



# INFORMATARIO



101010101010 >>>>



**Globant** ➤



Subsecretaría de  
**Empleo**  
Chaco Gobierno de todos



Ministerio de  
**Producción, Industria y Empleo**  
Chaco Gobierno de todos



**CHACO**  
Gobierno de todos



# INFORMATARIO



## Iniciando a programar en Python

Manejo de variables y tipos de datos



Subsecretaría de  
**Empleo**  
Chaco Gobierno de todos



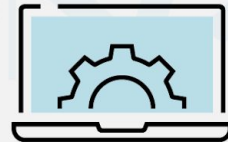
Ministerio de  
**Producción, Industria y Empleo**  
Chaco Gobierno de todos



**CHACO**  
Gobierno de todos



Son elementos que nos permiten guardar un dato para que pueda ser accesible fácilmente, manipulado y transformado a lo largo de un programa.



Para definir una variable en Python, basta con nombrarla y definir su valor

Para crear y darle valores a nuevas variables utilizamos la operación de asignación (=)

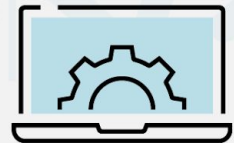
Para modificar el valor de una variable en Python, basta con asignarle un nuevo valor después de definirla



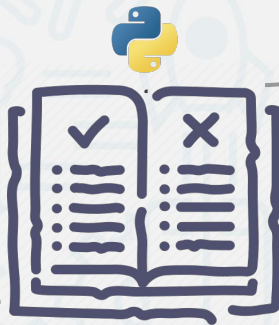
Se puede asignar a una variable un valor literal, una expresión, una llamada a una función o una combinación de todos ellos



010010  
10110



Pueden empezar con una letra o un guión bajo, por convención no usamos mayúsculas.



Pueden contener letras, números y se puede usar el guión bajo (\_).

Son "case sensitive", es decir, distingue si contienen mayúsculas o minúsculas.

No se puede utilizar palabras claves o reservadas de Python como nombres de variables



010010  
10110



## Matemáticos

```
operadores.py
1 #Suma
2
3 suma = 20 + 30
4
5 print("El resultado de la suma es:", suma)
6
```

**Suma**

Prueba ingresando el ejemplo del video en el editor de texto y luego ejecuta el archivo .py en la consola

```
Microsoft Windows [Versión 10.0.18363.836]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\nati\Documents\Informatorio\Ejemplos>python operadores.py
El resultado de la suma es: 50

C:\Users\nati\Documents\Informatorio\Ejemplos>
```



010010  
10110



Subsecretaría de  
**Empleo**  
Chaco Gobierno de todos



Ministerio de  
**Producción, Industria y Empleo**  
Chaco Gobierno de todos



**CHACO**  
Gobierno de todos

```
File Edit View Selection Find Packages Help
operadores.py
1  #Resta
2
3  resta = 50 - 10
4
5  print("El resultado de la resta es:", resta)
6
```

### Resta

Prueba ingresando el ejemplo del video en el editor de texto y luego ejecuta el archivo .py en la consola

```
Microsoft Windows [Versión 10.0.18363.836]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\nati\Documents\Informatorio\Ejemplos>python operadores.py
El resultado de la resta es: 40

C:\Users\nati\Documents\Informatorio\Ejemplos>
```



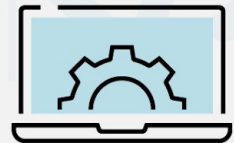
010010  
10110





## División

Prueba ingresando el ejemplo  
del video en el editor de  
texto y luego ejecuta el  
archivo .py en la consola



