

▼ Python

Python es un lenguaje de programación que te permite trabajar rápidamente e integrar sistemas de manera más eficaz. Aprende python aquí: <https://wiki.python.org/moin/BeginnersGuide/Programmers>

▼ Básico

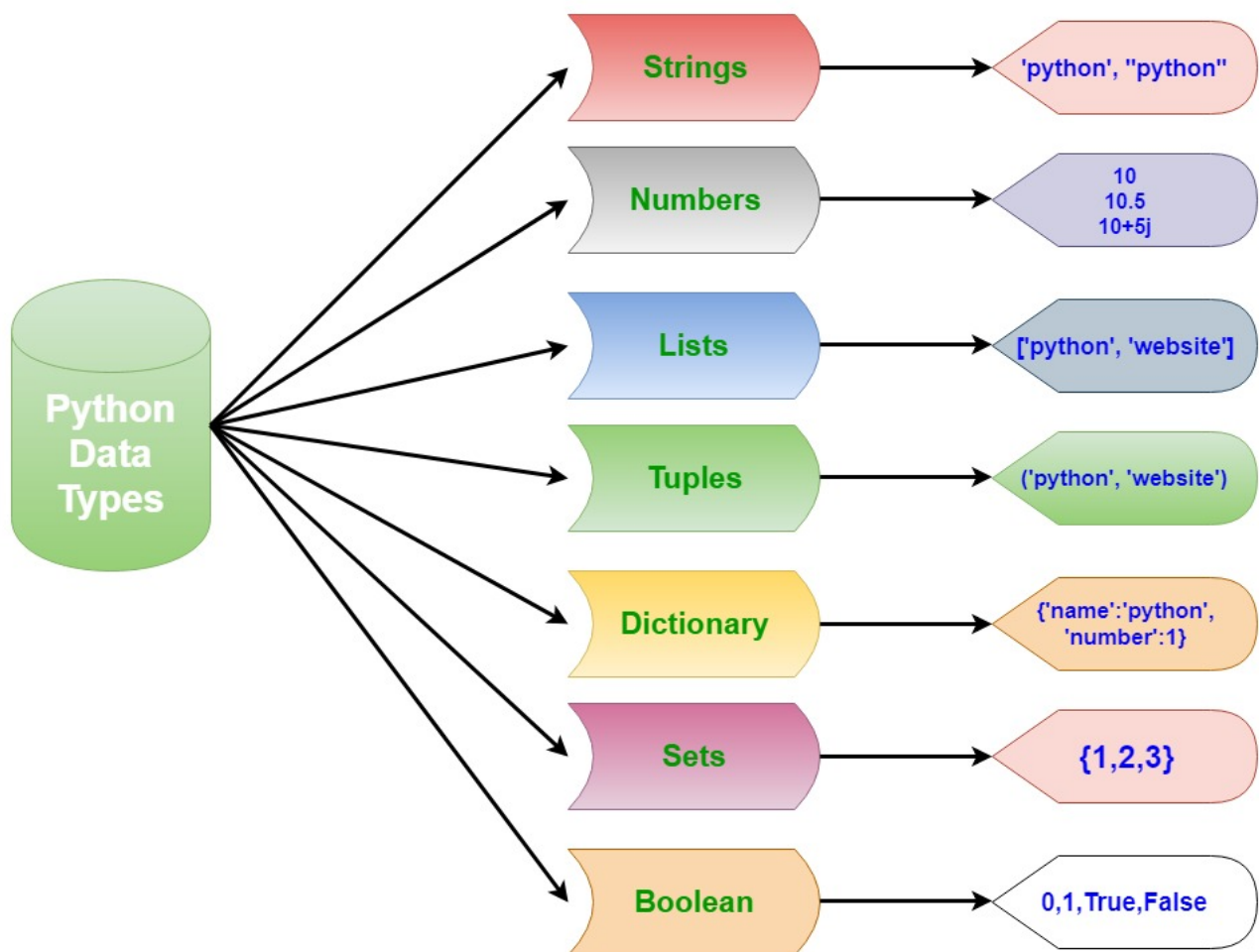
Python es un lenguaje muy simple, y tiene una sintaxis directa. Motiva a los programadores a programar sin código repetitivo.

```
1 # imprimir una línea "Esto es una función print"
2 variable = "Hola mundo"
3 print(variable)

Hola mundo
```

