## POLITICAS PYTHON

## **Tokens**

* Se denominan *tokens* a los componentes léxicos del lenguaje de programación.

Los tipos de *token* son: identificadores, palabras clave, operadores, delimitadores y literales.

Para separar los *tokens* se utilizan espacios.

## **Identificadores**

Un *identificador* es el nombre empleado para identificar una variable, una función, una clase, un módulo u otro objeto.

* Un identificador comienza con una letra (de la A a la Z o de la a a la z) o con un guión bajo (\_) guiones bajos y números.
* **Python NO permite** signos de puntuación como @, $ y %, excepto el guión bajo (\_).
* Se recomienda nombrar las clases comenzando por una letra mayúscula y el resto de los identificadores por una letra minúscula.

*# Identificadores validos:*

radio

RaDio

contAlumnos

Cont\_Alumnos

num1

*# Identificadores NO validos:*

2E2

Miércoles

Cont-Alumnos

Conjunto Primero

print *#palabras reservadas*

1\_cantidad

Recuerda que Python distingue las mayúsculas de las minúsculas, entonces por ejemplo **area**, **Area**, **AREA**, **aRea**, son 4 identificadores distintos. También comentar que, el último ejemplo, **aRea** no es recomendado, dado su poca legibilidad.

**Palabras clave**

* Python tiene 30 palabras reservadas que no podremos utilizar para ningún tipo de identificador:
* Las palabras reservadas de Python son las siguientes:

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

**Operadores**

* Python utiliza caracteres no alfanuméricos y combinación de caracteres como operadores.

Se reconocen los siguientes operadores:**+ - \* /  % / < <> > < >= ==**

* Operaciones que aparezcan más arriba se evalúan después que las que aparecen más abajo.
* Operaciones de la en la misma fila se evalúan de izquierda a derecha.[[1]](https://es.wikibooks.org/wiki/Python/Generalidades/Tabla_de_operadores_con_asociatividad_y_precedencia#cite_note-1)

**Delimitadores**

* Python utiliza los siguientes símbolos y combinación de símbolos como delimitadores en expresiones, listas, diccionarios, etc: **( ) [ ] { ,  : . ` =  ; += -= \*= /= //=  %= &= |= ^= >>= <<= \*\*=**
* Los caracteres **$** y **?** y todos los caracteres de control nunca podrán ser parte de un programa Python, excepto en comentarios y *strings*.
* La utilización de **@** también está limitado en algunas versiones.

**Literales**

* Un *literal* es un número o un *string* que aparece directamente en un programa.

Los siguientes son literales en Python:

***Definir dentro del código:***

1. **Empieza con comillas**
2. **Puede venir cualquier carácter especial**
3. **Salida con el mismo numero de comillas del inicio**
4. **El limite de comillas es de 3 para definir una constante string**

***Dentro de variables***

***\*No contiene reglas del tipo string***

42 *# Entero*

3.14 *# Punto flotante*

1.0j *# Imaginario*

'hola' *# String*

**"mundo"** *# Otro string*

*"""Buenas*

*noches"""* *# String limitado con tres comillas*

**Constantes enteras y reales**

* 1. **Empieza con una constante entera**
  2. **Puede seguir dígitos/ punto**
  3. **No pueden ir mas de 2 puntos**
  4. **Después del punto debe ir por lo menos 1 digito**
  5. **Puede seguir dígitos/ punto**
  6. **Si comienza con digito el siguiente carácter deberá llevar**
  7. **No puede llevar mas de 1 punto**
  8. **Después del punto el siguiente carácter deberá ser un digito**