Tema 2: Bases de programación

Apuntes

[**Tipos de datos y valores 2**](#_dnkx20po30o2)

[**Variables 2**](#_bnm5o8s80vpb)

[**Constantes 3**](#_15rpt6mtz8if)

[**Palabras reservadas 3**](#_ii7e0qrr3px9)

[**Identificadores 3**](#_p9x12p3ja598)

[**Tipos numéricos 3**](#_11nlbnahvy9h)

[**Tipos de texto 4**](#_62m3frkr9bxp)

[**Tipos booleanos 4**](#_vhr0pbldcdz4)

[**Operadores básicos 4**](#_bxoakegmcxm9)

[Ejemplos 4](#_xjtsgffsl3b1)

[**Expresiones 6**](#_vuedcgquo4li)

[**Asignaciones 7**](#_o7zl1p9493oz)

Un lenguaje de programación necesita: asignar valores, repetir acciones y dar saltos en el código

# Tipos de datos y valores

**Básicos**

* Números
  + int (enteros)
  + float (decimales, la separación es con punto)
  + …
    - Enteros largos
    - Alta precisión
    - Números imaginarios → importantes en electrónica. Python es de los pocos lenguajes con soporte para estos números
* Texto (str/string)
* Booleano: Sus valores son True o False. En python acepta también el valor None. Importante mantener las mayúsculas de True y False.
* Valor especial: None → representa la ausencia de valor, igual que null. Es un buen valor de partida para inicializar una variable

**Complejos**

Son la composición de varios tipos de datos básicos, por ejemplo serían listas (secuencias de valores), los conjuntos, los diccionarios (estructuras de datos asociativas), entre otros.

# Variables

Recuadros/espacios de memoria que sirven para almacenar valores.

Operaciones básicas con una variable: asignarle valor y leer su valor.

El contenido puede variar durante la ejecución de un programa.

Deben tener un nombre, este debe ser significativo y seguir ciertas reglas:

* Hay caracteres que no pueden usarse (ej: !, ñ, #, @, ‘,”)
* Hay palabras reservadas (ej: for)
* Es case sensitive (mayús y min)
* Deben ser únicos, no se pueden repetir

En Python la variable no necesita ser declarada al principio para ser usada. En un lenguaje estático (necesita compilación) sí que hay que declararla, se necesita hacer una reserva de espacio antes de compilar el programa.

# Constantes

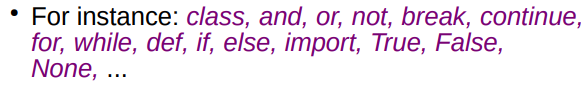
Hay lenguajes en los que se pueden definir constantes pero Python no es uno de ellos. Para sustituir las constantes podemos definir las variables en mayúsculas, así se podrán diferenciar y sabremos que no hay que modificar su valor.

Por ejemplo: PI=3.1416

# Palabras reservadas

Son términos ya definidos con un significado concreto. Por consiguiente estas palabras no se pueden usar para nombrar otros elementos o variables.

En Python hay más de 30 keywords.



<https://www.w3schools.com/python/python_ref_keywords.asp>

# Identificadores

Parecido a los nombres de las variables, engloba los nombres de las variables, nombres de las funciones, clases, etc. Básicamente cualquier cosa en un programa que requiera ser nombrado.

Las reglas para su nombramiento:

* Podemos utilizar letras (tanto mayus como minus), dígitos y barra baja.
* No pueden empezar con un dígito.
* Las palabras reservadas están prohibidas.
* Espacios en blanco y los símbolos especiales no están permitidos (: $, !, @, #, %, etc).

# Tipos numéricos

* Enteros (int) → sin decimales, pueden ser 0 o negativo
* Enteros largos (long) → con el doble de bits/bytes, para expresar grandes números.
  + Por ejemplo 10bits=1024 y pasaríamos a tener 20bits=1.048.576
* Reales (float) → números de punto flotante de doble precisión (decimales)
* Complejos (complex) → suele representarse con i, tienen parte real e imaginaria

# Tipos de texto

Se delimitan con comillas simples o dobles, hay que ser consistentes con su uso.

