



5. FUNCIONES

Conversión de tipos

Python proporciona un conjunto de funciones internas que convierten valores de un tipo a otro. P ej:

- int** intenta transformar cualquier dato al tipo de dato entero.
- str** intenta transformar cualquier dato en una cadena de texto.

SIN STR

```
a = int(input("Introduce operando a:"))
b = int(input("Introduce operando b:"))
res = a + b
print("El resultado es: " + res)
```

```
Traceback (most recent call last):
  File "C:/Users/franp/AppData/Local/Programs/
  4, in <module>
    print("El resultado es: " + res)
TypeError: can only concatenate str (not "int")
```



CON STR

```
a = int(input("Introduce operando a:"))
b = int(input("Introduce operando b:"))
res = a + b
print("El resultado es: " + str(res))
```

```
Introduce operando a:4
Introduce operando b:3
El resultado es: 7
```



5. FUNCIONES

Funciones matemáticas

Python tiene un módulo matemático (math) que proporciona la mayoría de las funciones matemáticas habituales. Para utilizarlo, lo tenemos que importar con “import math” y después podemos hacer uso de diversas funciones como, por ejemplo, “sqrt” que calcula la raíz cuadrada.

```
import math

a = int(input("Introduce un valor: "))
resul = math.sqrt(a)
print(resul)
```

Números aleatorios

Conseguimos números aleatorios importando el módulo random con “import random” y haciendo uso de la función “randint” del siguiente modo:

```
import random

#devuelve un numero aleatorio entre 5 y 10
a = random.randint(5,10)
print(a)
```



5. FUNCIONES



Actividad 8

Haz un programa en Python que pida los coeficientes a, b y c de una ecuación de segundo grado e imprima por pantalla sus 2 soluciones posibles x1 y x2.

$$ax^2 + bx + c = 0$$

Recuerda que las soluciones a una ecuación de segundo grado vienen dadas por la formular siguiente:



$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

```
Introduzca coeficiente a:2
Introduzca coeficiente b:5
Introduzca coeficiente c:3
Las 2 soluciones son:
x1 =  -1.0   x2=  -1.5
```



5. FUNCIONES



Actividad 9

Haz un programa que muestre al azar una carta de la baraja americana. Una carta está compuesta de un número y de un palo:

- Los números pueden ser: AS, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K.
- Los palos pueden ser: *Picas*, *Corazones*, *Rombos* y *Tréboles*.

Cada vez que el usuario pulse ENTER, el programa le devolverá una carta aleatoria y, cuando pulse “salir”, el programa finalizará.

```
Pulse ENTER para pedir carta o 'salir' para finalizar:
3 de Rombos
Pulse ENTER para pedir carta o 'salir' para finalizar:
3 de Picas
Pulse ENTER para pedir carta o 'salir' para finalizar:
2 de Treboles
Pulse ENTER para pedir carta o 'salir' para finalizar:
3 de Corazones
Pulse ENTER para pedir carta o 'salir' para finalizar:
As de Rombos
Pulse ENTER para pedir carta o 'salir' para finalizar: salir
>>> |
```



5. FUNCIONES

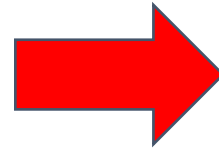
Definición de funciones propias

Las funciones se definen con la palabra clave “def” junto con el nombre de la función:

```
def buenos_dias():  
    print("Hola buenos días")
```

```
def buenas_noches():  
    print("Buenas noches")
```

```
buenos_dias()  
buenas_noches()
```



```
Hola buenos días  
Buenas noches
```

Es necesario crear una función antes de llamarla, si lo hacemos al revés, Python no la encontrará y el programa generará una excepción.

En este caso, las funciones no se les proporcionaba ningún dato de entrada (o parámetros) ni tampoco devolvían ningún dato, así que vamos a complicar un poco la cosa...

