

GUIA PYTHON

Ideas clave

- Un computador es como una calculadora inteligente.
- Un programa es un documento que le indica paso a paso qué hacer.
- El rol de un programador es resolver problemas escribiendo estos programas.
- Estos programas se pueden escribir en distintos lenguajes de programación.

¿Qué vamos a aprender en este curso?

En este curso interactivo aprenderemos a escribir estos programas en un lenguaje llamado **Python**, el cual es uno de los lenguajes de programación más populares en el mundo.

El primer ejercicio

En este primer ejercicio aprenderemos a escribir un programa que muestre el texto **hola mundo**. Para lograrlo, utilizaremos el editor de texto, el cual consiste en un editor donde podemos escribir instrucciones, y un botón para ejecutar estas instrucciones.

Python tiene muchas versiones, pero en este curso utilizaremos la versión 3.8. Si por algún motivo ocupas código de internet para resolver problemas y ese código no es compatible entonces probablemente obtendrás un error.

Ejercicio

Haz clic en el editor de texto y escribe la siguiente instrucción:

```
print("hola mundo")
```

Luego haz clic en el botón **Ejecutar código**. Si todo sale bien, deberías ver el texto **hola mundo** en la consola (output).

Com entarios

Idea clave

- Todo lo que está en un comentario es ignorado a la hora de ejecutar el programa.

¿Qué son los comentarios?

Los comentarios son como notas al margen en un libro de texto, usadas por los programadores para dejar explicaciones o recordatorios sobre secciones del código. Estas notas no afectan cómo funciona el programa, pero son muy útiles para cualquier persona que lea o modifique el programa en el futuro.

Comentarios en Python

En python, los comentarios inican con el símbolo "#"

```
# Este es un comentario de una línea
```

Python, a diferencia de otros lenguajes, no tiene soporte construido para comentarios de varias líneas. Sin embargo, podemos simularlo con algo que más adelante conoceremos como un *docstring*.

```
""" Este es un comentario  
de varias líneas.  
Puede tener cuantas líneas  
sean necesarias """
```

Primera Suma

Ideas clave

- Un computador es como una calculadora inteligente.
- Un programa es un documento que le indica paso a paso qué hacer.

Operaciones matemáticas

Dado que un programa le indica paso a paso qué hacer a un computador, podemos escribir un programa que le indique al computador que realice operaciones matemáticas.

Por ejemplo, si queremos realizar la operación de suma, lo haremos de la siguiente forma:

```
print(2 + 2) # Mostrará 4
```

Basta la expresión $2 + 2$ para que el computador realice la operación. Sin embargo, es necesaria la instrucción `print` para que muestre el resultado calculado.

Ejercicio

Ahora te toca a ti. Utilizando como base el ejemplo anterior, escribe un programa que muestre el resultado de la operación $3 + 3$

Variables

Ideas clave

- En un programa podemos ejecutar múltiples instrucciones.
- Usualmente escribiremos una instrucción por línea.
- Para guardar los resultados de estas instrucciones, utilizamos **variables**.

¿Qué son las variables?

Las variables son como cajas donde podemos guardar y modificar información durante la ejecución del programa.

```
resultado = 17 + 15  
print(resultado)
```

En este ejemplo, realizamos una operación de suma y guardamos el resultado en la variable `resultado`. Luego, en una segunda instrucción, le pedimos al programa que muestre el valor de la variable `resultado`, que es 32.

Ejercicio

Utilizando como base el código presentado en el contenido, escribe un programa que muestre la suma de las variables `naranjas`, `manzanas` y `peras`.

Modificando Variables

Ideas clave

- Las *variables* son como cajas donde podemos guardar información.
- La información almacenada en una variable puede modificarse al almacenar un nuevo valor en ella.

¿Por qué se llaman variables?

Las variables se llama así porque el contenido de estas cajas puede modificarse al almacenar un nuevo valor en ellas.

```
edad = 30
print(edad) # Se mostrará 30
edad = 31
print(edad) # Se mostrará 31
```

Variables vs Texto

Ideas clave

- hola es un identificador.
- "hola" es un texto.
- 2 es un número.
- Esta diferencias es muy importante en el día a día y tenemos que dominarla.

¿Cuál es la diferencia?

Existe una gran diferencia entre:

```
print("hola") # Aquí se muestra el texto hola.
print(hola) # Aquí se muestra el valor de la variable hola.
print(2) # Aquí se muestra el número 2.
```

Hola es un identificador. Los **identificadores** son nombres que identifican a una variable u otro tipo de elemento de Python que todavía no hemos cubierto.

"Hola" es un texto. Cuando un texto está entre comillas, Python lo interpreta literalmente como texto.

2 es un número. Los números los podemos diferenciar de un texto específicamente por las comillas. "2" es una cadena de texto y no un número. Las comillas pueden ser simples o dobles.

Ejercicio

Modifica la línea dentro del print para que muestre el valor de la variable a en lugar de la letra a.

```
a = "Hola Mundo"
print("a"); # Modifica esta línea
```

Reglas de identificadores

Ideas clave

- Los identificadores son nombres
- Hay distintos tipos de identificadores, pero hasta el momento solo hemos visto los identificadores de variables.
- Ejemplos de identificadores de variables son: hola, holaMundo, hola_mundo, hola23
- Ejemplos **no válidos** de identificadores de variables son: 23hola y hola-mundo

Reglas para los identificadores

- Los identificadores pueden contener letras, números y guiones bajos.
- Los identificadores no pueden empezar por un número.
- Los identificadores no pueden contener espacios.

¿Por qué es importante?

Es importante conocer las reglas para los identificadores porque, si no las seguimos, nuestro código no funcionará. Al leer el código, el intérprete de Python no sabrá qué hacer con un identificador no válido y nos mostrará un error.

En este ejercicio aprenderemos como nombrar variables para no obtener errores.

Introducción a tipos de datos

Idea clave

- El resultado de las operaciones dependen de los tipos de datos involucrados

Tipos de datos

En Python, existen diferentes **tipos de datos**. Hasta ahora, hemos utilizado dos tipos: números enteros (ints) y cadenas de texto (string). Sin embargo, hay muchos otros tipos de datos que exploraremos más adelante.

```
print(2 + 2) # 4
print("hola " + "mundo") # hola mundo
print("2" + "2") # 22
```

- En el primer caso, sumamos dos números enteros, obteniendo como resultado 4.
- En el segundo caso, concatenamos dos cadenas de texto, resultando en "hola mundo".
- En el tercer caso, también concatenamos dos cadenas de texto, lo que produce "22".

Este comportamiento también se aplica cuando operamos con variables. Por ejemplo:

```
a = "2"
b = "2"
print(a + b) # "22"
```

En Python hay tipos de datos que no se pueden mezclar. Por ejemplo, si intentamos sumar un número y una cadena de texto, obtendremos un error:

```
a = 2
b = "2"
print(a + b) # TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Por lo tanto, es importante prestar atención al tipo de datos que estamos utilizando para evitar errores.

Ejercicio

Modifica el valor asignado a la variable a para que el resultado sea el número 4.

```
a = "2"; # Modifica esta línea
b = 2;
print(a + b);
```