Las comillas triples (poner tres veces las comillas simples ‘’’ o dobles “””) sirven para los comentarios de varias líneas y asignar valores largos

Se pondrá como str o unicode.

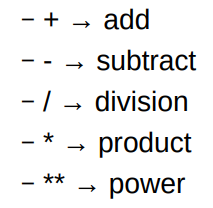
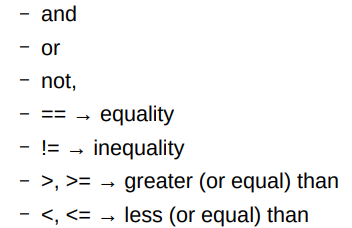
# Tipos booleanos

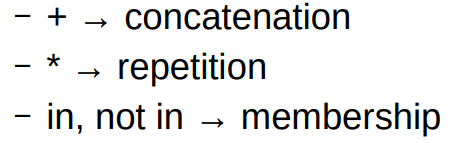
True o False

Son usados para evaluar condiciones

Algunos valores se pueden interpretar como False (0, None, cadena de texto vacía)

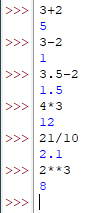
# Operadores básicos

Numéricos: Booleanos: Texto:

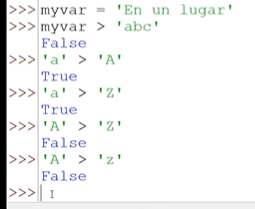
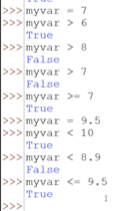
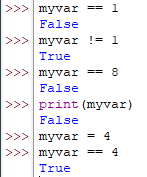
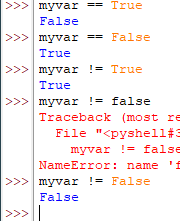
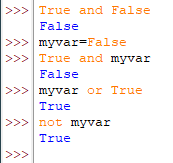


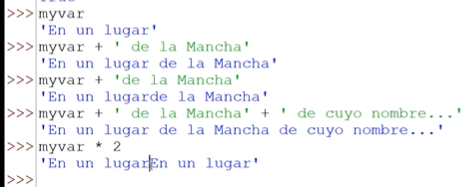
## Ejemplos

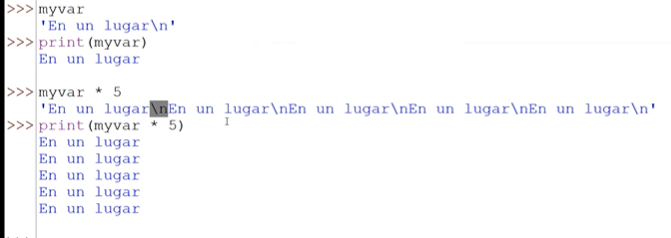
Numéricos:

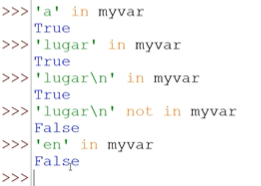


Booleanos:

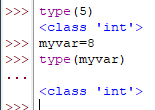


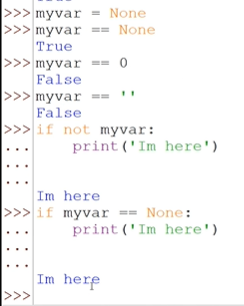
Texto:  






Extra:





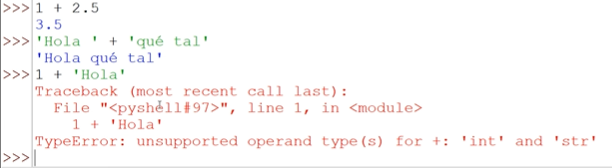
# Expresiones

Pueden formarse por valores individuales o variables.

Para construir una expresión, además de los valores, se necesitan las siguientes herramientas o elementos:

* Operadores
* Paréntesis
* Otras expresiones

Una expresión es correcta cuando su sintaxis (cuando se siguen las reglas del lenguaje, estructura, etc) y semántica (comprueba que los operandos son del tipo que se esperan, por ejemplo no sumar un número con una cadena de texto) son correctas.



# Asignaciones

Se compone de tres partes:

1—------- 2—--- 3—-----------

variable = expresión

\_\_\_\_\_\_\_\_\_