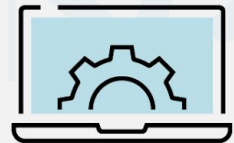


## Multiplicación

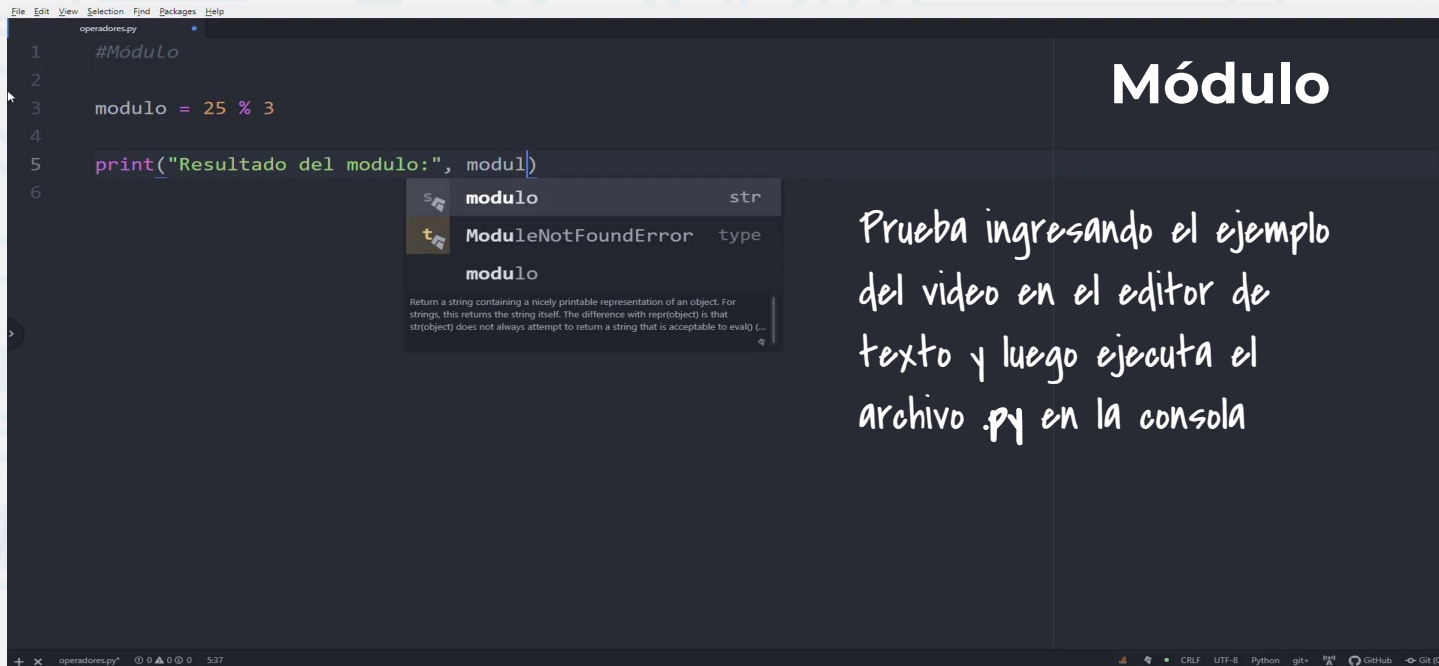
Prueba ingresando el ejemplo del video en el editor de texto y luego ejecuta el archivo .py en la consola



010010  
10110



## Matemáticos



The screenshot shows a Python IDE with a file named `operadores.py`. The code in the editor is:

```
1 #Módulo
2
3 modulo = 25 % 3
4
5 print("Resultado del modulo:", modul)
6
```

An autocomplete dropdown menu is open at line 5, showing the following options:

- `modulo` (type `str`)
- `ModuleNotFoundError` (type `type`)
- `modulo` (type `str`)

The description for the selected `modulo` option is: "Return a string containing a nicely printable representation of an object. For strings, this returns the string itself. The difference with `repr(object)` is that `str(object)` does not always attempt to return a string that is acceptable to eval [...]"

### Módulo

Prueba ingresando el ejemplo del video en el editor de texto y luego ejecuta el archivo .py en la consola



010010  
10110



Subsecretaría de  
**Empleo**  
Chaco Gobierno de todos



Ministerio de  
**Producción, Industria y Empleo**  
Chaco Gobierno de todos



**CHACO**  
Gobierno de todos

## Potencia

Prueba ingresando el ejemplo  
del video en el editor de  
texto y luego ejecuta el  
archivo .py en la consola



010010  
10110



## Lógicos

```
operadores.py
1  a = True
2  b = True
3  c = False
4  #AND
5  x = a and b
6  print("Resultado AND:", x)
7
8  #OR
9  y = c or c
10 print("Resultado OR:", y)
11
12 #NOT
13 z = not a
14 print("Resultado NOT:", z)
15
```

(c) 2019 Microsoft Corporation. Todos los derechos reservados.

