PYTHON

Variables, salida y entrada de datos – 100 días de Python #1

PROGRAMACIÓN FÁCIL
AGOSTO 27, 2022

Empezamos nuevo curso viendo las variables, la salida y entrada de datos, concatenación y mucho más.

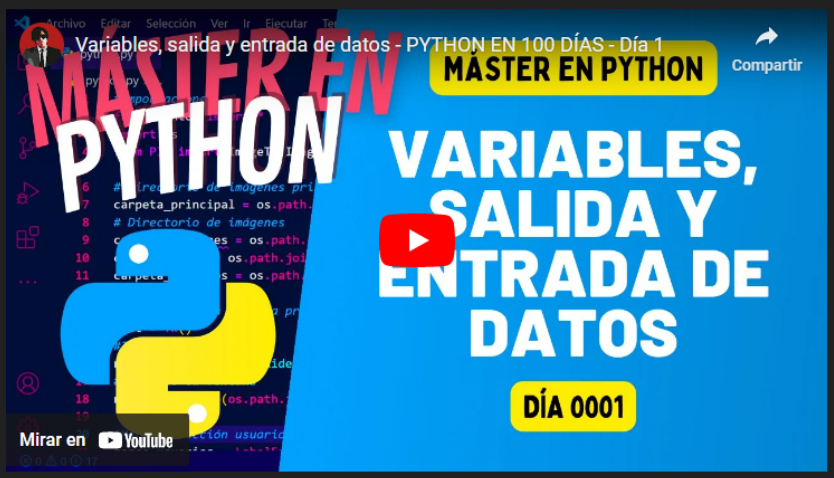

Tabla de contenidos

1. Imprimir en la consola
2. Variables
   2.1. Asignar variables a otras variables
   2.2. Reasignación de valores en variables
3. Finalización de instrucciones
4. Espacios en blanco y tabulaciones
5. Entrada de datos en el programa por el usuario
6. Saltos de línea en Python
7. Concatenación
8. Comentarios

Imprimir en la consola

Lo primero que se suele hacer cuando empiezas en cualquier lenguaje de programación, es imprimir (mostrar) algo en la consola, para comprobar que todo está correcto. Esto lo haremos con la función de Python print(). Escribe el mensaje que quieras.

Solo tienes que escribir esto en un archivo con extensión .py y escribir sin borrar las comillas, lo que quieras.

Python

```
1 print("Primer día de Python.")
```

Dale al botón de ejecutar sin depurar (CTRL+F5 en VSCode).

Si todo ha ido bien, te imprimirá el mensaje en la consola ¡Enhorabuena si lo has conseguido! Ya tienes creado tu primer programa en Python.

> **Resultado en la consola**
>
> **Primer día de Python.**

Variables

Las variables permiten almacenar datos de todo tipo. Vamos a crear una y la vamos a imprimir. Así aprenderás lo siguiente de un plumazo:

- Declarar variables
- Convención de escritura
- Inicializarlas
- Accederlas
- Reasignarlas

En Python, para declarar (crear) una variable. Le daremos un nombre. Este no debería contener acentos ni ñ, ni ç. Tampoco debe empezar por un número o un símbolo, excepto un guión bajo (_). La convención de nombres para variables en Python, dice que se haga en snake_case, todo en minúsculas y con una barra baja de separador de palabras. En conclusión, los siguientes nombres de variables, son correctos:

```python
nombre_variable
_nombre_variable
nombre
numero1
```

Los siguientes nombres de variables, son incorrectos:

```python
nombre-variable
1numero
nombre$variable
nombre variable
```

Estos nombres no te darán errores, pero no son recomendables:

```python
número1
China
España
adreça
```

Esta variable se inicializa con el símbolo «=» (que representa asignación, no igualdad). Una vez utilizas este símbolo y le das un valor, como el texto que ves en el código, la variable ha sido inicializada.

