

P02 Como leer y escribir en Python

Todos los programas en algún momento tratan con datos. Una de las primeras cosas que tendremos que aprender será leer y escribir estos datos.

La función `print()` realiza la función de salida. Para usarla pasaremos uno o varios argumentos separados por una coma.

Veamos un primer ejemplo

```
print(5 + 10)
print(3 * 7, (17 - 2) * 8)
print(2 ** 16) # exponenciacion (2 elevado a 16)
print(37 / 3) # division
print(37 // 3) # division entera
print(37 % 3) # resto de la division entera
```

Para leer datos en un programa usamos la función `input()` que lee una cadena de texto como una variable de tipo "String"

El siguiente programa pregunta el nombre de alguien y le saluda

```
print('Como te llamas?')
name = input() # lee lo escrito y lo almacena en la variable
print('Hola ' + name + '!')
```

2. Sumas de números y de cadenas

Vamos a escribir un programa que lee dos números e imprime su suma. Leemos los dos números y los guardamos en las variables `a` y `b` usando el operador de asignación `=`. A la izquierda del operador ponemos el nombre de la variable. El nombre puede ser una cadena de caracteres (A-Z, a-z, 0-9, _) que debe empezar por una letra. En la derecha podemos poner cualquier expresión que python pueda interpretar. El nombre se asocia al resultado.

```
a = input()
b = input()
s = a + b
print(s)
```

Al ejecutar el programa obtenemos como resultado **57**. En matemáticas nos enseñaron que, **5 + 7** tiene por resultado **12**. El programa está equivocado, y es muy importante entender por qué. El programa en la tercera línea **s = a + b** ha "sumado" dos cadenas, en vez de dos numeros. En Python la suma de dos cadenas funciona como sigue: las dos cadenas se escriben una a continuación de la otra, se concatenan

En Python todos los valores se llaman “objetos”. Cada objeto es de un cierto tipo. El numero 2 corresponde a un objeto de tipo int (números enteros) mientras que “hola” corresponde a un objeto “str” o cadena. Los números decimales se representan en objetos de tipo “float”. El tipo de los objetos determina que operaciones se pueden hacer con ellos.

Por ejemplo, si dos variables contienen objetos de tipo int es posible multiplicarlos, pero si son de tipo str es imposible, como se puede ver en el ejemplo

```
first = 5
second = 7
print(first * second)
# se usan comillas para determinar una cadena str
first = '5'
second = "7"
print(first * second)
```

Para convertir una cadena str de dígitos en un numero entero int podemos emplear la function `int()`. Por ejemplo, `int('23')` da como resultado un objeto int con valor 23.

Con todo esto podemos arreglar el programa anterior para que realice bien la suma.

```
a = int(input())
b = int(input())
print(s)
```

Ejercicios:

1.- Vuestro primer programa:

Realizar un programa que haga aparecer “hola mundo” en la pantalla

2.- Suma de tres números

Escribir un programa que lea tres números y escriba su suma. Cada numero se lee en una línea distinta.

3.- Área de un triángulo rectángulo

Escribir un programa que lea la base y la altura de un triángulo rectángulo y escriba su área. Cada número se lee en una línea distinta

4.- Compartir manzanas.

N estudiantes compran K manzanas y las reparten entre ellos, llevándose todos el mismo número. Las que sobran se quedan en la bolsa. ¿Cuántas manzanas se queda cada uno?. ¿Cuántas quedan en la bolsa?

El programa debe leer los números N y K y responder las dos respuestas a estas preguntas.

5.- Reloj digital

Dado el entero N- El numero de minutos que han pasado desde media noche, ¿cuántas horas y minutos aparecen en la pantalla de un reloj digital?

El programa debe leer el Número N y debe devolver el número de horas y el número de minutos.

Como ejercicio adicional se puede intentar conseguir que imprima los dos números en la misma línea, o todavía mejor, con el formato de un reloj digital, separando los dos números con un carácter “:”

Ejercicios con un poco más de teoría:

1.- Hola, Paco!

El programa debe preguntar como te llamas, leer el nombre y responder con la palabra hola, una coma, el nombre del usuario y un signo de exclamación después¹.

T1:

Para unir dos cadenas se puede emplear el operador “+”

2.- Anterior y siguiente.

Escribe un programa que lea un número entero y escriba sus números anterior y siguiente.

T2

Puedes convertir números en cadenas usando la función `str()`.

3.- Pupitres:

El instituto decide comprar nuevos pupitres para tres aulas. Cada pupitre permite sentarse a dos estudiantes. Dado el numero de estudiantes en cada clase devuelve el numero mínimo de pupitres que deberían comprarse.

Ejemplo:

Si el primer grupo tiene 20 estudiantes, el segundo 21 y el tercero 23 se necesitaran 10 pupitres para el primero, 11 para el segundo y 12 para el tercero, y el programa debería responder “33”.

¹ En castellano los símbolos de exclamación e interrogación aparecen al principio y al fin de la oración, desgraciadamente la mayor parte de lenguajes de programación están escritos por angloparlantes, que no las emplean, y pese a que no es imposible emplearlas, pueden dar lugar a algunos problemas que en este curso preferiremos simplemente ignorar.