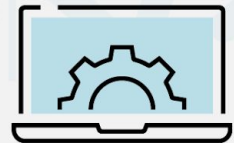
```
C:\Users\nati\Documents\Informatorio\Ejemplos>python operadores.py
Resultado AND: True
Resultado OR: False
Resultado NOT: False

C:\Users\nati\Documents\Informatorio\Ejemplos>
```

Prueba ingresando el ejemplo del video en el editor de texto y luego ejecuta el archivo .py en la consola



010010  
10110



## Strings

File Edit View Selection Find Packages Help

```
operadores.py
1  #Concatenación
2  concatenacion = "Hola " + "mundo"
3  print("Resultado concatenación:", concatenacion)
4
5  #Multiplicación
6  multiplicacion = 'Hola' * 3
7  print("Resultado multiplicación:", multiplicacion)
8
9  #Mezcla
10 mezcla = 'Hola' * 3 + ' mundo'
11 print("Resultado mezcla:", mezcla)
12
```

Prueba ingresando el ejemplo del video en el editor de texto y luego ejecuta el archivo .py en la consola

(c) 2019 Microsoft Corporation. Todos los derechos reservados.

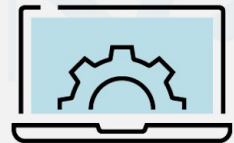
```
C:\Users\nati\Documents\Informatario\Ejemplos>python operadores.py
Resultado concatenación: Hola mundo
Resultado multiplicación: HolaHolaHola
Resultado mezcla: HolaHolaHola mundo
```

```
C:\Users\nati\Documents\Informatario\Ejemplos>
```

+ [X] X operadores.py 0 0 0 0 10:31



010010  
10110



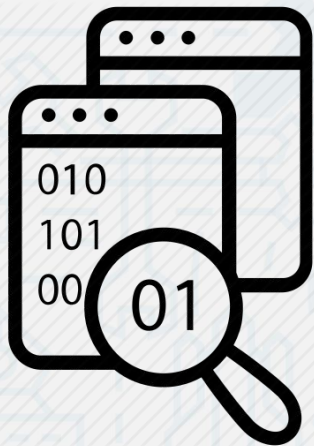
Subsecretaría de  
**Empleo**  
Chaco Gobierno de todos



Ministerio de  
**Producción, Industria y Empleo**  
Chaco Gobierno de todos

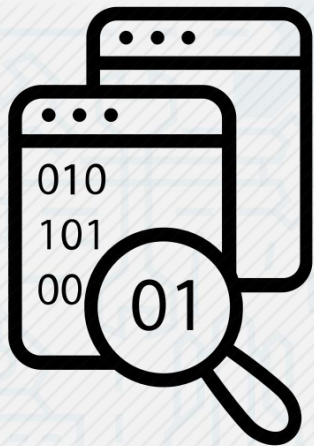


**CHACO**  
Gobierno de todos



Es una representación simbólica (numérica, alfabética, etc.) de un atributo o cualidad de una entidad.





Los datos aisladamente pueden no contener información pero si los analizamos en conjunto sirven para realizar cálculos o tomar decisiones.







En Python todos los datos son referencias a objetos (no son constantes ni variables), son como etiquetas para poder identificarlos

Los objetos en Python pueden ser 'mutables' si pueden modificarse algunas posiciones del dato o 'inmutables' si no pueden modificarse.





El tipo de un dato está definido por el conjunto de valores que puede tomar a lo largo de un programa.

Para ver los tipos de datos a continuación usaremos la función `type()`, que nos devuelve el tipo de objeto que enviamos como argumento



## Números

```
File Edit View Selection Find Packages Help
tipos_de_datos.py
1 #Prueba Entero
2 entero = 20
3
4 print(type(en|))
```

entero	int
EnvironmentError	type
enumerate(...)	function
enc	# coding=utf-8
env	#!/usr/bin/env python
type(...)	dict

samples1 Docs

4:14

CRLF UTF-8 Python git+ GitHub Git (0)

### Entero

Prueba ingresando el ejemplo del video en el editor de texto y luego ejecuta el archivo .py en la consola



## Números

**Real**

Prueba ingresando el ejemplo  
del video en el editor de  
texto y luego ejecuta el  
archivo .py en la consola



## Números

```
File Edit View Selection Find Packages Help
tipos_de_datos.py
1 #complejo
2 complejo = 2.1 + 6j
3
4 print(type(complejo))
```

complejo complex | float

compile(...) function

complex type

Convert a string or a number to floating point. If the argument is a string, it must contain a possibly signed decimal or floating point number, possibly embedded in whitespace. The argument may also be (+)jnan or (+)jinf. Otherwise, the argument may be a plain or long...

