

Variables. Entrada y Salida

Aspectos generales de la práctica:

- Arrancar el ordenador con el sistema Linux y crear una carpeta en el escritorio denominada *practicas_python*.
- Guardar todos los ejercicios y programas realizados durante la práctica en esa carpeta.
- Puedes hacer comentarios o responder los ejercicios en el editor de texto de PYTHON . Todo comentario debe ir precedido del símbolo #.
- Al finalizar la práctica sube todos los archivos generados a moodle.

■ Variables en PYTHON

En algunas situaciones (no siempre) las variables en PYTHON se pueden entender como *cajas* en las que se guardan los objetos. Las variables se crean cuando se definen por primera vez, es decir, cuando se les asigna un valor por primera vez. Para asignar un valor a una variable se utiliza el operador asignación =. A la izquierda del operador se escribe el nombre/identificador de la variable y a la derecha el valor que se quiere dar a la variable.

De este modo, el operador asignación = de PYTHON se corresponde con la operación que suele denotarse en matemáticas con el símbolo := y que en los diagramas de flujo hemos denotado con una flecha orientada hacia la izquierda (\leftarrow), ilustrando el paso del valor descrito a la derecha al alias escrito a la izquierda.

Es muy importante distinguir entre una asignación y una ecuación. La primera es una instrucción, una acción: cierta variable «toma» un valor. Una ecuación matemática, en cambio, no «cambia» nada: solo indica que dos cantidades son iguales.

Antes de utilizar un nombre o identificador, hay que darle algún valor. Esta es otra diferencia entre las ecuaciones, que son simétricas, y las asignaciones, en las que es importante distinguir dónde se coloca cada término.

```
>>> a = 2
>>> a
2
>>> 2 = c
File "<stdin>", line 1
SyntaxError: can't assign to literal
```

Una variable puede almacenar objetos numéricos, de texto y estructuras más complicadas (que se verán más adelante). En particular, el objeto cadena (**str**) que es una sucesión de caracteres encerrada entre comillas (simples o dobles).

```
>>> type("Hola, mundo !")
<class 'str'>
>>> variable
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'variable' is not defined
>>> variable='Alto Campoo'
>>> variable
'Alto Campoo'
```

El valor de un variable puede variar. A lo largo de un mismo programa, una variable puede incluso asociarse a objetos de tipos distintos.

```

>>> x = "cadena"
>>> type(x)
>>> x = 2**(1 / 2); type(x)
>>> x = x * 3
>>> horas = 5
>>> minutos = 60 * horas
>>> segundos = 60 * minutos
>>> segundos
18000

```

No podemos utilizar cualquier cadena de caracteres como **identificador** para denominar una variable. Por ejemplo, parece razonable que Python no permita llamar a una variable **g=h**, ya que si quisiéramos asignarle un valor: **g=h=2**, el intérprete no podría decidir entre crear dos variables (**g** y **h**) con el valor 2 o una sola.

En general, podemos utilizar letras, números y el signo **_**, con las dos restricciones siguientes:

- El primer carácter del identificador no puede ser un número.
- No pueden utilizarse algunas palabras que reserva el intérprete (por ejemplo: **for**, **break**, **and**, **None**, **if**, **else**,...)

```

>>> 3d = 34
File "<stdin>", line 1
  3d = 34
  ^
SyntaxError: invalid syntax
>>> d3 = 2015
>>> D3 = 2016
>>> d3, D3
2015, 2016

```

Existe otro tipo de sentencias de asignación:

```

>>> x = 2015
>>> x = x + 1
>>> x
2016
>>> x = 2014
>>> x = x % 7
>>> x
5

```

Los lenguajes de programación suelen disponer de símbolos para representar operaciones aritméticas que no son comunes al escribir matemáticas. En PYTHON la expresión:

$$a(\text{operador}) = b$$

es, en general, equivalente a

$$a = a(\text{operador})b$$

```

>>> a = 3; a += 4; a
7
>>> a %= 5; a
2

```

■ Salida por pantalla en PYTHON

La función `PRINT()` permite mostrar texto en pantalla, evalúa las expresiones que involucra según las reglas del intérprete y las muestra por la salida estandar.

```

>>> print("Hola Santander")
Hola Santander
>>> print('Hola Santander')
Hola Santander

