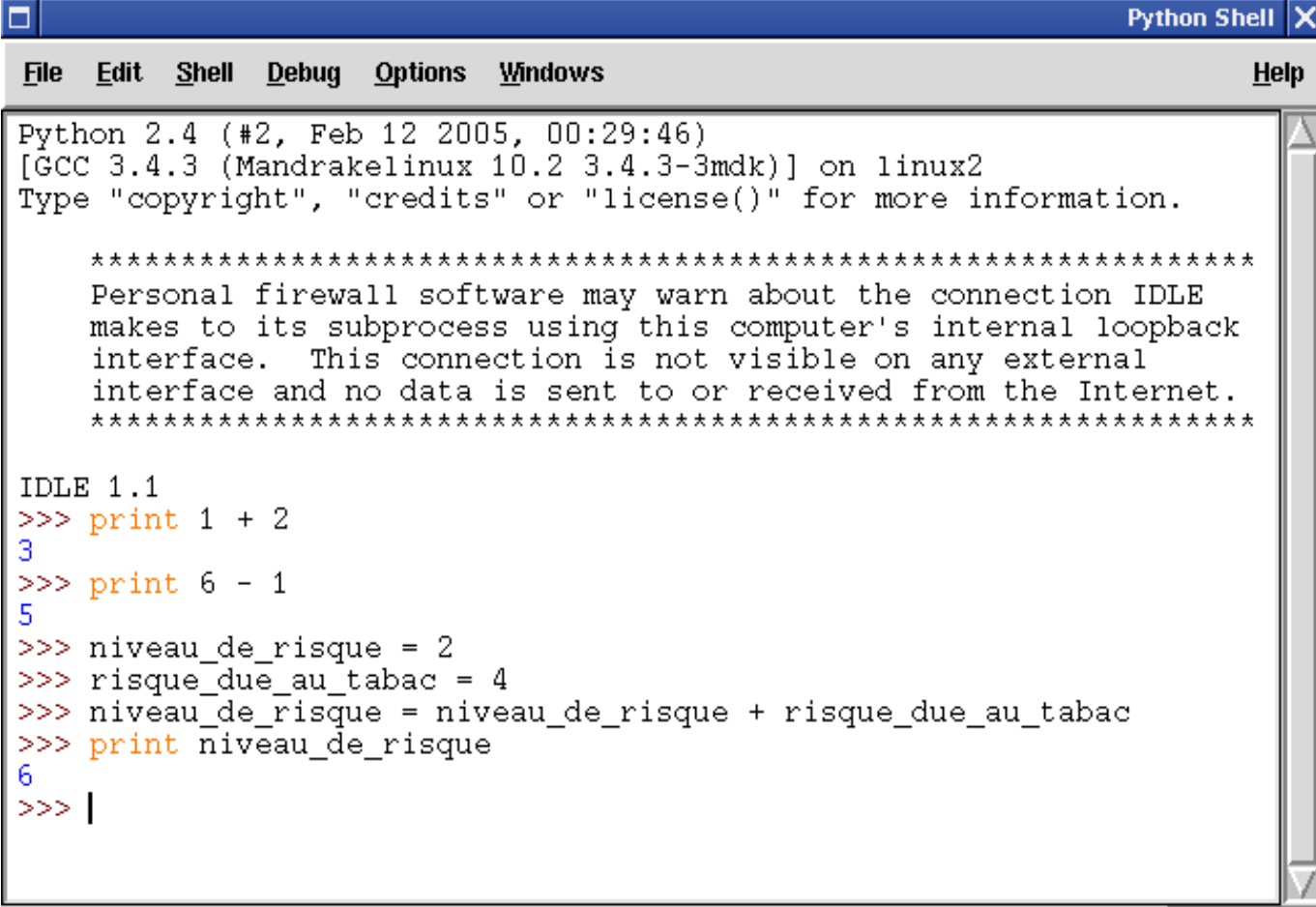
```
niveau_de_risque = 2
risque_due_au_tabac = 4

niveau_de_risque = niveau_de_risque + risque_due_au_tabac
print niveau_de_risque
```

The status bar at the bottom right indicates 'Ln: 7 Col: 0'.