type(...) dict

### Complejo

Prueba ingresando el ejemplo del video en el editor de texto y luego ejecuta el archivo .py en la consola



## Cadena o String

Prueba ingresando el ejemplo  
del video en el editor de  
texto y luego ejecuta el  
archivo .py en la consola



## Booleano

```
File Edit View Selection Find Packages Help
tipos_de_datos.py
1 #Booleano
2
3 booleano = True
4
5 print(type(booleano))
```

booleano bool  
bool type  
Booleano  
type(...) dict

Prueba ingresando el ejemplo del video en el editor de texto y luego ejecuta el archivo .py en la consola





## Conjunto

```
File Edit View Selection Find Packages Help
tipos_de_datos.py
1 #Conjunto
2 conjunto = {'naranja', 1, 'c', 2.5, True, 'ciruela'}
3
4 print(type(conjunto))
5 type(...) -> dict
6
```

Prueba ingresando el ejemplo del video en el editor de texto y luego ejecuta el archivo .py en la consola

```
Microsoft Windows [Versión 10.0.18363.836]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\nati\Documents\Informatorio\Ejemplos>python tipos_de_datos.py
<class 'set'>

C:\Users\nati\Documents\Informatorio\Ejemplos>
```



## Lista

```
File Edit View Selection Find Packages Help
tipos_de_datos.py
1 #Lista
2
3 lista = [0.5 , 'manzana', 1500 , 'azul']
4
5 print(type(lista))
   type(...) -> dict
   samples1 Docs P

Microsoft Windows [Versión 10.0.18363.836]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\nati\Documents\Informatorio\Ejemplos>python tipos_de_datos.py
<class 'list'>

C:\Users\nati\Documents\Informatorio\Ejemplos>
```

Prueba ingresando el ejemplo del video en el editor de texto y luego ejecuta el archivo .py en la consola



## Tupla

```
File Edit View Selection Find Packages Help
tipos_de_datos.py
1 #Tupla
2
3 tupla = 1, 25, 1500, 'Hola Mundo'
4
5 tupla_anidada = tupla, True, "Hola", 5
6
7 print(type(tupla))
8
9 print(type(tupla))
```

Prueba ingresando el ejemplo del video en el editor de texto y luego ejecuta el archivo .py en la consola

tupla	tuple
tupla_anidada	tuple
TabError	type
TimeoutError	type
True	bool

Microsoft Windows [Version 10.0.17134.0] (c) 2019 Microsoft Corporation. Todos los derechos reservados.  
C:\Users\nati>python tipos\_de\_datos.py  
type(...) dict



## Diccionario

```
File Edit View Selection Find Packages Help
tipos_datos.py
1 #Diccionario
2
3 diccionario = {
4     1: "Susana Gimenez",
5     5: "Marcelo Tinelli",
6     7: "Mirtha Legrand"
7 }
8
9 print(ty)
10
11
```

Prueba ingresando el ejemplo del video en el editor de texto y luego ejecuta el archivo .py en la consola

Microsoft (c) 2019

ty	type
TypeError	type
type	type
try	Try/Except
trye	Try/Except/Else
tryef	Try/Except/Else/Finally
tryf	Try/Except/Finally

choso reservados.

With one argument, return the type of an object. The return value is a type object. The `isinstance()` built-in function is recommended for testing the type of an object. With three arguments, return a new type object. This is essentially a dynamic form of the class statement. The...



## None

```
File Edit View Selection Find Packages Help
tipos_datos.py
1 #None
2
3 x = None
4
5 print(type)
6
```

TypeError type

type type

try Try/Except

trye Try/Except/Else

tryef Try/Except/Else/Finally

tryf Try/Except/Finally

With one argument, return the type of an object. The return value is a type object. The `instance()` built-in function is recommended for testing the type of an object. With three arguments, return a new type object. This is essentially a dynamic form of the class statement. The...

Docs »

Microsoft Windows [Versión 10.0.18363.836]  
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\nati\Documents\Informatorio\Ejemplos>

Prueba ingresando el ejemplo del video en el editor de texto y luego ejecuta el archivo .py en la consola





**Prueba los ejemplos en tu computadora y si tienes dudas  
escribenos a través del Foro**



Subsecretaría de  
**Empleo**  
Chaco Gobierno de todos



Ministerio de  
**Producción, Industria y Empleo**  
Chaco Gobierno de todos



**CHACO**  
Gobierno de todos