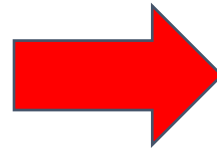


5. FUNCIONES

Parámetros de entrada y de salida de una función

En la cabecera podemos pasar el nombre de un argumento que luego utilizaremos en el cuerpo de la función. Por ejemplo:

```
def muestra_dos_veces(paco):  
    print(paco)  
    print(paco)  
  
muestra_dos_veces(13)  
muestra_dos_veces("Helicoptero")
```



```
13  
13  
Helicoptero  
Helicoptero  
>>>
```

Por otro lado, una función también puede devolver un valor con la palabra “**return**”:

```
def suma(a,b):  
    return a+b  
  
res = suma(1,2)  
print(res) #Imprime un 3
```

```
def suma(a,b):  
    return a+b  
  
res = suma("Buenos ", "dias")  
print(res) #Imprime Buenos dias
```



5. FUNCIONES

Paso por valor y paso por referencia

En un paso **por valor**, se realiza una copia de la variable, de forma que no se modifica nunca la original del programa principal, a menos que se utilice un **return**. Es el paso que se da con los tipos de datos simples.

```
def incrementa(a):  
    a = a + 1  
  
a = 4  
incrementa(a)  
print(a) #Imprime 4
```

```
def incrementa(a):  
    a = a + 1  
    return a ←  
  
a = 4  
a = incrementa(a) ←  
print(a) #Imprime 5
```

En un paso **por referencia**, sin embargo, sí que se modifica la estructura facilitada sin necesidad de hacer un return. Es el paso que se usa en los tipos compuestos (como listas):

```
def incrementa(a):  
    a[0] = a[0] + 1  
  
a = [1,2,3]  
incrementa(a)  
print(a) #Imprime [2,2,3]
```



5. FUNCIONES

Variable global dentro de una función

Si queremos asignar valor a una variable del main en una subrutina, Python nos ofrece la posibilidad de llevar a cabo esta asignación mediante el uso de la palabra “**global**”. De esta forma, la función busca en el programa principal el valor de esa variable.

```
def incrementa():  
    global a  
    print("Dentro, antes del incremento: ",a)  
    a = a + 1  
    print("Dentro, despues del incremento: ",a)  
  
a = 5  
incrementa() #llamada a la funcion sin parametros  
print("Fuera, al final del programa: ",a)
```

```
Dentro, antes del incremento:  5  
Dentro, despues del incremento:  6  
Fuera, al final del programa:  6  
>>>
```




5. FUNCIONES



Actividad 10

Haz un programa que simule 2 tiradas de dado de un jugador1, 2 tiradas de dado de un jugador 2 y que luego compare la suma de tiradas de los 2 jugadores. Si la suma de ambos jugadores coincide, se considerará empate y se volverá a tirar.

Para ello, se hará uso de la función “dados()”, que calcula un número aleatorio entre 1 y 6 y lo devuelve al programa principal. Un ejemplo de ejecución sería el siguiente:

```
Tirada 1 del jugador 1: 2
Tirada 2 del jugador 1: 5
Tirada 1 del jugador 2: 2
Tirada 2 del jugador 2: 5
Empate a 7 puntos.
Tirada 1 del jugador 1: 6
Tirada 2 del jugador 1: 6
Tirada 1 del jugador 2: 6
Tirada 2 del jugador 2: 1
Ha ganado el jugador 1 : 12 > 7
```



5. FUNCIONES



Actividad 11

Escribe las siguientes 3 funciones en Python:

-*pares(a,b)*: imprime los valores pares entre a y b.

-*impares(a,b)*: imprime los valores impares entre a y b.

-*cuadrados(a,b)*: imprime elevado al cuadrado todos los valores entre a y b.

Una vez codificadas, realiza un programa que pida al usuario 2 números: un límite inferior y uno superior. Deberá validar que el inferior es estrictamente más pequeño que el superior, solicitándolos en bucle hasta que esto se cumpla. Una vez validados, ofrecerá el resultado de las 3 funciones:

```
Introduce un limite inferior: 2
Introduce un limite superior: 1
Introduce un limite inferior: 2
Introduce un limite superior: 5
Valores pares entre 2 y 5 :
2
4
Valores impares entre 2 y 5 :
3
5
Cuadrados entre 2 y 5 :
El cuadrado de 2 es: 4
El cuadrado de 3 es: 9
El cuadrado de 4 es: 16
El cuadrado de 5 es: 25
>>>
```