

# Introducere în Python

Python este un limbaj de scripting de nivel înalt, ceea ce înseamnă că este interpretat și nu compilat, economisind mult timp în procesul de dezvoltare și depanare.

Python permite scrierea de programe complexe mult mai rapid și ușor decât în limbaje de genul C sau C++. Programele scrise în Python sunt mult mai mici datorită faptului că:

- tipurile de date de nivel înalt permit realizarea operațiilor complexe într-o singură instrucțiune;
- gruparea instrucțiunilor este dată de paragrafare în loc de blocuri begin/end sau de acolade;
- nu este necesară declararea variabilelor.

Limbajul combină o putere remarcabilă cu o sintaxă foarte clară. Conține module, clase, excepții și tipuri dinamice de nivel înalt. Oferă interfețe pentru o serie de apeluri sistem și librării, precum și la diverse sisteme de ferestre (X11, Motif, Tk, Mac, MFC). În plus, implementarea Python este portabilă: rulează pe Linux, Unix, Windows, OS/2, Mac OS X, Amiga și chiar pe unele telefoane mobile. Python a fost portat, de asemenea, pe mașinile virtuale Java și .NET.

## Alegerea mediului de programare

În Windows există mai multe variante de a instala Python.

- Mediul integrat de dezvoltare: PyCharm (<https://www.jetbrains.com/pycharm/>)
- ActivePython
- **Spyder**
- Jupyter notebook

## Instalarea versiunii oficiale Python (<http://www.python.org/>)

1. Descărcați ultima versiune a installer-ului Python pentru Windows de la adresa <http://www.python.org/>

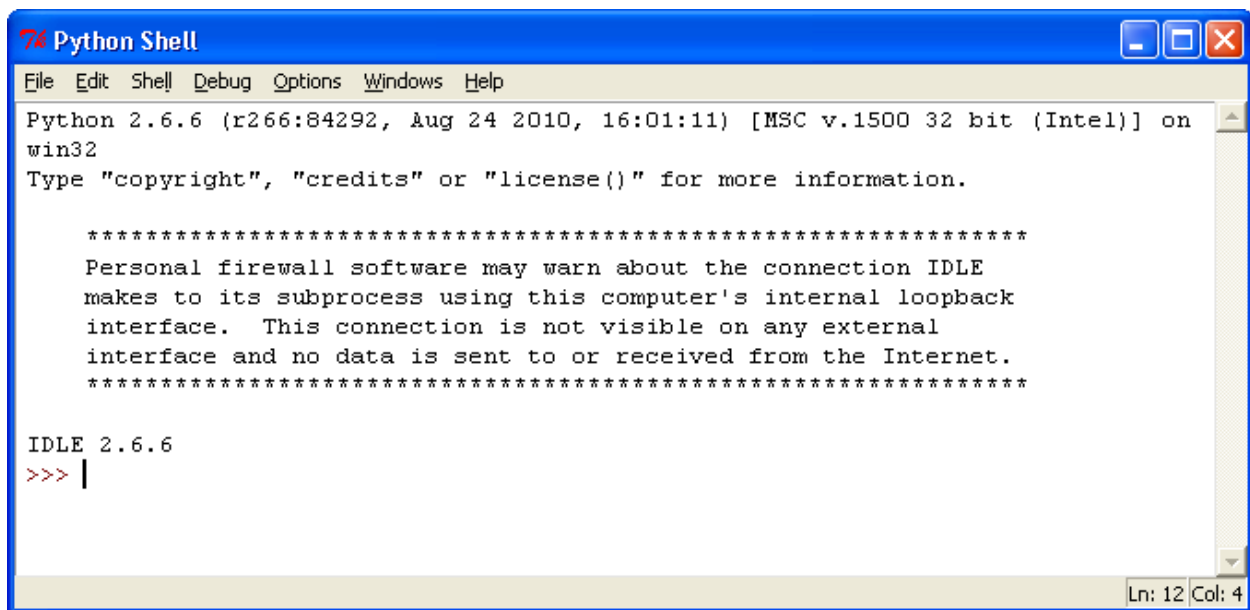
2. Dublu click pe fișierul executabil

3. Uurmați pașii de instalare, lăsând opțiunile implicite.

4. După finalizarea instalării, deschideți editorul Python selectând **Start**→**Programs**→**Python**

### 2.6→IDLE (Python GUI).

5. Va apărea interpretorul interactiv Python:



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 2.6.6 (r266:84292, Aug 24 2010, 16:01:11) [MSC v.1500 32 bit (Intel)] on
win32
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 2.6.6
>>> |
```

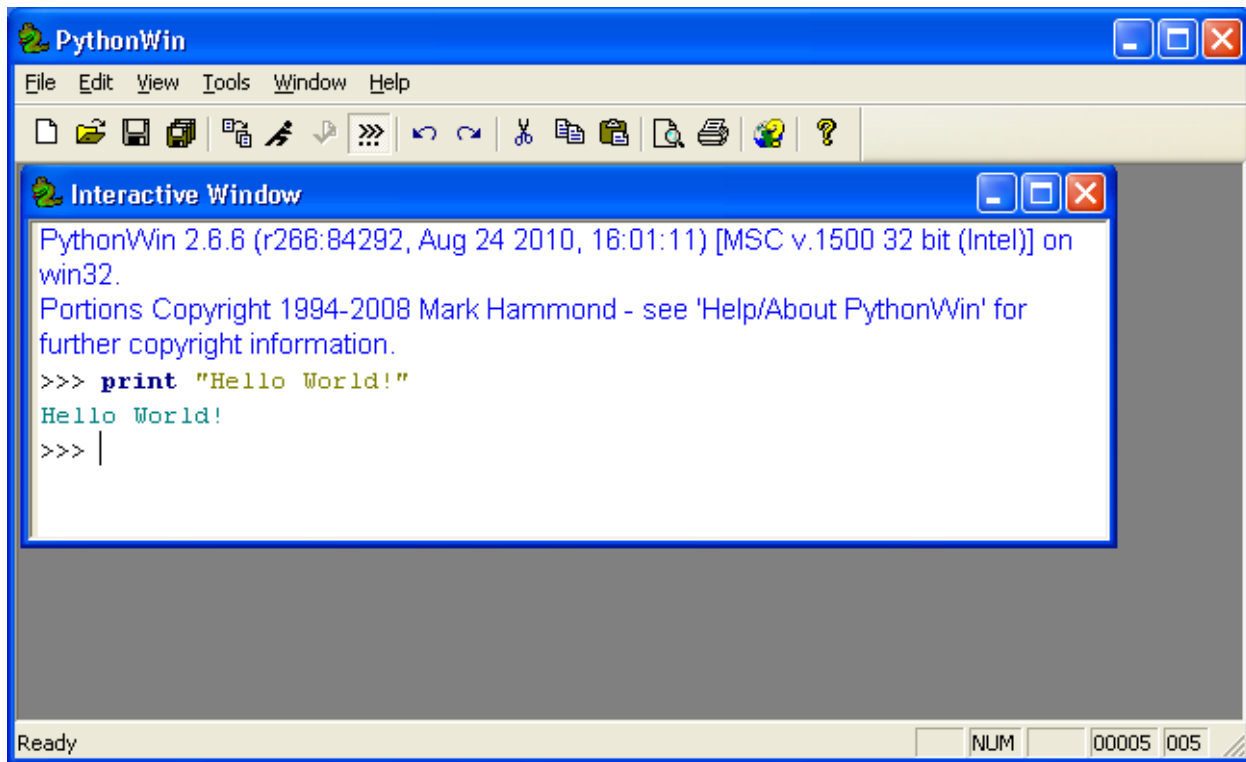
## 1. Utilizarea ferestrei interactive ActivePython

În cadrul interpretorului există două prompt-uri: *prompter-ul principal* (de obicei „>>>”) și *prompter-ul secundar* (de obicei „...”), folosit pentru continuarea liniilor. Interpretorul afișează imediat după pornire un mesaj de întâmpinare în care se specifică versiunea și autorii Python-ului, urmate de primul prompter : „>>>”

În cadrul ferestrei interactive PythonWin pot fi introduse linii singulare de cod.

**Exemplu:** introduceți următoarea instrucțiune, după care apăsați Enter:

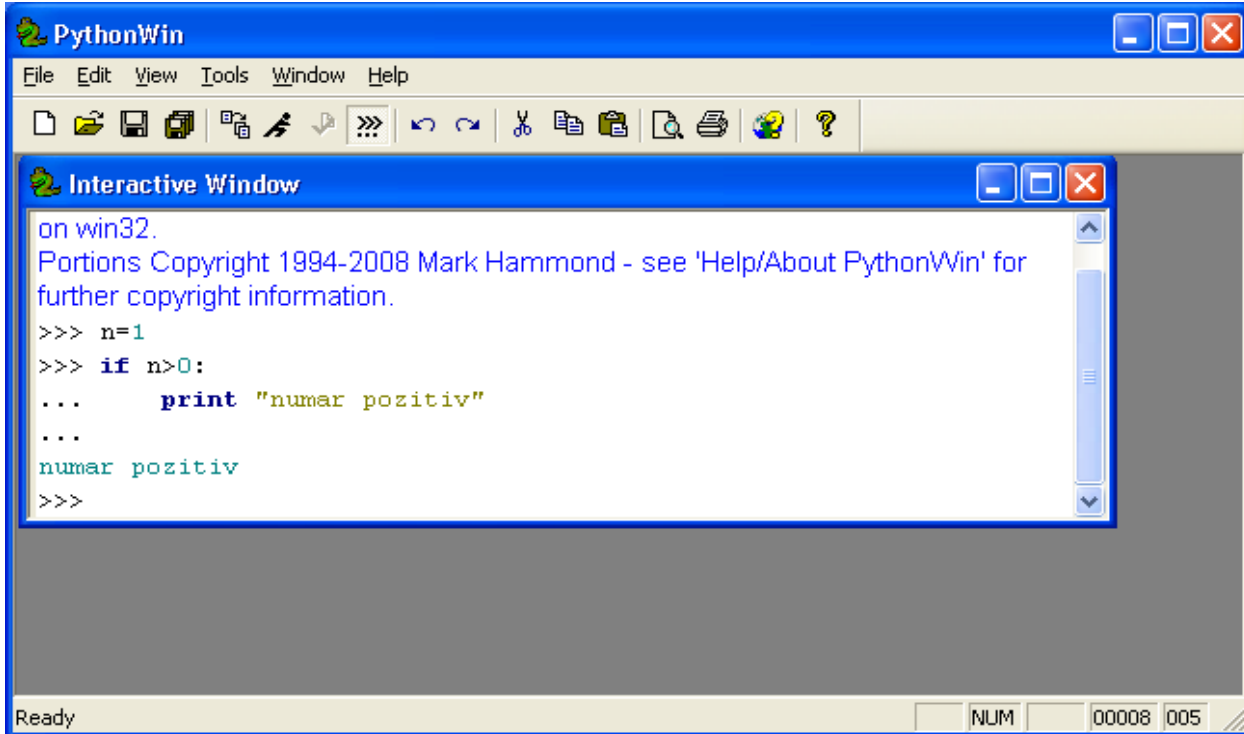
```
>>> print "Hello World!"
```



Se observă afișarea cuvintelor '**Hello World!**'. Interpretorul compilează imediat instrucțiunile introduse. Acest lucru este extrem de util pentru testare, sau pentru a verifica dacă o anumită linie de cod funcționează.

Liniile de continuare sunt folosite la introducerea unei construcții multilinie.

## Exemplu:



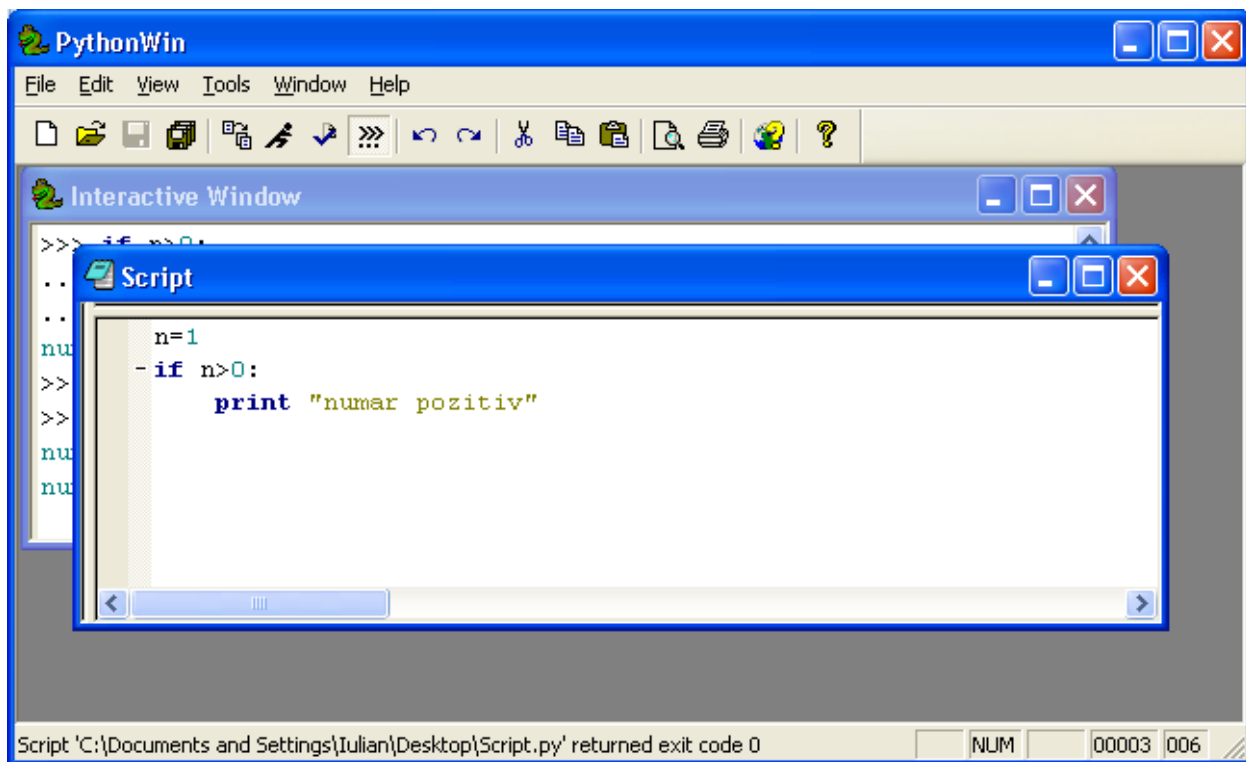
The screenshot shows the PythonWin application window. The title bar reads "PythonWin". The menu bar includes "File", "Edit", "View", "Tools", "Window", and "Help". The toolbar contains various icons for file operations and editing. An "Interactive Window" is open, displaying the following text:

```
on win32.  
Portions Copyright 1994-2008 Mark Hammond - see 'Help/About PythonWin' for  
further copyright information.  
>>> n=1  
>>> if n>0:  
...     print "numar pozitiv"  
...  
numar pozitiv  
>>>
```

The status bar at the bottom shows "Ready" and a numeric keypad with "NUM", "00008", and "005".

**PythonWin** poate fi utilizat și pentru a crea scripturi Python, nu doar în linie de comandă. Pentru aceasta, se creează un document nou accesând **File > New**.

**Exemplu:** creați următorul script:



Salvați fișierul cu numele **Script**. Scripturile Python au extensia **py**.

Pentru a rula programul: click **File>Run>OK**. Programul va rula în ecranul principal.

Intrările și ieșirile vor fi deosebite prin prezența sau absența prompt-urilor (principal '>>>' sau secundar '...'). Liniile care nu au caracterele '>>>' sau '...' în față sunt linii de ieșire, rezultate în urma comenzilor anterioare, sau a ultimei comenzi.

În Python comentariile încep cu un caracter '#' și se termină la sfârșitul liniei. Un comentariu poate apărea la sfârșitul unei linii, dar nu poate apărea în cadrul unui string. Un caracter '#' în cadrul unui string nu reprezintă începutul unui comentariu ci doar un caracter '#'

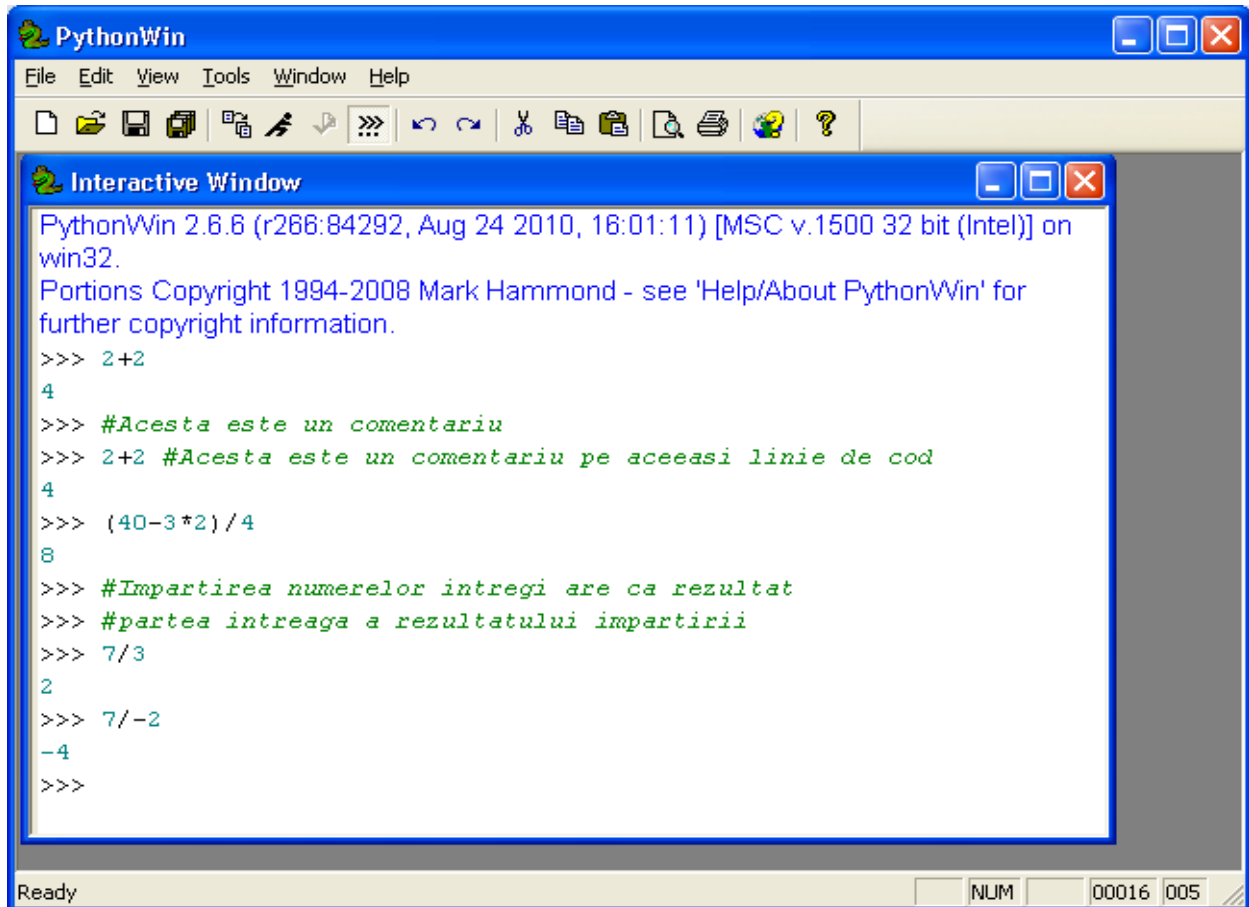
### **Exemplu:**