Ahora, solo hay que accederla desde el print() para poder imprimir su valor. Solo tienes que llamarla con su nombre, allá donde la necesites.

```python
frase_bienvenida = "Primer día de Python."

print(frase_bienvenida)
```

El resultado es el mismo que el de antes, solo que ahora, tenemos el string (texto) de forma reutilizable. Antes, escribiéndolo directamente en el print(), no se podía usar nada más que ahí mismo.

Asignar variables a otras variables

Si, suena raro, ¿verdad? Pues no lo es en realidad. Podemos asignar valores a variables utilizando los valores de otras. Aquí tienes un ejemplo:

```python
variable_1 = "Hola"

variable_2 = variable_1

print(variable_2)
```

El resultado es que ambas variables ahora tienen el mismo valor. Imprimiendo la segunda, podemos corroborarlo.

Resultado en la consola

Hola

Reasignación de valores en variables

Las variables contienen un dato. En cuanto se guarda en ellas otro dato diferente, el valor desaparece y se reemplaza por el nuevo.

```python
a = "Este es mi valor inicial"
print(a)

a = "Mi valor ha cambiado"
print(a)
```

Resultado en la consola

Este es mi valor inicial

Mi valor ha cambiado

Finalización de instrucciones

Te habrás percatado, si ya programas en otros lenguajes, que las instrucciones en casi cualquier lenguaje de programación, se finalizan con punto y coma. En Python no es habitual y te recomiendo no utilizarlo, solo que sepas que se permite su uso aunque no es nada común verlo en ningún código.

```python
frase_bienvenida = "Primer día de Python.";

print(frase_bienvenida);
```

Un uso realmente útil del punto y coma en Python, es la separación de instrucciones en la misma línea. Personalmente, no me gusta, pero que sepas que no es incorrecto:

```python
frase_bienvenida = "Primer día de Python."; print(frase_bienvenida)
```

Espacios en blanco y tabulaciones

Los espacios en blanco entre palabras, nombres de variables, instrucciones, son ignorados por el intérprete de Python. Debemos ir separando las cosas para que queden lo más legibles y ordenadas posibles.

Por otro lado, las tabulaciones en Python son muy importantes y no se ignoran. Hablaremos otro día de esto.

Entrada de datos en el programa por el usuario

Algo que vamos a necesitar muy frecuentemente, es la entrada de datos al programa. Es decir, poder guardar cosas en el programa para poder usarlas en él.

En Python, podemos hacer esto con la función input().

```python
nombre = input("Por favor, introduzca su nombre")

print(nombre)
```

La salida en la consola, es un poco confusa. Ahora lo mejoraremos. Tienes que escribir un nombre con el teclado y pulsar la tecla ENTER. La variable guardará lo que le has escrito. Eso es lo que se muestra en el print().

Resultado en la consola

Por favor, introduzca su nombreQuique

Quique

Saltos de línea en Python

Para presentar este input() mejor, podemos poner un salto de línea con esto: «\n».

```python
```

```
1 nombre = input("Por favor, introduzca su nombre\n")
2
3 print(nombre)
```

Esta vez, nos deja escribir en la línea de abajo. No te preocupes por que salga el nombre dos veces. La primera es la que hemos escrito nosotros. En un programa fuera de la consola, el usuario no verá esto. Ahora solo estamos practicando con la consola.

> **Resultado en la consola**
>
> **Por favor, introduzca su nombre**
> **Quique**
> **Quique**

Concatenación

Algo que vamos a necesitar al trabajar con strings (texto) es la concatenación. Esta se hace con el operador «+» y lo que hace es unir dos trozos de texto. Prueba ahora a añadir una frase con el nombre.

Python

```
1 nombre = input("Por favor, introduzca su nombre\n")
2
3 print("Bienvenido/a " + nombre + ".")
```

> **Resultado en la consola**
>
> **Por favor, introduzca su nombre**
> **Quique**
> **Bienvenido/a Quique.**

Primero, el mensaje de bienvenida. Dentro de las comillas, he dejado un espacio para que lo deje entre la palabra y el nombre que introduzca el usuario. Finalmente, una concatenación más para que se añada un punto y final.

