



MÓDULO 2: TIPOS DE DATOS Y OPERADORES

Tipos de datos

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

- Tipos de datos
 - Numéricos (admiten guiones bajos como separador)
 - Enteros
 - Base decimal (198)
 - Base octal (0o341)
 - Base hexadecimal (0x3A1)
 - Base binaria (int('101',2))
 - Flotantes (coma o punto flotante)
 - 198.34
 - .34
 - 198.
 - 5e3
 - 5E3
 - 5e-5

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

- Tipos de datos

- Cadenas de caracteres (string)

- Un string es una secuencia de caracteres: en una secuencia el orden importa.
 - Son inmutables.
 - “Este texto es una cadena de caracteres”
 - Operador concatenación: +
 - Comillas dentro de las comillas
 - Cambiar comillas
 - Escapar comillas \”
 - cadena[3]
 - Slicing : cadena[?:?]
 - Función len
 - Comparación de cadenas <, >, =

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

- Tipos de datos
 - Cadenas de caracteres (string)
 - Método capitalize()
 - Método count() → Cuantas veces aparece una subcadena ó **0 si no hay ninguna ocurrencia.**
 - Método index() → Proporciona la posición de una cadena o **error si no hay ninguna ocurrencia.**
 - Operador in
 - Método startswith() y endswith()
 - Método find() → Indica dónde está la primera ocurrencia de un texto. Devuelve **-1 si no hay ninguna ocurrencia.**
 - Métodos isdigit() (True si todos los caracteres son dígitos), isnumeric()
 - Método islower() → Determina si todos los caracteres están en minúscula.
 - Método isspace() → Determina si todos los caracteres son espacios.
 - Método isupper() → Determina si todos los caracteres están en mayúscula.

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

- Tipos de datos

- Cadenas de caracteres (string)

- Método `lower()` → Convierte todos los caracteres a minúscula
 - Método `replace()` → Sustituye una subcadena por otra
 - Método `split()` → Devuelve una lista con las partes.
 - Método `splitlines()` → Devuelve una lista con las líneas del texto.
 - Método `upper()` → Convierte todos los caracteres a mayúscula
 - Método `join()` → Crea un string a partir de una lista de strings, separando cada elemento de la cadena por un símbolo.

```
>>> nombres = ("Vicky", "Cristina", "Barcelona")  
>>> "---".join(nombres)  
'Vicky---Cristina---Barcelona'
```

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

- Tipos de datos
 - Booleanos
 - True (1)
 - False (0)
 - Comparar booleanos

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

Conversiones

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

- Conversiones

- `isinstance` → `isinstance(variable, int)`
- Cambio de tipo de una variable
- Una operación aritmética en la que participa un número flotante devuelve un número flotante.

- Casting

- `int()`, `float()`, `str()`
 - `int('101')`
 - `int('101',2)` → En base 2
- `2 * .5`

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

Operadores

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

- Operadores aritméticos
 - + (Incremento y Suma)
 - - (Decremento y Resta)
 - * (Multiplicación)
 - / (División) (No se admite división entre 0)
 - // (División entera)
 - % (módulo)
 - ** (exponencial)

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

- Operadores aritméticos abreviados

- $+=$

- $x+=1$ es equivalente a $x=x+1$

- $-=$

- $x-=1$ es equivalente a $x=x-1$

- $*=$

- $/=$

- $\% =$

- $**=$

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

- Precedencia de operadores aritméticos(jerarquía de operadores)
 - +,- (Unario)
 - **
 - *,/,%
 - +,-
- De izquierda a derecha (enlazado hacia la izquierda) excepto ** (enlazado hacia la derecha) de derecha a izquierda.
- Uso de paréntesis

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

- Operadores de cadenas de caracteres
 - + (Concatenación)
 - “Hola” + “ ” + “Mundo” → “Hola Mundo”
 - * (Replicación)
 - “Hola” * 3 → “HolaHolaHola”

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

- Operadores booleanos

- == (es igual)
- != (no es igual)
- > (Mayor)
- >= (Mayor o igual)
- < (Menor)
- <= (Menor o igual)

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

- Operadores lógico
 - and
 - or
 - not

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

- Operadores de bit (bitwise)
 - & (conjunction)
 - | (disjunction)
 - ~ (negation)
 - ^ (ó exclusivo - xor)
 - &=, |=, ^=
- Evalúan todos los bits por separado.
- Sólo funcionan con números enteros.

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

- Ejemplo &
 - 15 en base decimal es 1111 en binario
 - 8 en base decimal es 1000 en binario
 - $15 \& 8 \rightarrow 1000$

```
>>> a=15  
>>> b=8  
>>> c = a & b  
>>> c  
8
```

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

- Ejemplo ~.
 - 14 en base decimal es 1110 en binario
 - ~ 14
 - 1110

```
>>> a=14  
>>> ~14  
-15  
>>>
```

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

■ Máscara de bits:

- Dados los siguientes números binarios:

- 000000000000000000000000000000001011 → 11 en decimal
 - 000000000000000000000000000000001111 → 15 en decimal

- Se desea saber cuál es el valor del tercer bit:

- La máscara es 0100 (4 en decimal)
 - 11 & 4 → 0 (por lo tanto es False)
 - 15 & 4 → 4 (por lo tanto es True)

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

■ Reinicio de bit:

- Dado el siguiente número binario:
 - 000000000000000000000000000000001111 → 15 en decimal
- Se desea poner el tercer bit a 0 sin tocar el resto:
 - La máscara es 0100 (4 en decimal)
 - `numero = numero & ~mascara`
 - Resultado: 000000000000000000000000000000001011 → 11 en decimal.