```
# primul comentariu
n = 1 #al doilea comentariu
# ... al treilea comentariu
str = "# Acesta nu este un comentariu."
```

## Calculare aritmetice

Interpretorul funcționează ca un simplu calculator: puteți scrie o expresie, și ca răspuns veți primi rezultatul. Ordinea operațiilor și utilizarea parantezelor se face la fel ca în majoritatea limbajelor (ex: Pascal sau C).

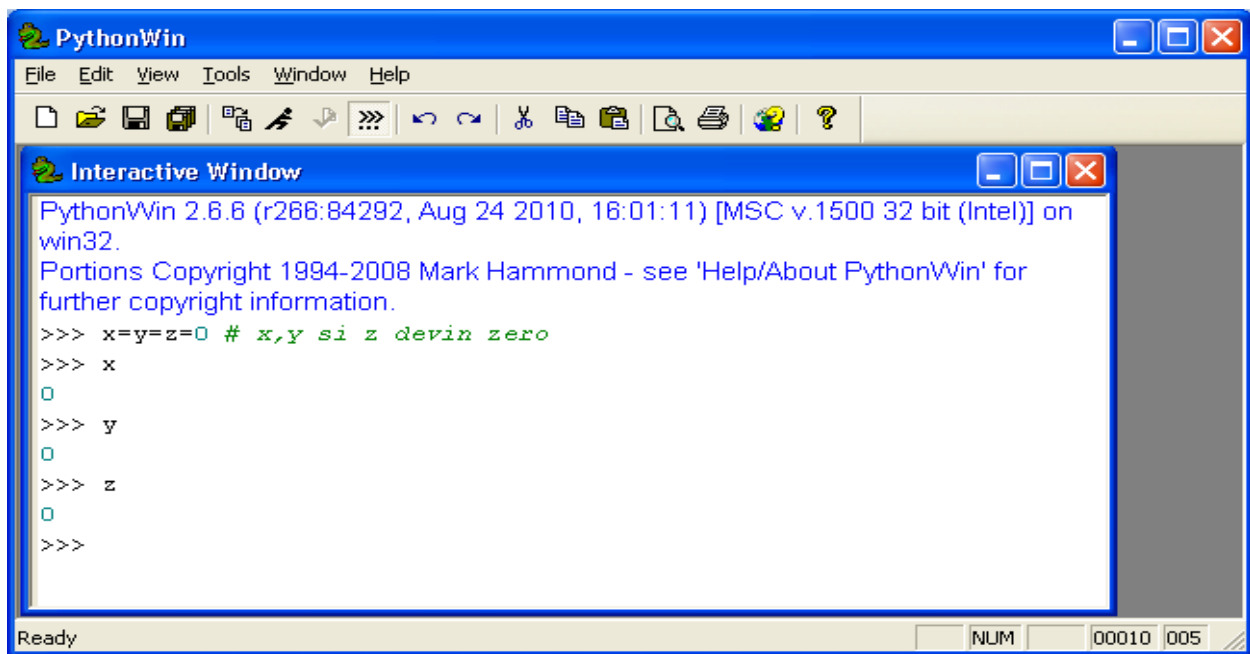
### Exemplu:



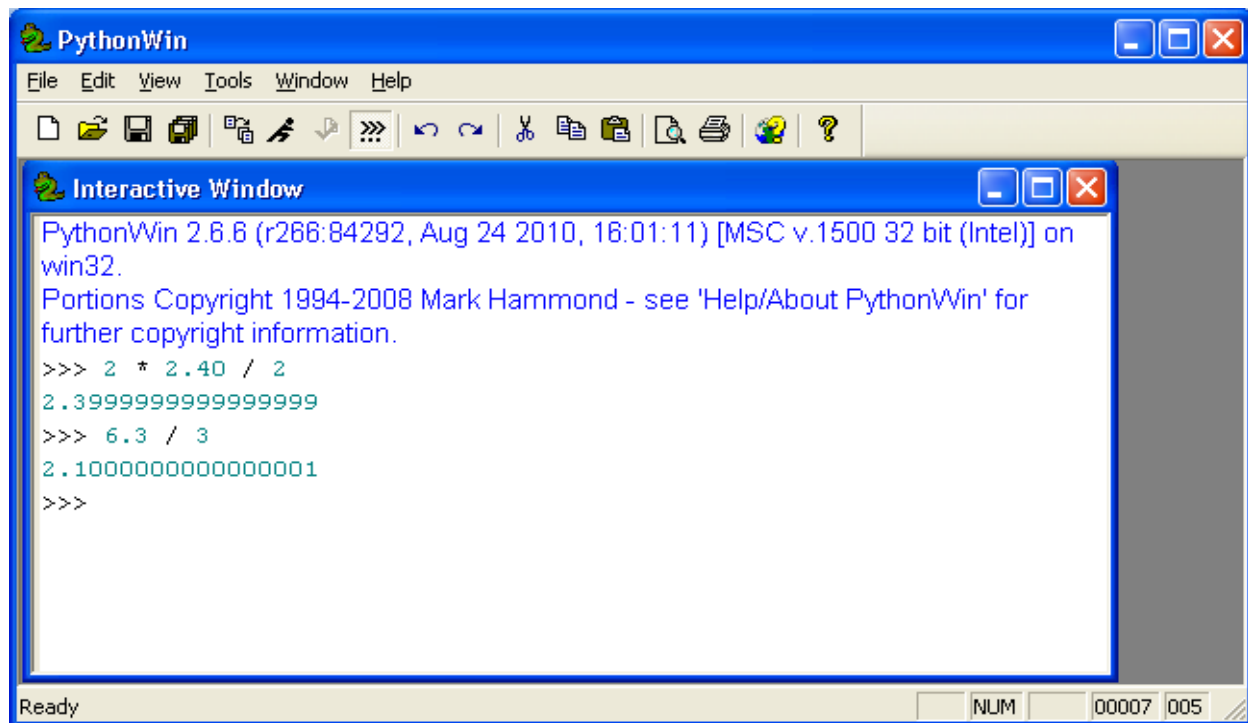
```
PythonWin 2.6.6 (r266:84292, Aug 24 2010, 16:01:11) [MSC v.1500 32 bit (Intel)] on win32.
Portions Copyright 1994-2008 Mark Hammond - see 'Help/About PythonWin' for further copyright information.
>>> 2+2
4
>>> #Acesta este un comentariu
>>> 2+2 #Acesta este un comentariu pe aceeași linie de cod
4
>>> (40-3*2)/4
8
>>> #Impartirea numerelor intregi are ca rezultat
>>> #partea intreaga a rezultatului impartirii
>>> 7/3
2
>>> 7/-2
-4
>>>
```

O valoare poate fi atribuită simultan mai multor variabile.

### Exemplu:

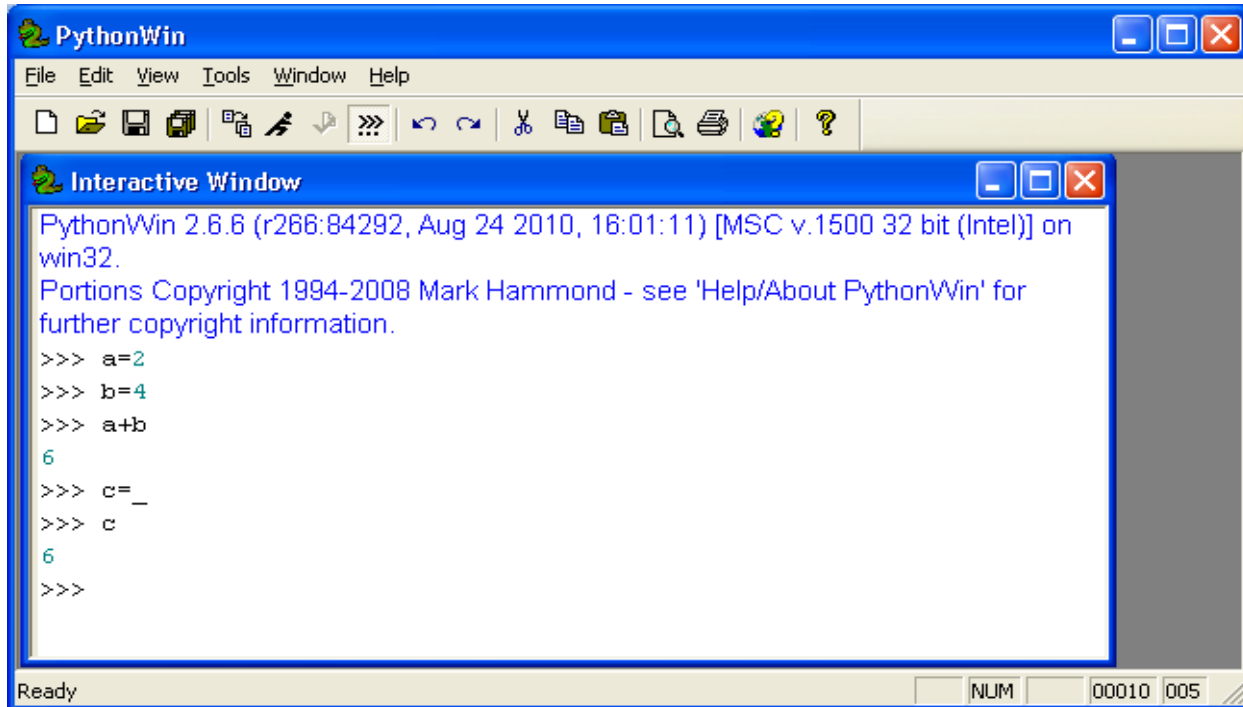


În Python, există suport complet pentru virgula mobilă. Operatorii care au operanzi de tipuri diferite vor converti operandul întreg în virgulă mobilă:



În modul interactiv, ultima valoare afișată este atribuită unei variabile `_`. Dacă folosiți Python ca un calculator de birou este mult mai ușor să păstrați continuitatea calculelor.

## Exemplu:



The screenshot shows the PythonWin application window. The title bar reads 'PythonWin'. The menu bar includes 'File', 'Edit', 'View', 'Tools', 'Window', and 'Help'. The toolbar contains various icons for file operations and execution. The main window is titled 'Interactive Window' and contains the following text:

```
PythonWin 2.6.6 (r266:84292, Aug 24 2010, 16:01:11) [MSC v.1500 32 bit (Intel)] on win32.  
Portions Copyright 1994-2008 Mark Hammond - see 'Help/About PythonWin' for further copyright information.  
  