Comment faire un programme ?

- Les ordres sont exécutés **au fur et à mesure** : mode console
 - => pour faire des tests ou «débuguer»



The screenshot shows a window titled "Python Shell" with a menu bar (File, Edit, Shell, Debug, Options, Windows, Help). The main text area displays the following content:

```
Python 2.4 (#2, Feb 12 2005, 00:29:46)
[GCC 3.4.3 (Mandrakelinux 10.2 3.4.3-3mdk)] on linux2
Type "copyright", "credits" or "license()" for more information.

*****
Personal firewall software may warn about the connection IDLE
makes to its subprocess using this computer's internal loopback
interface. This connection is not visible on any external
interface and no data is sent to or received from the Internet.
*****

IDLE 1.1
>>> print 1 + 2
3
>>> print 6 - 1
5
>>> niveau_de_risque = 2
>>> risque_due_au_tabac = 4
>>> niveau_de_risque = niveau_de_risque + risque_due_au_tabac
>>> print niveau_de_risque
6
>>> |
```

At the bottom right of the window, the status bar shows "Ln: 22 Col: 4".

Le langage Python : les bases

- Attention aux majuscules et aux minuscules !
 - Pour Python, un a n'est pas un A !
- Commentaires :
 - # Ce programme a été écrit par JB Lamy !
- Pour écrire une valeur sur l'écran :
 - print valeur*

Types de données

- Les données sont stockées dans la mémoire
- 4 grands **types de données** en Python :
 - Les nombres
 - Les chaînes de caractères (=texte)
 - Les listes
 - Les dictionnaires

Nombres

- Nombres entiers :

0

3

-2

...

- Nombres «flottants» (= décimal = à «virgule»)

0.3

100.009

Nombres

- Opérations :

<code>print 1 + 2</code>	<code>-> 3</code>
<code>print 6 - 1</code>	<code>-> 5</code>
<code>print 2 * 2</code>	<code>-> 4</code>
<code>print 10 / 3</code>	<code>-> 3</code>
<code>print 10.0 / 3</code>	<code>-> 3.3333333333333335</code>

- Parenthèses :

<code>print (3 + 4) * (5 + 8)</code>	<code>-> 91</code>
--------------------------------------	-----------------------

- Exercice :

Comment afficher la valeur de 2 fois 4 ?

Nombres

- Opérations :

<code>print 1 + 2</code>	<code>-> 3</code>
<code>print 6 - 1</code>	<code>-> 5</code>
<code>print 2 * 2</code>	<code>-> 4</code>
<code>print 10 / 3</code>	<code>-> 3</code>
<code>print 10.0 / 3</code>	<code>-> 3.3333333333333335</code>

- Parenthèses :

<code>print (3 + 4) * (5 + 8)</code>	<code>-> 91</code>
--------------------------------------	-----------------------

- Exercice :

Comment afficher la valeur de 2 fois 4 ?

```
print 2 * 4
```

Variables

- Une variable permet de stocker une valeur :
`age = 27`
- Le nom d'une variable contient des lettres, des `_` et des chiffres (mais pas en première position)

`print age` -> 27

`print 2 * age` -> 54

`age = age + 1`

`print age` -> 28

=> la variable age vaut maintenant 28

Variables

- Exercice
 - La surface corporelle est donnée par la formule suivante :
`surface corporelle = poids * taille * taille`
 - Calculer ma surface corporelle en utilisant les variables suivantes :
`poids = 64.0`
`taille = 1.7`

`print poids * taille * taille`
 - J'ai grossi de 2 kilos pendant les fêtes.
Recalculer ma surface corporelle !

