

# Fundamentos de Programación.

## Primer Curso de ASIR.

### **UT 02.02 – Funciones y métodos para números.**

# UT 02.02 – Funciones y métodos para números.

## 1.- Introducción.



- Python 3 soporta tres tipos de datos numéricos:
  - Enteros.
  - Decimales (coma flotante).
  - Complejos.
- Por otro lado, un número no solo puede ser representado en base decimal sino que puede ser representado en otras bases.  
Las bases de representación mas importantes son: **decimal, binario, octal y hexadecimal.**

# UT 02.02 – Funciones y métodos para números.

## 2.- Operaciones aritméticas básicas.

Operador	Resultado
+	Suma.
-	Resta.
*	Producto.
/	División.
//	Div. entera (truncado).
%	Resto de la división entera.
**	Potencia.

- Python 3 soporta las operaciones aritméticas.
- Es importante tener en cuenta el tipo de dato de las variables involucradas ya que pueden dar resultados inesperados.
- Por ejemplo, al realizar operaciones con decimales, los resultados pueden presentar errores de redondeo.

```
>>> 100 / 3
33.333333333333336
```

- Este error es debido a que Python almacena los números decimales en binario y se provocan **errores de redondeo en los procesos de conversión**. Es un error que sufren casi todos los lenguajes de programación.

Operación	Resultado
8/3	2.6666666666666665
8//3	2
14%3	2
2**3	8

# UT 02.02 – Funciones y métodos para números.

## 2.- Operaciones aritméticas básicas.

Función	Resultado
int()	Convierte a entero.
float()	Convierte a decimal.
str()	Convierte a cadena.

- La operación de cast o casting consiste en convertir un tipo de dato a otro.
- En programación se realiza dos tipos de casting:
  - **Implícita.** Se realiza de forma automáticamente por Python. Se produce cuando se realizan operaciones con variables de distintos tipos.
  - **Explícita.** Se realiza por el programador para asegurar que una variable es de un cierto tipo.

Operación	Resultado
int(5.2)	5
str(3.25)	'3.25'
print(int("Cadena"))	ValueError: invalid literal for int() with base 10: 'Cadena'

# UT 02.02 – Funciones y métodos para números.

## 3.- Operaciones de redondeo.

Función	Resultado
round()	Redondea un número.
trunc()	Trunca un número decimal.
floor()	Redondea al número entero anterior.
ceil()	Redondea al número entero posterior.

Función	Resultado
round(129.148, 2)	129.15
round(129.142, 2)	129.14
round(129.142, -1)	129
round(129.142, -2)	130
trunc(43.12)	43
floor(43.12)	43
ceil(43.12)	44

- Existen dos operaciones para redondear número.
  - **round()**. Redondea un número. Es decir, para redondear a una determinada cifra se mira la siguiente cifra. Si el número es menor que cinco se redondea por defecto; es caso contrario por exceso.
  - **trunc()**. Trunca un número. Es decir, simplemente se elimina la parte decimal del número.
- Ambas funciones toman dos parámetros: el número y la precisión.
  - Si la precisión es positiva se redondea a la cifra decimal.
  - Si la precisión es negativa se redondea a la cifra entera.

# UT 02.02 – Funciones y métodos para números.

## 4.- Conversión entre bases (I).

### Conversión de decimal a otras bases.

Función	Resultado
bin(entero)	Convierte el número entero a binario.
oct(entero)	Convierte el número entero a octal.
hex(entero)	Convierte el número entero a hexadecimal.

### Conversión a base decimal.

Función	Resultado
int(número, base)	Convierte a decimal el número en la base indicada.

### Ejemplos.

Función	Resultado
bin(12)	'0b1100'
oct(12)	'0o14'
hex(12)	'0xc'
int('0xfc2',16) + int('0o3712', 8)	6028

- Para expresar un número en otra base que no sea la base decimal lo haremos de la siguiente forma:
  - **Binario.** '0b[número]'.
  - **Octal.** '0o[número]'.
  - **Hexadecimal.** '0x[número]'.
- Python 3 tiene tres funciones que permiten la conversión a binario, octal y hexadecimal. **El resultado es una cadena de texto.**
- Para realizar la operación contraria usaremos la función int(número, base). La base se expresa mediante su número: 2 (binario), 8 (octal) y 16 (hexadecimal).

# UT 02.02 – Funciones y métodos para números.

## 4.- Conversión entre bases (II). Ejemplo.

```
# Solicitamos un numero y lo mostramos en binario.
numero = int(input("Introduce un número: "))
binario = bin(numero)
print(f"El numero {numero} en binario es {binario}.")

# Formateamos binario a 8 digitos con ceros a izquierda.
# Para ello, hay que eliminar "0b" de la cadena.
cadena = str(binario)
cadena = f"{cadena[2:]:0>8}"
print(f"El resultado del formateado es {cadena}.")

# Por último convertimos el numero binario contenido
# en entero para comprobar que se ha realizado la
# conversion.
numero2 = int(cadena,2)
print(f"numero2 = {numero2}")
```

```
Introduce un número: 45
El numero 45 en binario es 0b101101.
El resultado del formateado es 00101101.
numero2 = 45
```

# UT 02.02 – Funciones y métodos para números.

## 5.- Otras funciones integradas en el lenguaje.

- Python 3 tiene una serie de funciones integradas en el lenguaje. Además de estas, se implementan multitud de funciones mediante paquetes matemáticos.
- Entre las funciones mas representativas están las representadas en la tabla anexa.

Función	Resultado
abs(x)	Devuelve el valor absoluto de x.
max(iterable)	Devuelve el elemento mayor contenido en el iterable. El iterable puede contener conjunto de números, pero también puede contener cadenas de caracteres.
min(iterable)	Devuelve el elemento menor contenido en el iterable. El iterable puede contener conjunto de números, pero también puede contener cadenas de caracteres.
sum(iterable)	Suma los elementos contenidos en el iterable.
d,r = divmod(x, y)	Devuelve el cociente y el resto al dividir el número x por el número y en las variables d y r respectivamente.
sorted(iterable)	Devuelve otro iterable con los elementos del iterable pasado como parámetro ordenados.



# UT 02.02 – Funciones y métodos para números.

## 6.- El paquete math.

- Para mas funciones matemáticas hay una serie de paquetes que contienen multitud de ellas. Entre los paquetes mas importantes está el paquete **math**.
- Se puede ver una relación de funciones contenida en la documentación de Python en la dirección:  
**<https://docs.python.org/es/3/library/math.html>**
- La forma de usar las funciones del paquete math es importandolo en el comienzo del script mediante la sentencia.

**import math**

```
import math

numero = float(input("Introduce un numero: "))
numero = math.trunc(numero)
print(f"El resultado de truncar numero es {numero}.");
```

```
Introduce un numero: 315.275

El resultado de truncar numero es 315.
```