ntic master
revolucionamos la comunicación

UNIVERSIDAD COMPLUTENSE DE MADRID

Programación Python

José Javier Galán Hernández

ntic master
revolucionamos la comunicación

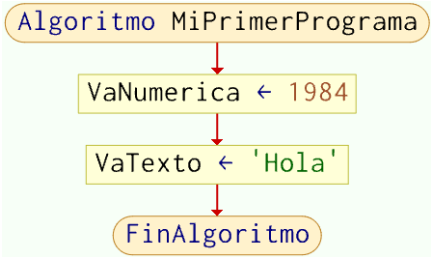
UNIVERSIDAD COMPLUTENSE DE MADRID

Programación Python

4. Variables

4. Variables

Los datos que manejamos durante la ejecución de un programa deben ser almacenados para su tratamiento, este almacenamiento se hará por defecto de forma temporal en la memoria principal del ordenador hasta que se decida almacenar en la memoria física. Estos datos sufrirán cambios a lo largo de la ejecución del programa, su contenido variara y por eso lo llamaremos variables.




```
Algoritmo MiPrimerPrograma
    VaNumerica  ←1984
    VaTexto     ←'Hola'
FinAlgoritmo
```

El tipo de dato que podemos almacenar en cada variable será distinto dependiendo del uso que queramos darle, no es lo mismo disponer de variables para operaciones aritméticas que contendrán números que variables cuyo contenido es texto y que por lo tanto las letras que contienes no se podrán sumar entre sí.

4. Variables

Los tipos de datos primitivos más comunes en Python son:

Tipo	Descripción
bool	Booleano
int	Numérico
str	Texto

Python es un lenguaje de tipos (llamado tipado), esto quiere decir que el tipo de dato que almacena puede cambiar a lo largo de la ejecución. Por ello aunque usaremos en Python el término variable, por motivos didácticos, sería más correcto hablar de tipos.

En Python no es necesario crear una variable ni definir su tipo. Directamente podemos asignarla valor y utilizarla.

4. Variables

Ejemplo:

```
Valor1=1
Valor2=2
Resultado=Valor1+Valor2
Resultado=True
Resultado="Hola"
Resultado
```

```
'Hola'
```

```
Resultado=Resultado+Valor1
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [7], in <cell line: 1>()
----> 1 Resultado=Resultado+Valor1

TypeError: can only concatenate str (not "int") to str
```

4. Variables

Usamos **None** para indicar la ausencia de valor y **type** para conocer de qué tipo es una variable.

`Valor1=None` `Valor1`	Indicamos con None que la variable Valor1 no tiene valor, por ello al intentar mostrarla por pantalla no muestra nada.

`type(Valor1)`
 	
`NoneType`
	Continuando con el ejemplo anterior usamos type sobre la variable Valor1. Como no tiene valor el tipo mostrado por pantalla es NoneType.

`Valor1=2`
`type(Valor1)`
 	
`int`
	A la variable Valor1 le asignamos el valor 2, por ello al preguntar cual es su tipo mediante el comando type obtenemos como resultado int.

`Valor1="Hola"`
`type(Valor1)`
 	
`str`
	A la variable Valor1 le asignamos el valor “Hola”, por ello al preguntar cuál es su tipo mediante el comando type obtenemos como resultado str.

`Valor1=False`
`type(Valor1)`
 	
`bool`
	A la variable Valor1 le asignamos el valor False, por ello al preguntar cuál es su tipo mediante el comando type obtenemos como resultado bool.

ntic master
revolucionamos la comunicación

UNIVERSIDAD COMPLUTENSE DE MADRID

Programación Python

5. Tipos y operadores

5. Tipos y operadores. Conversión de tipo

Con el comando **int()** podemos convertir texto en número

```
Valor1=20
Valor2="2"
Resultado=Valor1+Valor2
Resultado
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [22], in <cell line: 3>()
      1 Valor1=20
      2 Valor2="2"
----> 3 Resultado=Valor1+Valor2
      4 Resultado

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

```
Valor1=20
Valor2="2"
Resultado=Valor1+int(Valor2)
Resultado
```

```
22
```

5. Tipos y operadores. Conversión de tipo

Con el comando **str**() podemos convertir números en texto siempre que sea posible

```
Valor1="El numero elegido es el: "
Valor2=2
Frase=Valor1+Valor2
Frase
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [24], in <cell line: 3>()
      1 Valor1="El numero elegido es el: "
      2 Valor2=2
----> 3 Frase=Valor1+Valor2
      4 Frase