Chaînes de caractères

- Les chaînes de caractères (*string*) représentent du texte
- Elles s'écrivent entre guillemet :
`nom_du_patient = "Mr X"`
- Demander à l'utilisateur d'entrer une chaîne :
`nom_du_patient = raw_input("Entrez le nom du patient : ")`
- Opérations :
`nom_du_patient = "Jean" + " " + "Dupont"`
`print nom_du_patient` -> Jean Dupont
`print len(nom_du_patient)` -> 11
`print nom_du_patient[0]` -> J
`print nom_du_patient[-1]` -> t
`print nom_du_patient[0:4]` -> Jean

Chaînes de caractères

- Indices :

J	e	a	n	_	D	u	p	o	n	t
0	1	2	3	4	5	6	7	8	9	10
-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

Chaînes de caractères

- Les chaînes de caractères permettent aussi de représenter des chaînes d'ADN
`adn = "agccgtaggctatttcgacgcaa"`
- Exercice :
 - Comment afficher la chaîne d'ADN ?
 - Comment afficher la longueur de cette chaîne d'ADN ?
 - Comment afficher la première base de la chaîne d'ADN ?
 - Comment ajouter le fragment "tga" au début de l'ADN ?
 - Comment ajouter le fragment "ccc" à la fin de l'ADN ?
 - Comment afficher la longueur de la nouvelle chaîne obtenue ?

Chaînes de caractères

```
adn = "agccgtaggctatttcgacgcaa"
```

- Exercice :

- Comment afficher la chaîne d'ADN ?

```
print adn
```

- Comment afficher la longueur de cette chaîne d'ADN ?

```
print len(adn)
```

- Comment afficher la première base de la chaîne d'ADN ?

```
print adn[0]
```

- Comment ajouter le fragment "tga" au début de l'ADN ?

```
adn = "tga" + adn
```

- Comment ajouter le fragment "ccc" à la fin de l'ADN ?

```
adn = adn + "ccc"
```

- Comment afficher la longueur de la nouvelle chaîne obtenue ?

```
print len(adn)
```

Chaînes de caractères

- Exercice :
 - Faire un programme qui écrit « $3 \times 4 = 12$ » (en faisant en sorte qu'il calcule véritablement la multiplication !)

Chaînes de caractères

- Exercice :
 - Faire un programme qui écrit «3 x 4 = 12» (en faisant en sorte qu'il calcule véritablement la multiplication !)
- ```
print "3 x 4 = ", 3 * 4
```

# Chaînes de caractères

- Attention ! "3" est différent de 3 !

```
print 3 + 3 -> 6
```

```
print "3" + "3" -> "33"
```

- Conversion en entier avec int, en flottant avec float, en chaîne de caractères avec str :

```
print int("3") -> 3
```

```
print float("3") -> 3.0
```

```
print str(3) -> "3"
```

```
age = int(raw_input("Entrez l'âge du patient : "))
```

```
poids = float(raw_input("Entrez le poids du patient : "))
```

# Exercice

- L'indice de masse corporelle d'une personne peut être calculé avec la formule suivante :  
$$\text{imc} = \text{poids} / (\text{taille}^2)$$
- Écrire un programme qui demande à l'utilisateur les paramètres nécessaires, calcule l'indice de masse corporelle et affiche le résultat
- Quels sont les paramètres ? Quels sont leur type ?
- Comment calculer  $\text{taille}^2$  ?

# Exercice

- L'indice de masse corporelle d'une personne peut être calculé avec la formule suivante :
$$\text{imc} = \text{poids} / (\text{taille}^2)$$
- Écrire un programme qui demande à l'utilisateur les paramètres nécessaires, calcule l'indice de masse corporelle et affiche le résultat
- Quels sont les paramètres ? Quels sont leur type ?
- Comment calculer  $\text{taille}^2$  ?

```
poids = float(raw_input("Quel est votre poids (kg) ? "))
taille = float(raw_input("Quel est votre taille (m) ? "))
```

```
imc = poids / (taille * taille)
```

```
print "indice de masse corporelle :", imc
```