Comentarios

Los comentarios son una parte imprescindible de todo lenguaje de programación. Permiten anular código que no queremos borrar por el momento. Esto con el fin de ir haciendo pruebas y ver como funciona parte del programa, con esas líneas anuladas o bien, para especificar que hace cada parte del código, con el fin de que al volver a editar en algún momento el código o que lo editen otros, sea todo más claro y fácil de realizar.

Para escribir un comentario, debemos utilizar el símbolo «#» y a continuación escribir lo que queramos. Esta línea es ignorada. No afecta en nada al código.

Python

```
1 # Este es un comentario de Python.
```

Podemos anular las línea que queramos así:

Python

```
1 print("Línea 1")
2 # print("Línea 2")
3 print("Línea 3")
4 # print("Línea 4")
```

Esto hace que al ejecutarlo, en la consola, solo estén las líneas que no hemos comentado:

> **Resultado en la consola**
>
> **Línea 1**
> **Línea 3**

También puedes añadir comentarios a la derecha del código:

Python

```
1 print("Línea 1")
2 print("Línea 2") # Esta línea es una más.
3 print("Línea 3")
```

```
print("Línea 4")
```

Esto no va a afectar en absoluto a la línea que tiene el comentario:

> **Resultado en la consola**
>
> **Línea 1**
> **Línea 2**
> **Línea 3**
> **Línea 4**

Hasta aquí el primer día de Python. Espero que haya sido productivo. Te recomiendo que descanses y que no te pases, aunque si quieres hacer varios días del curso en uno, no hay problema, lo acabarás antes.

Tienes unos cuantos ejercicios con soluciones y el proyecto del día en este enlace.

+7

EN CURSO PYTHON, ENTRADA DE DATOS PYTHON, PYTHON, PYTHON 100 DIAS, PYTHON EN ESPAÑOL, VARIABLES

Un comentario en «0»

Anderson garcia
febrero 17, 2023 a las 1:48 pm

Maestro muchas gracias de verdad, no tengo palabras para expresar lo que siento al ver personas como tú que se convierten en un ejemplo a seguir.

+1

Responder

DEJA UNA RESPUESTA

Tu dirección de correo electrónico no será publicada. Los campos obligatorios están marcados con *

Comentario *

Nombre *

Correo electrónico *

Web

Guarda mi nombre, correo electrónico y web en este navegador para la próxima vez que comente.

Publicar el comentario

Entrada relacionada

CUSTOMTKINTER MYSQL PYTHON TKINTER

Cómo crear una interfaz gráfica en Python para visualizar bases de datos de MySQL

PROGRAMACIÓN FÁCIL
MARZO 15, 2023

CUSTOMTKINTER MYSQL PYTHON TKINTER

Cómo crear una interfaz gráfica para consultas SQL en Python: Tutorial paso a paso

PROGRAMACIÓN FÁCIL
MARZO 8, 2023

CUSTOMTKINTER PYTHON TKINTER

Clase para funciones del programa con TK y CTK

PROGRAMACIÓN FÁCIL
MARZO 4, 2023

CUSTOMTKINTER PYTHON TKINTER

Bucle autogenerador de BOTONES y DESTRUIR ventanas con Tkinter y CustomTkinter

PROGRAMACIÓN FÁCIL
MARZO 2, 2023

CUSTOMTKINTER PYTHON TKINTER

Ventanas secundarias TopLevel con Tkinter y CustomTkinter

PROGRAMACIÓN FÁCIL
FEBRERO 27, 2023

CUSTOMTKINTER PYTHON TKINTER

Ventana de Login con CustomTkinter de Python

PROGRAMACIÓN FÁCIL
FEBRERO 23, 2023