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

■ Establecimiento de bit:

- Datos el siguiente número binario:
 - 000000000000000000000000000000001011 → 11 en decimal
- Se desea poner el tercer bit a 1 sin tocar el resto:
 - La máscara es 0100 (4 en decimal)
 - numero = numero | mascara
 - Resultado: 000000000000000000000000000000001111 → 15 en decimal.

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

■ Negación de bit:

- Datos el siguiente número binario:
 - 000000000000000000000000000000001011 → 11 en decimal
- Se desea invertir el valor del tercer bit:
 - La máscara es 0100 (4 en decimal)
 - numero = numero ^ mascara
 - Resultado: 000000000000000000000000000000001111 → 15 en decimal.
 - numero = numero ^ mascara Aplicado sobre el resultado
 - Resultado: 000000000000000000000000000000001011 → 11 en decimal

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

- Desplazamiento (shifting) izquierdo binario y derecho binario.
- Sólo válido para números enteros.
 - >> dividir entre dos
 - $8 >> 2 \rightarrow 2$ (Desplazar dos veces)
 - $1000(8) \rightarrow 0010(2)$
 - $16 >> 3 \rightarrow 2$ (Desplazar tres veces hacia la derecha)
 - $10000(16) \rightarrow 00010(2)$
 - << multiplicar por dos
 - $3 << 4 \rightarrow 48$

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

■ Precedencia de operadores:

Operator	Description
(expressions...), [expressions...], {key: value...}, {expressions...}	Binding or parenthesized expression, list display, dictionary display, set display
x[index], x[index:index], x(arguments...), x.attribute	Subscription, slicing, call, attribute reference
await x	Await expression
**	Exponentiation [5]
+x, -x, ~x	Positive, negative, bitwise NOT
*, @, /, //, %	Multiplication, matrix multiplication, division, floor division, remainder [6]
+, -	Addition and subtraction
<<, >>	Shifts

&	Bitwise AND
^	Bitwise XOR
	Bitwise OR
in, not in, is, is not, <, <=, >, >=, !=, ==	Comparisons, including membership tests and identity tests
not x	Boolean NOT
and	Boolean AND
or	Boolean OR
if – else	Conditional expression
lambda	Lambda expression
:=	Assignment expression

- <https://docs.python.org/3/reference/expressions.html#operator-precedence>

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

Estructuras de datos - variables

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

- Variables
 - Definición
 - Declaración vs Inicialización vs Asignación
 - Identificador
 - Debe comenzar con una letra o guion bajo
 - Puede contener letras mayúsculas y minúsculas, dígitos y guiones bajos
 - Las letras mayúsculas y minúsculas son distintas
 - No se pueden utilizar las palabras reservadas (*keywords*) como identificadores

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

■ Variables

▪ Convenciones:

- <https://www.python.org/dev/peps/pep-0008/#naming-conventions>
- <https://google.github.io/styleguide/pyguide.html>

3.16.4 Guidelines derived from Guido's Recommendations

Type	Public	Internal
Packages	lower_with_underscores	
Modules	lower_with_underscores	_lower_with_underscores
Classes	CapWords	_CapWords
Exceptions	CapWords	
Functions	lower_with_underscores()	_lower_with_underscores()
Global/Class Constants	CAPS_WITH_UNDER	_CAPS_WITH_UNDER
Global/Class Variables	lower_with_underscores	_lower_with_underscores
Instance Variables	lower_with_underscores	_lower_with_underscores (protected)
Method Names	lower_with_underscores()	_lower_with_underscores() (protected)
Function/Method Parameters	lower_with_underscores	
Local Variables	lower_with_underscores	

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

■ Variables

■ Palabras reservadas (*keywords*)

```
>>> help("keywords")

Here is a list of the Python keywords. Enter any keyword to get more help.

False          break           for             not
None           class           from            or
True            continue        global          pass
__peg_parser__ def             if              raise
and             del             import          return
as              elif            in              try
assert         else            is              while
async          except          lambda          with
await          finally         nonlocal       yield
```

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

- Variables
 - Función dir() → Obtiene todas las variables
 - Función type(variable) → Obtiene el tipo de la variable
 - isinstance → isinstance(variable, int)
 - Función del → Eliminación total o parcial

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

- Variables

- Intercambios de valor y asignaciones múltiples
 - variable1, variable2 = variable2, variable1
 - variablen0, variablen1 = variablen2, variablen3
 - n1,n2,n3=3,8,9

MÓDULO 2: TIPOS DE DATOS Y OPERADORES

- Enlaces:

- Data types

- <https://docs.python.org/3/library/datatypes.html>

- Python data types:

- <https://www.geeksforgeeks.org/python/python-data-types/>

- Python operators:

- <https://www.geeksforgeeks.org/python/python-operators/>

- Operators and Expressions in Python:

- <https://realpython.com/python-operators-expressions/>