```

La función PRINT() admite varios argumentos seguidos. En el programa, los argumentos deben separarse por comas. Los argumentos se muestran en el mismo orden y en la misma línea, separados por espacios:

```
>>> print("Hola", "adios", 2015)
Hola adios 2015
```

Permite incluir variables o expresiones como argumento.

```
>>> nombre = "Alan Turing"
>>> edad = 19
>>> print("Me llamo", nombre, "y tengo", edad, "años.")
Me llamo Alan Turing y tengo 19 años.
```

■ Entrada por teclado en PYTHON

La instrucción INPUT puede considerarse contraria a PRINT: permite la comunicación entre programa y usuario en el sentido inverso. De forma predeterminada, la función convierte la entrada en una cadena (str).

Abrimos el editor de text de IDLE:

```
print("¿Como se llama ?")
nombre = input()
print("Un placer saludarle", nombre)
```

Si queremos que PYTHON interprete la entrada como un número entero o float se debe utilizar la función INT() o la función FLOAT(), para convertirlo (casting).

```
cantidad = int(input("Dígame una cantidad en pesetas: "))
print(cantidad, "pesetas son", cantidad / 166.386, "euros")
```

■ Ejercicios

1. Diseña un programa que pida el radio de una esfera y muestre por pantalla el valor del volumen.
2. Evalúa el polinomio $x^4 + x^3 + 2x^2 - x$ en $x = 1.1$. Utiliza variables para evitar teclear varias veces el valor de x .
3. Diseña un programa que pida tres números al usuario y muestre por pantalla su media aritmética.
4. Para determinar el valor de la dureza Brinell de un material, penetrando una bola de acero templado se emplea la siguiente fórmula:

$$HB = \frac{2F}{\pi D^2} \left(\frac{1}{1 - \sqrt{1 - \frac{d^2}{D^2}}} \right)$$

donde:

- F carga a utilizar medida en [[[kilopondio—kp]]].
- D diámetro de la bola medida en [mm].
- d diámetro de la huella en superficie en [mm].

Diseña un programa que introduciendo el usuario por el teclado F , D y d muestre en pantalla el valor HB .

5. Implementa un programa que pida los coeficientes a, b, c de una ecuación de segundo grado $ax^2 + bx + c$ y devuelva las raíces de la ecuación

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{y} \quad \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

6. Realiza un programa que pida el nombre de una persona y lo muestre en pantalla repetido, pero dejando dos espacios de separación.
7. ¿Qué resultará de ejecutar las siguientes sentencias ?

```
>>> z = 2
>>> z += 2
>>> z += 2 - 2
>>> z *= 2
>>> z *= 1+1
>>> z /= 2
```

```
>>> z %= 3
>>> z /= 3 -1
>>> z -= 2 + 1
>>> z -= 2
>>> z **= 3
>>> z
```

- Implementa un programa que pida una cantidad de euros, una tasa de interés y un número de años. Muestra por pantalla en cuanto se ha incrementado el capital inicial transcurridos esos años con ese interés. Recuerda que un capital C en euros a un interés del x por cien durante n años se convierten en $C \times (1 + x/100)^n$ euros. Comprueba, si la cantidad de 1000 euros al 5,5% de interés anual se convierte en 2232.47 euros transcurridos 15 años.
- Realiza un programa que calcule el desglose del menor número de billetes y monedas de una cantidad entera de euros. Hay billetes de 500, 200,100,50,20,10 y 5 euros y, monedas de 2 y 1 euro. Por ejemplo:

```
Dame una cantidad entera de euros: 903
El desglose es:
1 billete(s)/moneda(s) de 500 euros
2 billete(s)/moneda(s) de 200 euros
0 billete(s)/moneda(s) de 100 euros
0 billete(s)/moneda(s) de 50 euros
0 billete(s)/moneda(s) de 20 euros
0 billete(s)/moneda(s) de 10 euros
0 billete(s)/moneda(s) de 5 euros
1 billete(s)/moneda(s) de 2 euros
1 billete(s)/moneda(s) de 1 euros
```

- Evalúa el polinomio $x^4 + x^3 + x^2 - x$ en $x = 10$. Asegúrate que el resultado sea un número flotante.
- Diseña un programa que muestre por pantalla:

```
+++++
++++
+++
++
+
```