

## Exercici 1: Introducció a l'ús de l'ordinador per a programar.

### AQUESTA SESSIÓ NO ÉS AVALUABLE I L'ALUMNE NO HA DE LLIURAR RES.

En aquesta assignatura usarem Python com a llenguatge d'alt nivell per implementar i avaluar les tècniques algorísmiques apreses a les classes magistrals.

Per començar a treballar cal tenir instal·lat el llenguatge Python a l'ordinador. Si treballeu a l'aula d'informàtica de la facultat, els ordinadors ja tenen instal·lat el llenguatge per defecte. Si treballeu amb el vostre ordinador, pot ser que no hi sigui!

### Com s'instal·la Python?

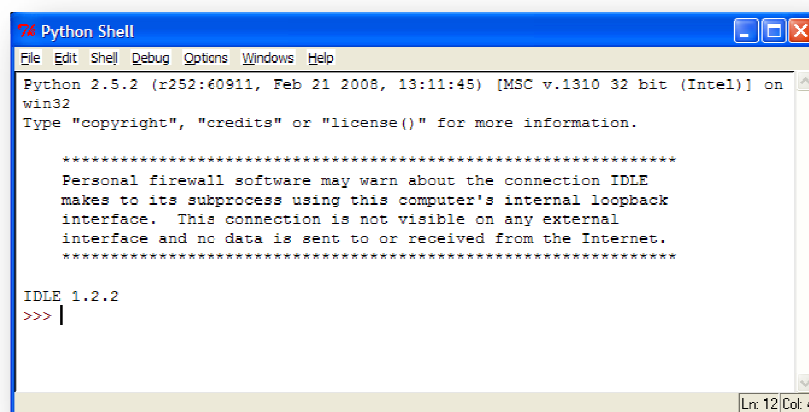
Instal·lar Python és extremadament senzill. Si aneu a l'adreça [www.python.org](http://www.python.org) trobareu un programa d'instal·lació pels sistemes operatius més comuns: Windows, Mac OS X i Linux. Seguiu els passos que us indiquen.

A [www.python.org](http://www.python.org) veureu que hi ha diverses versions de Python instal·lables. Nosaltres usarem la versió 2.6 i no la que aparentment és la darrera, la 3.x. El motiu és que la versió 3.x no és simplement una petita evolució del llenguatge, sinó que hi ha canvis substancials que modifiquen el seu comportament, i la immensa majoria de programadors de Python continuen usant la 2.6: <http://www.python.org/download/releases/2.6.6/>

Un cop Python està instal·lat, només cal cridar-lo i començar a treballar!

### Python i IDLE

Si treballeu en Windows, la forma més directa de treballar en Python és usant l'entorn IDLE. Al menú Start, dins de l'entorn Python 2.6, trobareu l'opció IDLE (Python GUI). Sel·leccioneu aquesta opció i veureu que s'obre una finestra com aquesta:



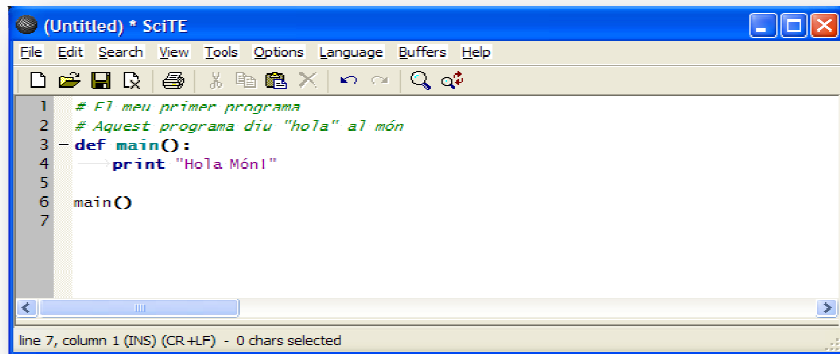
IDLE és el que s'anomena una *shell* (closca). Una *shell* és bàsicament un entorn de finestres que us permet teclejar comandes per comunicar-vos amb un programa.

El símbol `>>>` representa que el programa espera que vosaltres introduïu alguna comanda escrita.

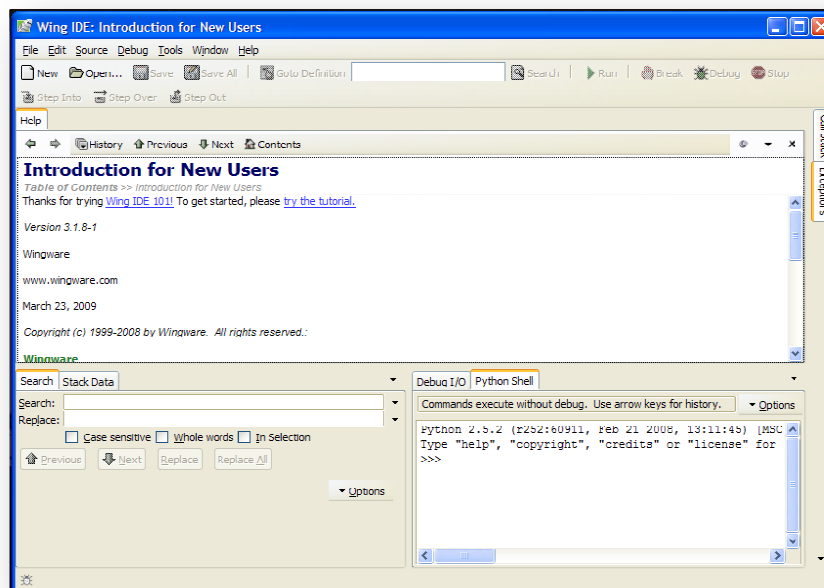
L'interpret de comandes no ens permet escriure programes llargs ni guardar el que anem escrivint, pel que necessitem un editor extern. Per aquesta tasca tenim dues opcions:

1. Usar l'editor que porta integrat IDLE (que és molt simple).
2. Usar qualsevol editor que pugui generar fitxers de Windows.
3. Usar editors especialitzats en escriure programes, pel que són més pràctics que altres editors de caràcter general.

Una possibilitat és l'editor *Scite* ([www.scintilla.org](http://www.scintilla.org)) que és simple però potent.



Una altra possibilitat, potent però més complexa d'usar, és *Wing IDE* ([www.wingware.com](http://www.wingware.com)). La seva principal avantatge a aquestes alçades és que integra en el mateix programa l'editor.



## El primer programa Python

Si escriviu després del símbol `>>>` la instrucció `print "Hola Mon!"` i premeu la tecla `Enter` el programa IDLE us contesta escrivint `Hola Mon!`

```
>>> print "Hola Mon!"
Hola Mon!
>>>
```

D'aquesta manera hem interactuat amb Python directament, sense escriure el programa en un fitxer, i hem obtingut una resposta.

Si el que escrivim no és correcte (Python no ho entén) rebem un missatge d'error:

```
>>> print "Hola
SyntaxError: EOL while scanning single-quoted string
>>>
```

Imaginem ara que volem escriure un programa més llarg, i per tant fer-ho en un fitxer. El programa és aquest:

```
# El meu primer programa
# Aquest programa diu "hola" al món
def main():
    print "Hola Món!"

main()
```

Per executar aquest programa, primer l'heu d'escriure a un fitxer que podeu anomenar "hello.py". L'extensió .py és obligatòria.

Si ho voleu escriure des de IDLE, el que heu de fer és seleccionar l'opció File>New Window i escriure el programa. Al final, heu de salvar amb l'opció File>Save As.

Un cop el fitxer està salvat, podeu executar-lo amb l'opció Run>Run Module de l'editor IDLE.

Si el programa l'heu editat amb un altre editor, el podeu carregar a l'entorn IDLE amb l'opció Open.

**ÉS CONVENIENT QUE SEMPRE GUARDEU ELS EXERCICIS QUE FEU A CLASSE A UNA MEMÒRIA USB PERSONAL, PER EVITAR PROBLEMES AMB EL SERVIDOR DE LA UB.**

## Programant en Python

Ara ja podeu experimentar i fer modificacions al programa:

- Escriu un programa que pregunta a una persona el seu nom, i després la saluda.
- Feu alguns canvis aleatoris al programa i observeu els missatges d'error de Python.
- Quina és la diferència entre un "+" i una "," en una instrucció print? Proveu-ho!
- Escriu un programa que pregunta a dues persones el seu nom, i després les saluda a les dues.
- Mireu d'entendre que fa aquest script. Executeu-lo.

```
#
# un programa numeric
#
def main():
    a = 3 - 4 + 10
    b = 5 * 6
    c = 7.0/8.0
    print "These are the values:", a, b, c
    print "Increment", a, "by one: "
    a = a + 1
    print a
    print "The sum of", a, "and", b, "is"
    d = a + b
    print d
    number = input("Input a number ")
    print number
```

- Escriu un programa que demani un nombre, li sumi 3, llavors multipliqui el resultat per 2, li resti 4, li resti el doble d'això al valor original, sumi 3 i llavors imprimeixi el resultat.

```
# convert.py
# Un programa per pasar de graus Celsius a Fahrenheit
# Escrit per : Aquí el vostre nom.

def main():
    celsius = input("What is the Celsius temperature? ")
    fahrenheit = 9.0 / 5.0 * celsius + 32
    print "The temperature is", fahrenheit, "degrees Fahrenheit."

main()
```

- Modifiqueu el programa convert.py per a que escrigui una presentació del programa. Escriviu-lo en el fitxer convert1.py i executeu-lo.

```
# avg.py
# Càlcul de la nota promig entre dos examens.
# Il·lustra l'ús de l'entrada multiple de Python

def main():
    print "Aquest programa calcula el promig de la nota de dos examens."

    score1, score2 = input("Entra les notes de dos examens separades per una coma: ")
    average = (score1 + score2) / 2.0

    print "El promig es:", average

main()
```

- Modifiqueu el programa avg.py per a que calculi el promig de tres exàmens. Escriviu-lo en el fitxer avg1.py i executeu-lo.