

Capítulo 2

Temario de Unida 2

2.1 Tu primer programa

- Tu primer programa
- La función print () - ¿cómo te habla la computadora?
- La función print () - formatear la salida

2.2 Literales de Python

- Literales - ¿qué son de todos modos?
- Literales - enteros
- Literales - flotantes
- Literales - cadenas
- Literales - valores booleanos

2.3 Operadores - herramientas de manipulación de datos

- Operadores y expresiones
- Operadores aritméticos
- Operadores y sus prioridades
- Operadores y sus vinculaciones

2.4 Variables - cajas con forma de datos

- Variables - ¿cómo nombrarlas?
- Nombres de variables vs palabras clave de Python
- ¿Cómo asignar una variable?
- ¿Cómo comentar tu código?
- Operadores de atajos

2.5 ¿Cómo hablar con la computadora?

- Salida vs entrada
- ¿Cómo ingresar datos con la función input ()?
- ¿Cómo convertir cadenas en números?
- Algunos programas interactivos simples
- Operadores de cadena
- ¿Cómo convertir números en cadenas?

2.1 Tu primer programa

La función *print()*

Una función forma parte del código de computadora y es capaz de:

- **Causar algún efecto** (por ejemplo, enviar texto a la terminal, crear un archivo, dibujar una imagen, reproducir un sonido, etc).
- **Evaluar un valor o algunos valores** (por ejemplo, la raíz cuadrada de un valor o la longitud de un texto dado); esto es lo que hace que las funciones de Python sean parientes de los conceptos matemáticos.

Funciones de Python

- Pueden provenir **del propio Python**; la función *print* es una de este tipo; tal función es un valor agregado recibido junto con Python y su entorno (está integrado, **built-in**)
- Pueden provenir de uno o más de los complementos de Python llamados **módulos**; algunos de los módulos vienen con Python, otros pueden requerir una instalación separada
- Puedes **escribirlos tú mismo**, colocando tantas funciones como quieras y necesites dentro de tu programa para hacerlo más sencillo, claro y elegante.

Funciones de Python

- ¿Qué es una **función** (incorporada, agregada por módulos, escrita por uno mismo)?
- Una función puede hacer (generar):
 - Un *efecto*
 - un *resultado*

También hay un tercer componente muy importante: el (los) **argumento** (s).

- `print ("¡Hola, mundo!")`
- `print ()`
- `print ("a")`
- `print (a)`

La función `print()`

```
print(";Hola, Mundo!")
```

- La cadena está delimitada por comillas, casi todo lo que ponga entre comillas se tomará literalmente, no como código, sino como datos.
- `print("a")`
- `print(a)`

Práctica de Laboratorio 2.1.1.7

En su primer laboratorio:

- Use la función `print()` para imprimir la línea ¡Hola, Python! a la pantalla. Utilice comillas dobles alrededor de la cadena;
- Una vez hecho esto, use la función `print()` nuevamente, pero esta vez escriba su nombre;
- Elimine las comillas dobles y ejecute su código. Observe la reacción de Python. ¿Qué tipo de error se produce?
- Luego, elimine los paréntesis, vuelva a colocar las comillas dobles y vuelva a ejecutar su código. ¿Qué tipo de error se lanza esta vez?
- Experimente tanto como pueda. Cambie las comillas dobles por comillas simples, use varias funciones `print()` en la misma línea y luego en líneas diferentes. Mira qué pasa.

La función `print` () - instrucciones

```
print("La pequeña araña subió, subió, subió.")
```

```
print("Vino la lluvia y se la llevó.")
```

La función `print ()`

Los caracteres **Escape** y Nueva Línea

```
print ("La pequeña araña \nsubió, subió, subió.")
```

```
print()
```

```
print("Vino la lluvia \ny se la llevó.")
```

caracter
escape

caracter
Nueva linea

La función `print()` – el caracter escape

Esta convención tiene dos consecuencias importantes:

- 1. Si desea poner solo una barra invertida dentro de una cadena, no olvide su naturaleza de escape; debe duplicarla, por ejemplo, una invocación de este tipo provocará un error:

• `print ("\\")` mientras que este no `print ("\\\\")`
- 2. No todos los pares de escape (la barra invertida junto con otro carácter) significan algo.

La función `print()` – usando múltiples argumentos

Puede tener más de un argumento con `print()`

- una función `print()` invocada con más de un argumento los envía todos en una línea;
- la función `print()` pone un espacio entre los argumentos generados por su propia iniciativa.

```
print ("La pequeña araña" , "subió," , "subió, subió.")
```

INCORRECTO

La pequeña arañasubió,subió, subió.

CORRECTO

La pequeña araña subió, subió, subió.

La separación con la coma de los argumentos,
agrega en forma automática un espacio

La función `print()` – los argumentos de la palabra clave

La función `print()` function tiene dos argumentos de palabra clave:

- `end`

```
print("Mi nombre es", "Python.", end=" ")  
print("Monty Python.")
```

- `sep`

```
print("Mi", "nombre", "es", "Monty", "Python.", sep="-")
```

Laboratorio 2.1.1.19

Escenario `print("Fundamentos", "Programación", "en")`
 `print("Python")`

Modifique la primera línea de código en el editor, utilizando las palabras clave `sep` y `end`, para que coincida con el resultado esperado. Recuerde, use dos funciones `print()`. No cambie nada en la segunda invocación `print()`.

Salida esperada

Fundamentos***Programación***en...Python