TypeError: can only concatenate str (not "int") to str
```

```
Valor1="El numero elegido es el: "
Valor2=2
Frase=Valor1+str(Valor2)
Frase
```

```
'El numero elegido es el: 2'
```

5. Tipos y operadores. Operadores Lógicos

Operador lógico	Descripción
And	Deben cumplirse todas las condiciones
Or	Debe cumplirse alguna condición
Not	Negación de la condición

```
#Usamos == porque no es una asignacion, es pregunta.
Valor1=2
Valor2=5
(Valor1==2 and Valor2==5)
```

```
True
```

```
Valor1=2
Valor2=5
(Valor1==2 and Valor2==2)
```

```
False
```

```
Valor1=2
Valor2=5
(Valor1==2 or Valor2==2)
```

```
True
```

```
Valor1=2
Valor2=5
not(Valor1==2 or Valor2==2)
```

```
False
```

Como se cumplen todas las condiciones, usando **AND**, el resultado es verdadero
Como no se cumplen todas las condiciones, usando **AND**, la evaluación en falsa
Con **OR** solo es necesario que se cumpla alguna condición, por lo tanto a diferencia del ejemplo anterior este caso es verdadero.
Con **NOT** conseguimos el valor lógico contrario porque lo negamos, como es verdadero lo convertimos en falso.

5. Tipos y operadores. Operadores Comparativos

Operador comparativo	Descripción
==	Igual a
!=	Distinto a
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que

```
Valor1=2
Valor2=5

Valor1==Valor2
```

```
False
```

Valor1 no es igual que Valor2, es falso.

```
Valor1=2
Valor2=5

Valor1!= Valor2
```

```
True
```

En este caso como no son iguales, pero es lo que preguntamos, la respuesta es verdadero.

```
Valor1=2
Valor2=5

Valor1>=Valor2
```

```
False
```

Valor1 no es mayor ni igual que Valor2, no se cumple, siendo falso.

5. Tipos y operadores. Operadores Aritméticos

Operador aritmético	Descripción
+	Suma
-	Resta
*	Multiplicación
/	División
**	Exponente
//	División entera
%	Modulo

```
print("Introduce el primer valor: ")
Valor1=int(input()) #Convertimos el texto introducido en numero, error si se introduce texto
print("Introduce el segundo valor: ")
Valor2=int(input())

resultado=Valor1+Valor2
print("El resultado de la suma es: " + str(resultado))
```

```
Introduce el primer valor:
2
Introduce el segundo valor:
4
El resultado de la suma es: 6
```

6. Entrada y Salida de datos.

input() obtiene el texto escrito por teclado.

print() puede recibir argumentos como end=“” que sustituye el salto de línea por el valor entre comillas, o valores que mostrar.

```
In [7]: print("Dame un numero")
        numero = input()
        print ("El numero introducido es: " + numero)

Dame un numero
4
El numero introducido es: 4
```

```
In [11]: print("Dame un numero ", end="")
         numero = input()
         print ("El numero introducido es: " + numero)

Dame un numero 4
El numero introducido es: 4
```

```
In [15]: print("Dame un numero ", end="")
         numero = input()
         print ("El numero introducido es: ", numero)

Dame un numero 4
El numero introducido es:  4
```

…ahora…comencemos!

Abramos Jupyter y creemos un proyecto llamado MiPrimerPrograma!

Ejercicio 1	*Crea 2 variables, Variable1 con valor numerico=5 y Variable2 con valor texto="Hola"*
Ejercicio 2	*Crea 2 variables, Variable1 y Variable2 que reciben valor por teclado. Despues muestra su multiplicación*
Ejercicio 3	*Pide tu nombre por teclado y muestra un mensaje personalizado deseandote un buen dia*
Ejercicio 4	*Pide 3 variables: Variable1, Variable2 y Variable3. Introduce en ellas por teclado valores numéricos. Si son iguales mostrara True, False en caso contrario. (Todavía no podemos usar IF)*
Ejercicio 5	*Pide 3 variables: Variable1, Variable2 y Variable3. Introduce en ellas por teclado valores numéricos. Si alguna es mayor de 100 mostrara True, False en caso contrario. (Todavía no podemos usar IF)*

ntic master
revolucionamos la comunicación

UNIVERSIDAD COMPLUTENSE DE MADRID