>>> a=2  
>>> b=4  
>>> a+b  
6  
>>> c=_  
>>> c  
6  
>>>
```

The status bar at the bottom shows 'Ready' and a numeric keypad with 'NUM', '00010', and '005'.

Această variabilă trebuie tratată de utilizator ca având numai capacitatea de a fi citită, nu și scrisă (READ-ONLY).

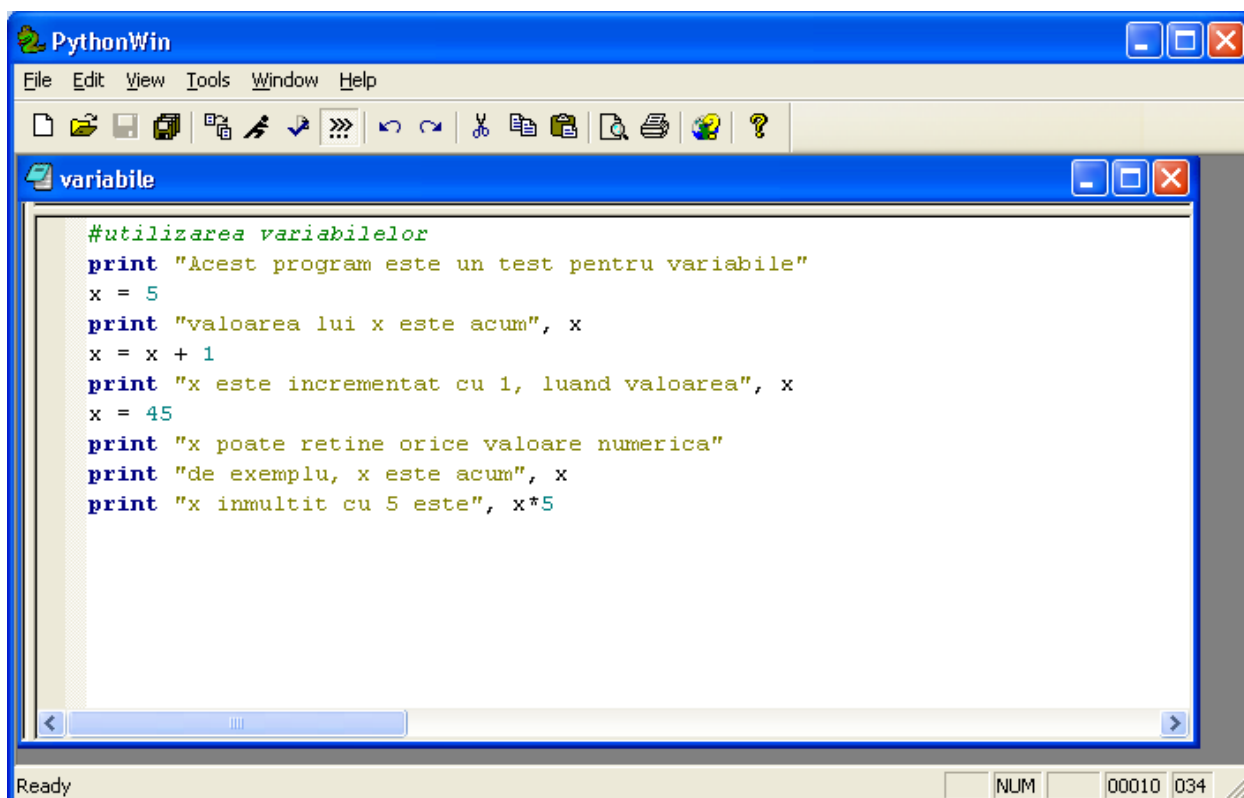
## Operatorii aritmetici

- Adunarea: +
- Scaderea: -
- Inmultirea: \*
- Ridicare la putere: \*\*
- Impartire: /
- Restul impartirii: %

## Variabile

Introduceți următorul cod într-un script cu numele **variabile.py**:





The image shows a screenshot of a PythonWin application window. The window title is "PythonWin". The menu bar includes "File", "Edit", "View", "Tools", "Window", and "Help". The toolbar contains icons for file operations (New, Open, Save, Print, Copy, Paste, Undo, Redo), search, and help. The main text area is titled "variabile" and contains the following Python code:

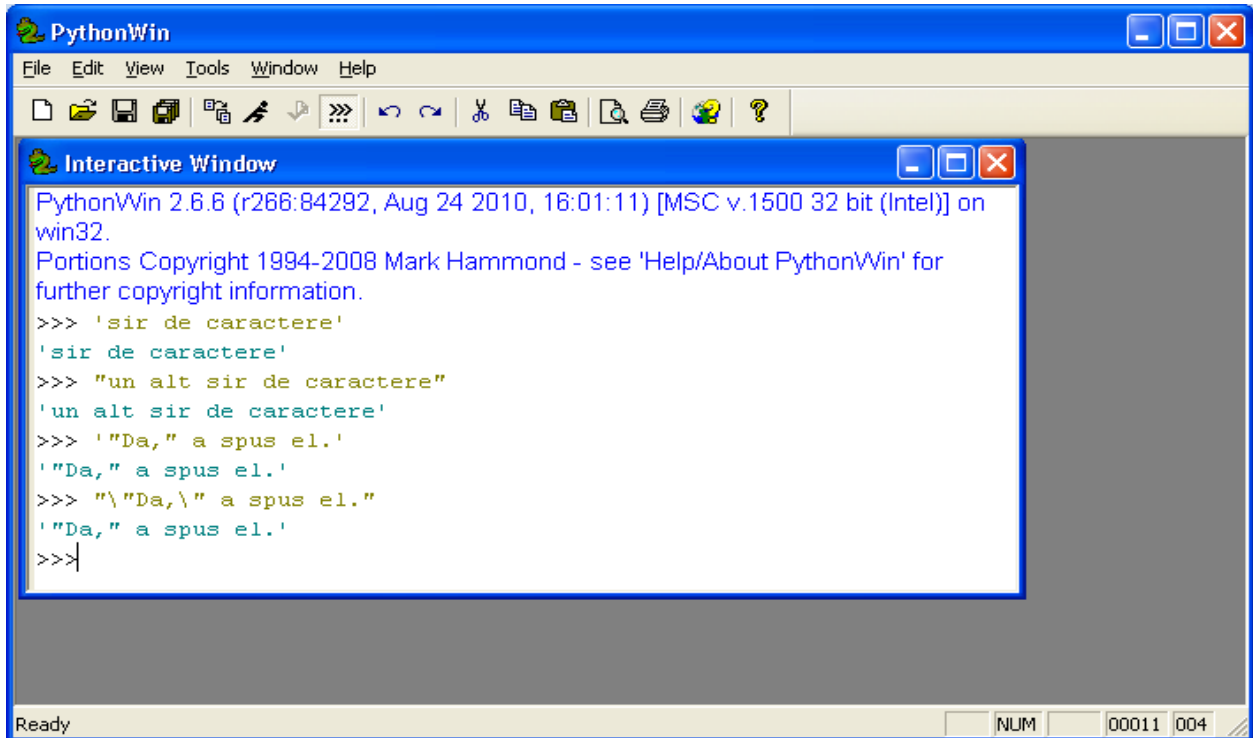
```
#utilizarea variabilelor
print "Acest program este un test pentru variabile"
x = 5
print "valoarea lui x este acum", x
x = x + 1
print "x este incrementat cu 1, luand valoarea", x
x = 45
print "x poate retine orice valoare numerica"
print "de exemplu, x este acum", x
print "x inmultit cu 5 este", x*5
```

The status bar at the bottom left shows "Ready". The bottom right shows "NUM" and "00010 034".

## 2. Șiruri de caractere

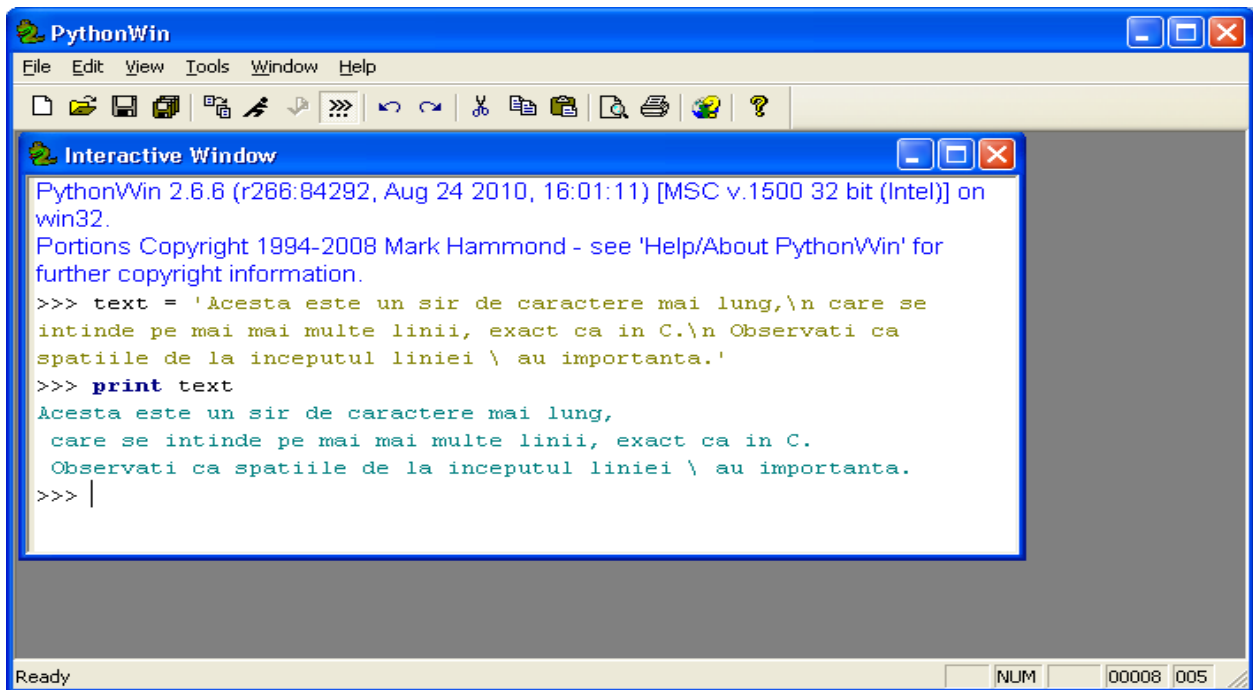
În afară de numere, Python poate manipula și șiruri de caractere, care pot avea mai multe forme. Pot fi incluse între apostroafe sau ghilimele:

### Exemplu:



```
PythonWin 2.6.6 (r266:84292, Aug 24 2010, 16:01:11) [MSC v.1500 32 bit (Intel)] on win32.
Portions Copyright 1994-2008 Mark Hammond - see 'Help/About PythonWin' for further copyright information.
>>> 'sir de caractere'
'sir de caractere'
>>> "un alt sir de caractere"
'un alt sir de caractere'
>>> '"Da," a spus el.'
'"Da," a spus el.'
>>> "\ "Da,\ " a spus el."
'"Da," a spus el.'
>>>
```

Șirurile de caractere pot exista pe mai multe linii, separarea realizându-se prin secvența de control `'\n'`, indicând astfel că linia următoare este continuarea logică a liniei curente:

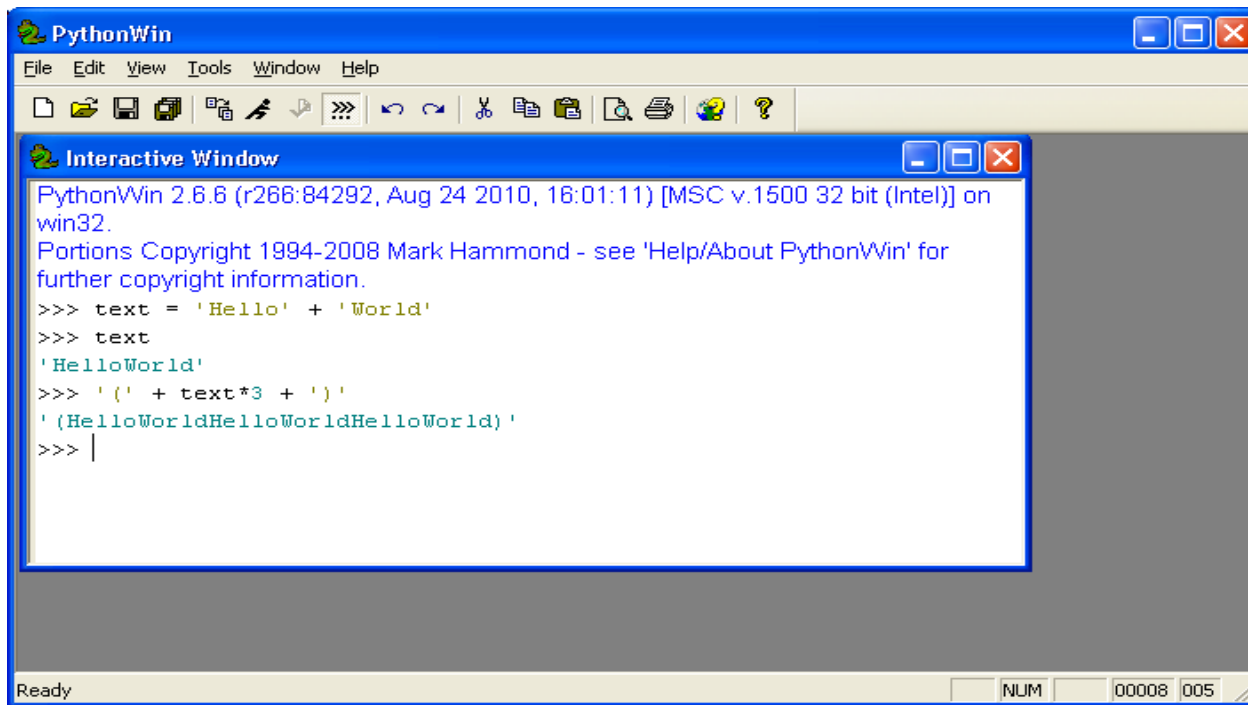


The screenshot shows the PythonWin application window. The title bar reads "PythonWin". The menu bar includes "File", "Edit", "View", "Tools", "Window", and "Help". The toolbar contains various icons for file operations and editing. The main window displays an "Interactive Window" with the following text:

```
PythonWin 2.6.6 (r266:84292, Aug 24 2010, 16:01:11) [MSC v.1500 32 bit (Intel)] on win32.
Portions Copyright 1994-2008 Mark Hammond - see 'Help/About PythonWin' for further copyright information.
>>> text = 'Acesta este un sir de caractere mai lung,\n care se intinde pe mai mai multe linii, exact ca in C.\n Observati ca spatiile de la inceputul liniei \ au importanta.'
>>> print text
Acesta este un sir de caractere mai lung,
 care se intinde pe mai mai multe linii, exact ca in C.
 Observati ca spatiile de la inceputul liniei \ au importanta.
>>> |
```

The status bar at the bottom shows "Ready", "NUM", "00008", and "005".

Șirurile de caractere pot fi concatenate:



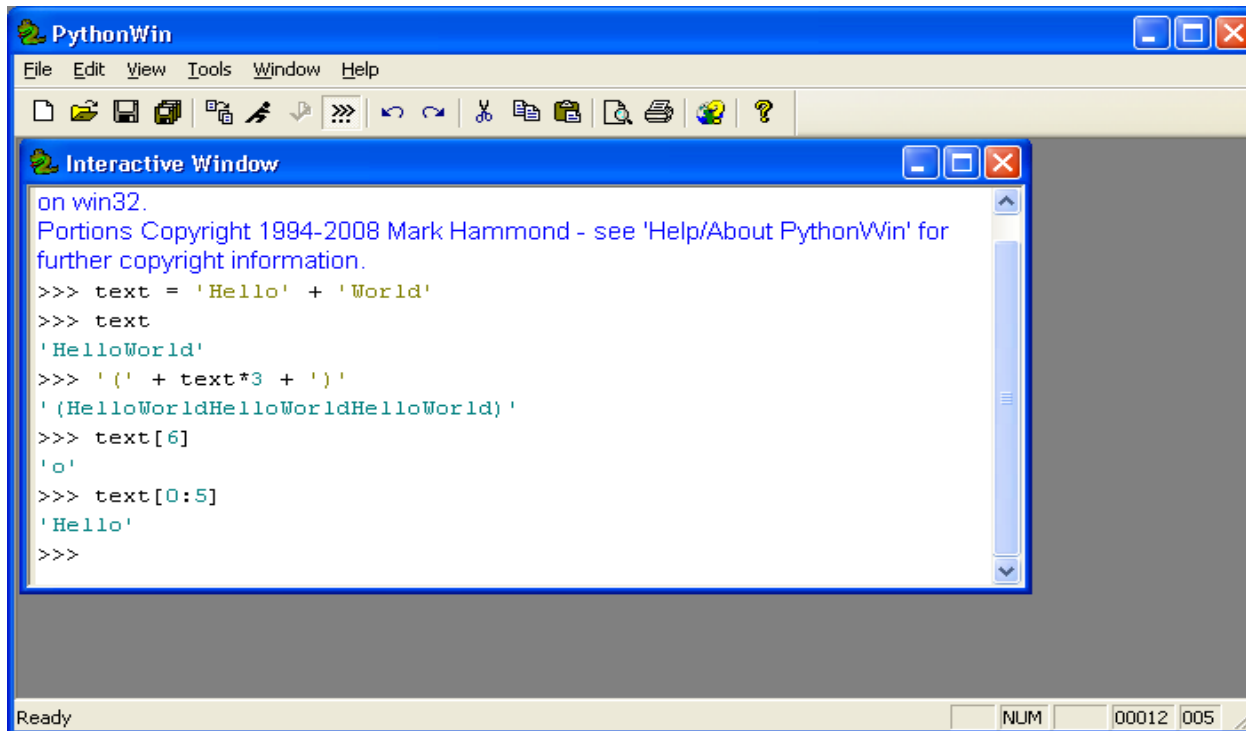
The screenshot shows the PythonWin application window. The title bar reads "PythonWin". The menu bar includes "File", "Edit", "View", "Tools", "Window", and "Help". The toolbar contains various icons for file operations and editing. The main window displays an "Interactive Window" with the following text:

```
PythonWin 2.6.6 (r266:84292, Aug 24 2010, 16:01:11) [MSC v.1500 32 bit (Intel)] on win32.
Portions Copyright 1994-2008 Mark Hammond - see 'Help/About PythonWin' for further copyright information.
>>> text = 'Hello' + 'World'
>>> text
'HelloWorld'
>>> '(' + text*3 + ')'
'(HelloWorldHelloWorldHelloWorld)'
>>> |
```

The status bar at the bottom shows "Ready", "NUM", "00008", and "005".

Șirurile de caractere sunt tratate ca vectori, la fel ca în C, primul caracter al unui șir are indicele 0. Nu există un tip de date separat pentru caracter. Un singur caracter este reprezentat ca un șir de lungime unu. Se pot specifica de asemenea subșiruri ale unui șir folosind doi indici, separați prin ':'.  
'[start:end]'

Primul indice reprezintă poziția de început a subșirului, iar cel de-al doilea indice, indicele în șirul principal la care se termină subșirul:



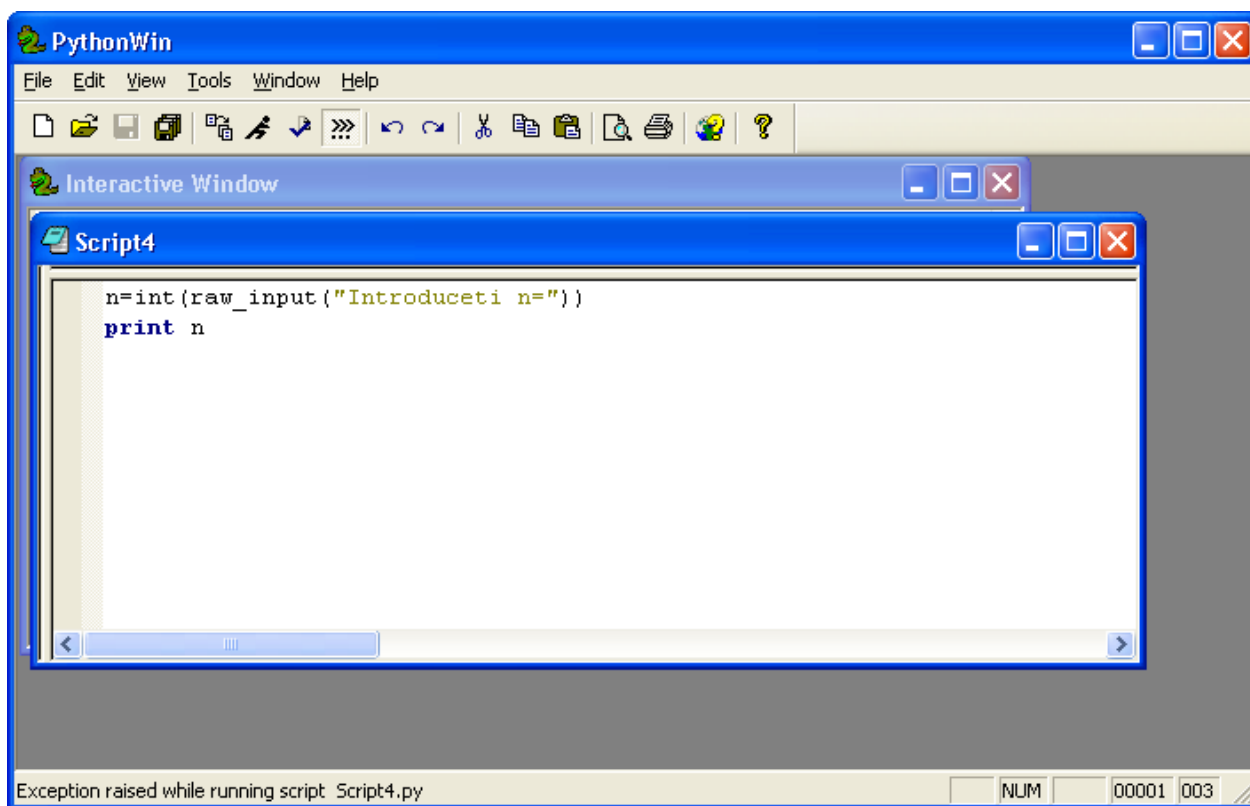
The screenshot shows the PythonWin application window. The title bar reads 'PythonWin'. The menu bar includes 'File', 'Edit', 'View', 'Tools', 'Window', and 'Help'. The toolbar contains various icons for file operations and editing. The main area is an 'Interactive Window' with a blue border. It displays the following Python code and its output:

```
on win32.  
Portions Copyright 1994-2008 Mark Hammond - see 'Help/About PythonWin' for  
further copyright information.  
>>> text = 'Hello' + 'World'  
>>> text  
'HelloWorld'  
>>> '(' + text*3 + ')'  
'(HelloWorldHelloWorldHelloWorld)'  
>>> text[6]  
'o'  
>>> text[0:5]  
'Hello'  
>>>
```

At the bottom of the window, the status bar shows 'Ready' on the left and 'NUM 00012 005' on the right.

### 3. Citirea datelor de la tastatură

Python permite citirea parametrilor dintr-o fereastră de dialog, folosind funcția `raw_input()/input()`. Exemplul următor permite citirea de la tastatură a unui număr întreg:



## 4. Structuri de control

### 6.1 Instrucțiunea if

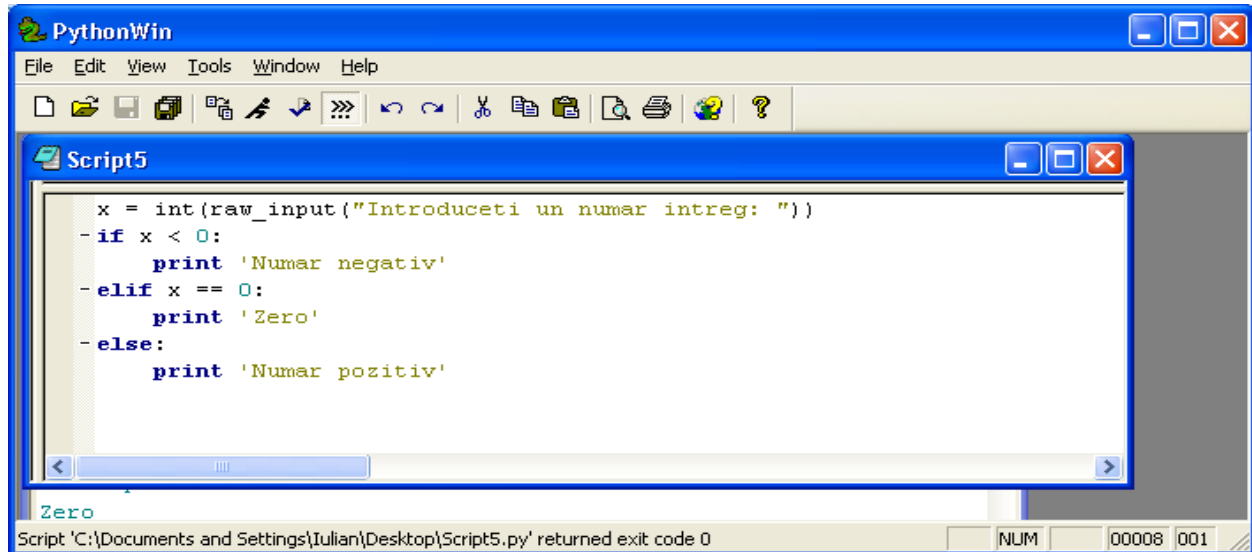
Sintaxa instrucțiunii If este:

```
if conditie1:
    instructiuni
elif conditie2:
    instructiuni
else:
    instructiuni
```

Pot exista mai multe sectiuni *elif* sau nici una, iar sectiunea *else* poate apărea o singură dată sau niciodată.

Numarul 0, listele și tuplurile goale, stringurile vide și valoarea **None**, sunt considerate false daca sunt folosite în evaluarea unei condiții.

## Exemplu:

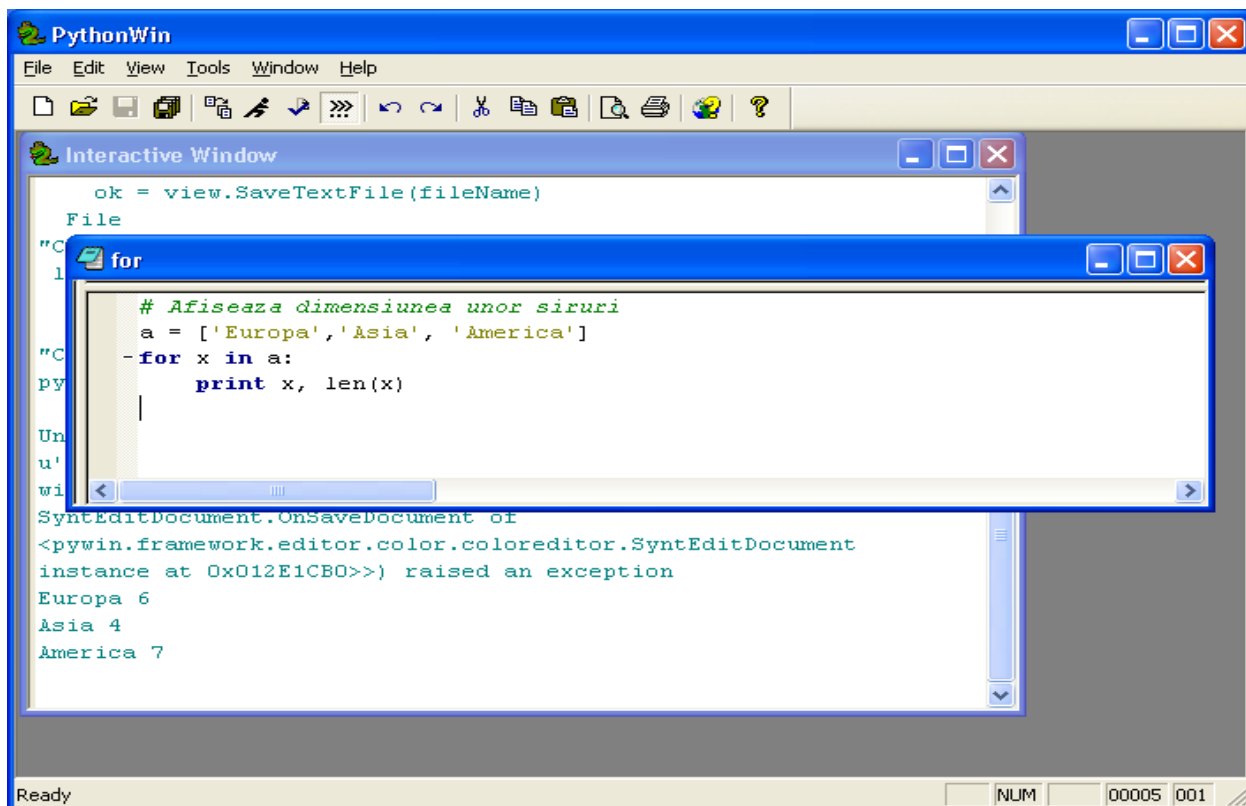


```
PythonWin
File Edit View Tools Window Help
Script5
x = int(raw_input("Introduceti un numar intreg: "))
-if x < 0:
    print 'Numar negativ'
-elif x == 0:
    print 'Zero'
-else:
    print 'Numar pozitiv'
Zero
Script 'C:\Documents and Settings\Iulian\Desktop\Script5.py' returned exit code 0
```

Scriptul de mai sus citește de la tastatură un întreg, testează dacă numărul introdus este pozitiv, negativ sau egal cu zero și afișează un mesaj ca atare.

## 6.2 Instrucțiunea for

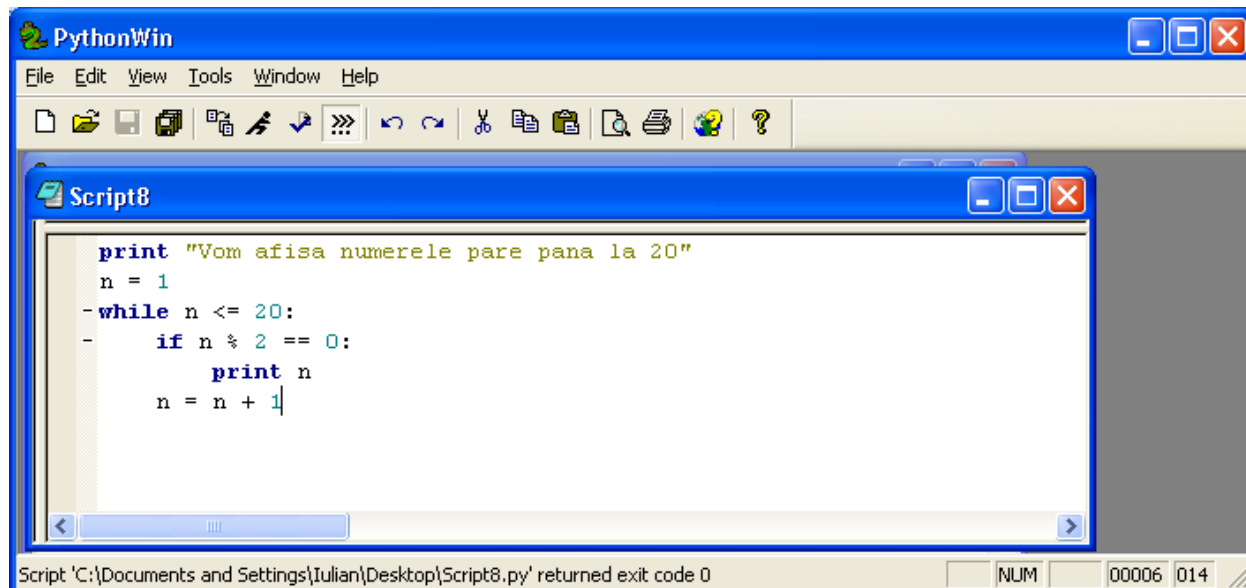
Instrucțiunea **for**, din Python, diferă un pic față de cea din C sau Pascal. În loc de o iterație dată de o progresie aritmetică (Pascal), sau de o iterație foarte flexibilă, pentru care programatorul poate defini atât pasul iterației, cât și condiția de oprire (C), iterațiile instrucțiunii Python **for** funcționează după elementele unei secvențe (șir sau listă).



### 6.3 Instrucțiunea While

Bucula **while** se execută atâta timp cât condiția este adevărată. La fel ca în C, zero înseamnă fals, și orice număr diferit de zero înseamnă adevărat. Condiția poate fi un șir, un element de listă, absolut orice. Orice secvență de lungime diferită de zero înseamnă adevărat, și invers.

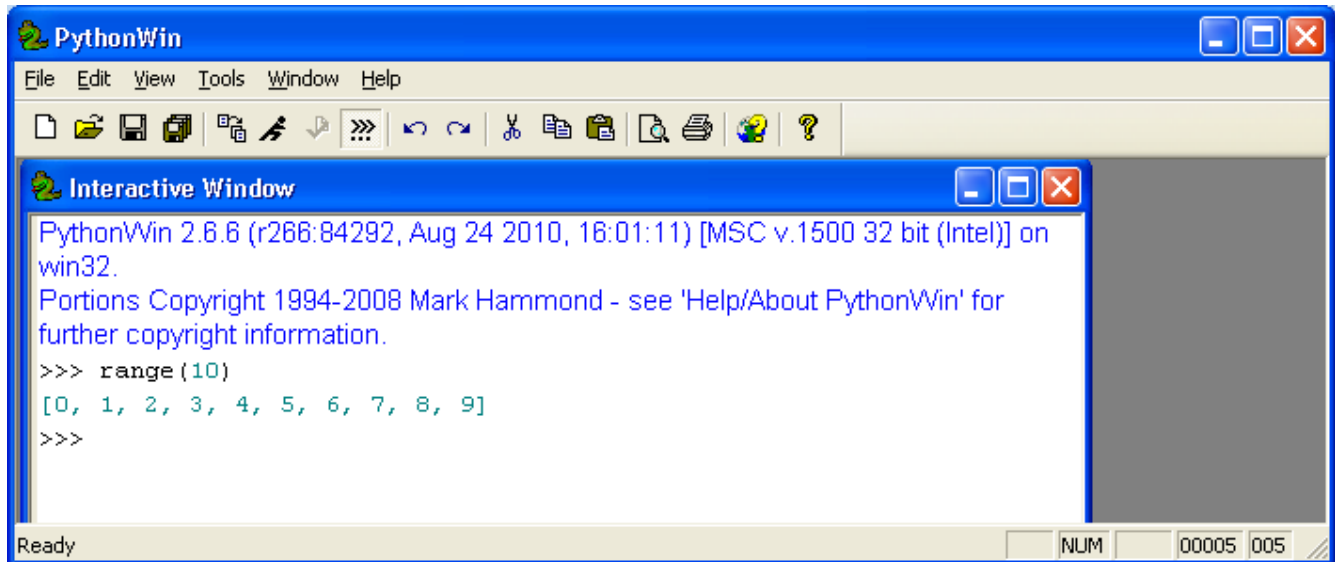
#### Exemplu:



## 6.4 Funcția range()

Dacă este necesară o iterație pe o mulțime de numere, puteți folosi funcția **range** pentru a genera liste ce contin progresii aritmetice.

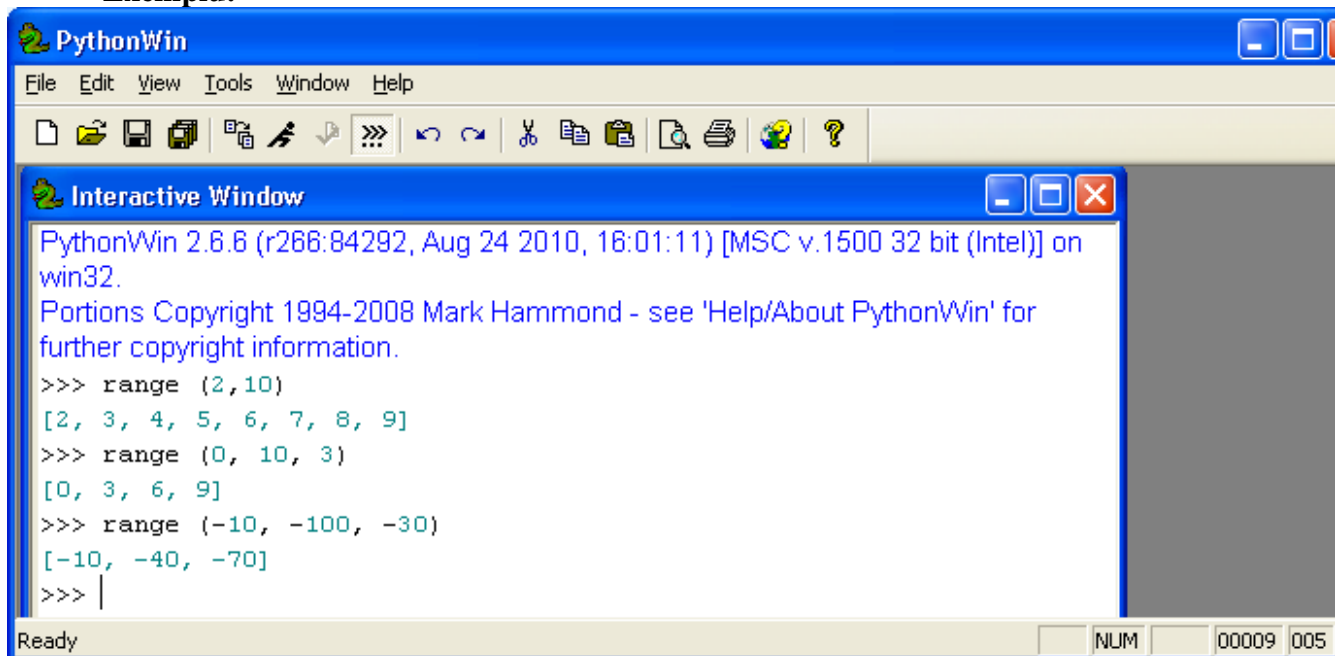
### Exemplu:



```
PythonWin 2.6.6 (r266:84292, Aug 24 2010, 16:01:11) [MSC v.1500 32 bit (Intel)] on win32.
Portions Copyright 1994-2008 Mark Hammond - see 'Help/About PythonWin' for further copyright information.
>>> range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>>
```

Parametrul furnizat funcției **range()** nu va fi niciodată un membru al secvenței. Este posibil ca funcția **range()** să genereze o secvență începând cu un alt număr decât 0, sau rația progresiei aritmetice poate fi modificată:

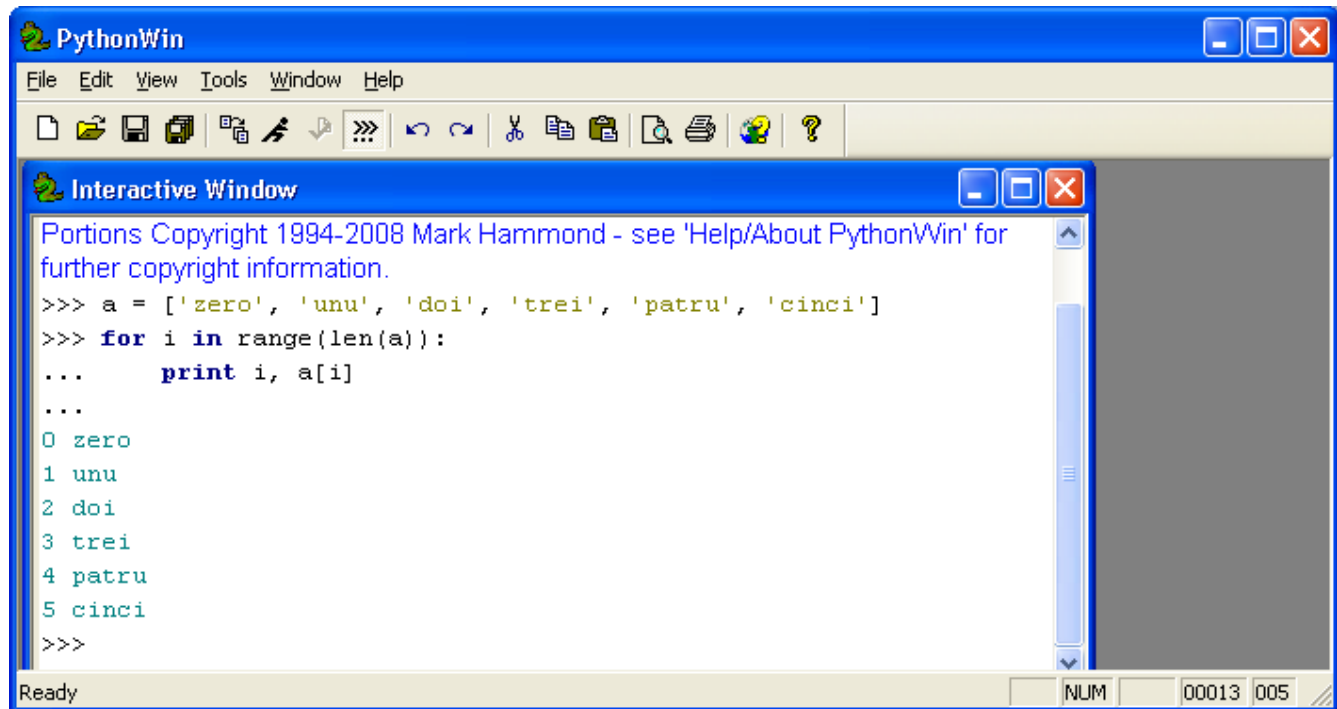
### Exemplu:



```
PythonWin 2.6.6 (r266:84292, Aug 24 2010, 16:01:11) [MSC v.1500 32 bit (Intel)] on win32.
Portions Copyright 1994-2008 Mark Hammond - see 'Help/About PythonWin' for further copyright information.
>>> range(2,10)
[2, 3, 4, 5, 6, 7, 8, 9]
>>> range(0, 10, 3)
[0, 3, 6, 9]
>>> range(-10, -100, -30)
[-10, -40, -70]
>>> |
```



Pentru a realiza o iterație pe o mulțime de numere folosiți funcțiile `range()` și `len()` astfel:



```
PythonWin
File Edit View Tools Window Help
Interactive Window
Portions Copyright 1994-2008 Mark Hammond - see 'Help/About PythonWin' for
further copyright information.
>>> a = ['zero', 'unu', 'doi', 'trei', 'patru', 'cinci']
>>> for i in range(len(a)):
...     print i, a[i]
...
0 zero
1 unu
2 doi
3 trei
4 patru
5 cinci
>>>
```

### 6.5 Instrucțiunile `break` și `continue`, și clauze `else` pentru bucle

La fel ca în C, instrucțiunea **break** termină forțat orice buclă **while** sau **for**. Instrucțiunea **continue** trece necondiționat la următoarea iterație. Instrucțiunile iterative pot avea și o clauză **else**. Instrucțiunile din cadrul unei astfel de clauze `else` sunt executate atunci când bucla se termină odată cu terminarea listei (`for`) sau atunci când condiția buclei devine falsă (pentru `while`) aceste instrucțiuni nu sunt executate dacă bucla este terminată printr-o instrucțiune `break`.

**Exemplu:** determinarea numerelor prime până la 10

The screenshot shows the PythonWin IDE with a script named 'nr\_prime'. The script contains a nested loop that checks for prime numbers between 2 and 10. For each number, it checks if it is divisible by any smaller number. If it is, it prints a message indicating it is not prime and breaks the inner loop. If it is not, it prints a message indicating it is prime. The output in the Interactive Window shows the results of the script: 2 and 3 are prime, 4 is not prime (divisible by 2), 5 is prime, 6 is not prime (divisible by 2), 7 is prime, 8 is not prime (divisible by 2), and 9 is not prime (divisible by 3).

```
PythonWin
File Edit View Tools Window Help
nr_prime
- for n in range(2, 10):
-     for x in range(2, n):
-         if n % x == 0:
-             print n, 'egal cu', x, '*', n/x
-             break
-         else:
-             # bucla s-a epuizat fara sa se gaseasca un factor
-             print n, 'este un numar prim'
Interactive Window
2 este un numar prim
3 este un numar prim
4 egal cu 2 * 2
5 este un numar prim
6 egal cu 2 * 3
7 este un numar prim
8 egal cu 2 * 4
9 egal cu 3 * 3
Script 'C:\Documents and Settings\Julian\Desktop\nr_prime.py' returned exit code 0
```

## 6.5 Instrucțiunea pass

Instrucțiunea **pass** nu execută nimic. Poate fi folosită atunci când este necesară prezența sintactică a unei instrucțiuni, fără ca aceasta să execute ceva:

The screenshot shows the PythonWin IDE with a script named 'pass'. The script contains a while loop with a pass statement. The comment indicates that the pass statement is used to allow for interruption from the keyboard. The output in the Interactive Window shows an exception raised while running the script.

```
PythonWin
File Edit View Tools Window Help
Interactive Window
pass
- while 1:
-     pass #Asteapta pentru intrerupere de la tastatura
Exception raised while running script pass.py
```

## **Exerciții propuse:**

1. Calculați suma primelor  $n$  numere întregi.
2. Interschimbați valorile a două numere întregi.
3. Verificați dacă un cuvânt este palindrom. Un cuvânt este palindrom dacă scris de la dreapta la stanga, este tot acel cuvânt.
4. Calculați CMMDC a două numere folosind algoritmul lui Euclid.
5. Verificați dacă un număr este prim.
6. Verificați dacă un număr întreg, citit de la tastatură, este perfect. Un număr este perfect dacă suma divizorilor lui este egală cu el.
7. Verificați dacă două numere întregi, citite de la tastatură, sunt prietene. Două numere se numesc prietene, dacă suma divizorilor unuia dintre numere este egală cu celalalt număr.

## **Bibliografie:**

1. Mark Pilgrim, **Dive Into Python**
2. Bruce Eckel, **Thinking in Python Design Patterns and Problem-Solving Techniques**
3. <http://www.python.org/doc/current/tut/tut.html>