



2. PRIMEROS PASOS

Definición de variables

Las variables que declaramos en Python utilizan tipado dinámico, es decir, el tipo se calcula automáticamente en cuanto le asignamos un valor

```
saludo = "Holi"      #variable cadena de texto
codigo_postal = 46025 #variable tipo entero
altura = 1.90        #variable tipo float
```

Y, si en algún momento tenemos una duda del tipo que estamos almacenando, siempre se lo podemos consultar al idle con el comando **type** facilitándole como parámetro entre paréntesis el nombre de la variable para saber su tipo de datos:

```
>>> type(saludo)
<class 'str'>
>>> type(codigo_postal)
<class 'int'>
>>> type(altura)
<class 'float'>
```



2. PRIMEROS PASOS

Comentarios y Operadores aritméticos

Fíjate como los comentarios van siempre precedidos del **carácter almohadilla (#)** y describirán información del código para facilitar su legibilidad y mantenimiento.

Por su parte, los **operadores aritméticos** que utilizaremos son:

Operador	Descripción	Ejemplo
+	Suma	<code>r = 3 + 2 # r es 5</code>
-	Resta	<code>r = 4 - 7 # r es -3</code>
-	Negación	<code>r = -7 # r es -7</code>
*	Multiplicación	<code>r = 2 * 6 # r es 12</code>
**	Exponente	<code>r = 2 ** 6 # r es 64</code>
/	División	<code>r = 3.5 / 2 # r es 1.75</code>
//	División entera	<code>r = 3.5 // 2 # r es 1.0</code>
%	Módulo	<code>r = 7 % 2 # r es 1</code>



2. PRIMEROS PASOS

Lectura de información

Utilizaremos el comando **input** para leer datos por teclado. Los datos se guardarán **siempre en una variable de tipo cadena de texto**.

Por ejemplo:

```
nombre = input("Introduce tu nombre: ")
apellido1 = input("Introduce tu primer apellido: \n")
apellido2 = input("Introduce tu segundo apellido: \n")
nombre_completo = nombre + " " + apellido1 + " " + apellido2
print("Tu nombre completo es: " + nombre_completo)
```

Que produciría por pantalla el siguiente resultado:

```
Introduce tu nombre: Paco
Introduce tu primer apellido:
Perez
Introduce tu segundo apellido:
Vidal
Tu nombre completo es: Paco Perez Vidal
```



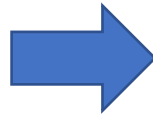
2. PRIMEROS PASOS

Escritura de información

Utilizaremos el comando **print** para imprimir datos por pantalla. Cada impresión se hará en una nueva línea, ya que la instrucción lleva implícito al final el carácter de “\n” (nueva línea).

Por ejemplo:

```
print("Hola ")  
print("Mundo")
```

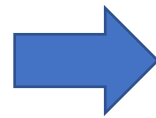


```
Hola  
Mundo
```

Sin embargo, si lo que queremos es imprimir en la misma línea, lo que haremos será especificar que queremos que ese carácter de final de línea desaparezca, empleando un espacio en su lugar (end="").

Por ejemplo:

```
print("Hola ", end="")  
print("Mundo")
```



```
Hola Mundo
```



2. PRIMEROS PASOS



Actividad 1

Crea un programa en Python que le solicite al usuario los siguientes datos: nombre completo, edad, altura, código postal, calle, puerta, numero y asignatura favorita. Luego ha de imprimir una ficha con toda esa información del siguiente modo:

```
Introduce tu nombre completo: Paco Perez Vidal
Introduce tu edad: 41
Introduce tu altura: 1.80
Introduce tu codigo postal: 46025
Introduce tu calle: Periodista
Introduce tu numero: 19
Introduce tu puerta: 20
Introduce tu asignatura favorita: Bases de datos
-----
- Paco Perez Vidal, 41 años, 1.80m.
- Domicilio:
    Periodista, num.19, pta.20 CP:46025
- Asignatura favorita: Bases de datos
-----
```



2. PRIMEROS PASOS

Lectura de información – Uso de la función int()

Como todo lo que leemos en Python se transforma en una cadena de texto, tendremos que hacer uso de la **función int()** para transformarlo en un entero y poder realizar operaciones aritméticas. En este punto, tenemos 2 opciones:

#Opción 1: Pasamos el dato a una variable de tipo entero

```
edad = input("Introduce un numero:") #Lee 4
```

```
edad_int = int(edad)
```

```
edad_int = edad_int + 1
```

```
print(edad_int) #Imprime 5
```

#Opcion 2: Aplicamos la funcion int a input

```
edad = int(input("Introduce un numero:")) #Lee 4
```

```
edad = edad + 1
```

```
print(edad) #Imprime 5
```



2. PRIMEROS PASOS



Actividad 2

Realiza un programa en Python que lea 2 operandos de tipo entero **a** y **b** y efectúe las siguientes operaciones sobre ellos: suma, resta, división, división entera, módulo, producto y potencia. Un ejemplo de ejecución sería el siguiente:

```
Introduce operando a: 5
Introduce operando b: 2
SUMA a + b: 7
RESTA a - b: 3
DIV a / b: 2.5
DIV ENTERA a // b: 2
MODULO a % b: 1
PRODUCTO: 10
POTENCIA: 25
```



2. PRIMEROS PASOS

Captura de excepciones

¿Qué ocurre cuando se solicita al usuario un número e introduce una letra?. Efectivamente, el programa falla y nos devuelve un mensaje de error. Decimos entonces que el programa “ha producido una excepción”.

Para “capturar esa excepción”, ubicaremos las instrucciones susceptibles de generar excepciones dentro de un bloque “**try**”. Cuando se produzca un fallo en estas instrucciones, se ejecutarán las instrucciones del bloque “**except**”

```
try:
    edad = int(input("Introduzca su edad:"))
except:
    print("Ojo, eso no es un numero")
```

```
Introduzca su edad:paco
Ojo, eso no es un numero
```




2. PRIMEROS PASOS

Captura de excepciones

También es posible que queramos generar nosotros nuestras propias excepciones cuando detectemos que el dato introducido no es el esperado.

Por ejemplo, imaginemos que queremos lanzar una excepción cuando la edad introducida no es mayor que 18 años.

```
print("Proceso de apertura de una cuenta bancaria")
try:
    edad = int(input("Introduce tu edad "))
    if (edad < 18):
        raise Exception
except:
    print("ERROR: No eres mayor de edad")
```