▼ Variables y tipos

Python está completamente orientado a objetos, y no a "tipos estáticos". No necesitas declarar variables antes de usarlas, o declarar su tipo. Cada variable en python es un objeto.



```
1 #Guarda un int e imprime por pantalla
2 myint = 7
3 myint # la última línea del código imprime

7
```

```
1 #Guarda un float e imprime por pantalla
2 myfloat = 7
3 print(myfloat)

7
```

```
1 #Guarda un string e imprime por pantalla
2 mystring = "hola"
3 print (mystring)
```

```
hola
```

```
1 #Guarda dos números, sumalos e imprime por pantalla
2 one = (1)
3 two = (2)
4 print ("one"+"two")
5
```

```
onetwo
```

```
1 # from _typeshed import ProfileFunction
2 #Guarda estos dos strings con un espacio en el medio
3 hello = "hola"
4 world = "mundo"
5 print("hello"+" "+"mundo")
```

```
hello mundo
```

```
1 #Mira como se asignan dos valores a la vez
2 a, b = 3, 4
3 a, b
```

```
(3, 4)
```

▼ Listas

Las listas son similares a los arrays. Pueden contener cualquier tipo de variable, y tantas variables como quieras. Las listas también se pueden iterar de una manera muy sencilla.

```
1 #Guarda una lista e imprime por pantalla
2 # primero utilizamos la orden mylist /despues el igual y entre corchetes separados por ,/ los elementos de la lista
3 mylist=[1,2,True,3]
4 mylist
```

```
[1, 2, True, 3]
```

```
1 #Guarda una lista con diferentes tipos e imprime por pantalla
2 mylist =[1,"hola", True]
3 mylist
```

```
[1, 'hola', True]
```

```
1 #Comprueba el tipo del objeto anterior
2 # primero utilizamos type (tipo)/ despues espacio / y entre () el nombre
3 type (mylist)
```

```
list
```

```
1 #Guarda valores en la siguiente lista usando append() e imprime por pantalla
2 mylist = [] # esta lista está vacia
3 mylist.append("hola")# nombre de la lista mylist /. append está orden añade un elemento a una lista/ y entre parentesis y comillas el
4 mylist.append("adios")
5 mylist.append(1)
6 mylist # para comprobar que lo ha añadido
```

```
☞ ['hola', 'adios', 1]
```

```
1 #Imprime el valor de una posición solo
2 # primero la orden print / despues entre parentesis: el nombre de la lista abrimos corchetes la posicion del elemento cerramos corchet
3 print (mylist[2])
4
```

```
1
```

```
1 #Guarda valores en la siguiente lista usando insert() e imprime por pantalla
2 mylista = ["casa","calle"]
3 mylista.insert(0,"pueblo") #primero nombre de la lista mylista /despues .insert / abre parentesis le decimos la posición donde quere
4 mylista
```

```
['pueblo', 'casa', 'calle']
```

► Operadores

[] ↪ 7 celdas ocultas

► Formato de cadena (string)

El formato de string es útil para insertar variables en cadenas.

[] ↪ 3 celdas ocultas

► Condiciones

Las condiciones permiten ejecutar código solo si se cumplen ciertas circunstancias.

[] ↪ 4 celdas ocultas

► Bucles (Opcional)

Los bucles permiten ejecutar el mismo código hasta que se cumpla cierta condición.

[] ↪ 8 celdas ocultas

► Funciones (Opcional)

Las funciones son una forma de dividir el código en bloques, permitiéndonos ordenarlo, hacerlo más legible, reutilizarlo y ahorrar tiempo. Las funciones también son importantes para definir interfaces para que los programadores compartan su código.

[] ↪ 6 celdas ocultas

► Diccionarios (Opcional)

Un diccionario es un tipo de dato similar a los arrays, pero funciona con claves y valores en vez de índices. Se puede acceder a cada valor almacenado en un diccionario usando una clave, que es cualquier tipo de objeto (una cadena, un número, una lista, etc.) en lugar de usar su índice para abordarlo.

[] ↪ 4 celdas ocultas

► Módulos y Paquetes (Opcional)

En programación, un módulo es una pieza de software con una funcionalidad específica. Por ejemplo, al crear un juego de ping pong, un módulo sería responsable de la lógica del juego y otro módulo se encargaría de dibujar el juego en la pantalla. Cada módulo es un archivo diferente, que se puede editar por separado.

[] ↪ 11 celdas ocultas

► Avanzado (Opcional)

[] ↪ 42 celdas ocultas

✓ 0 s completado a las 13:22

×