Curso de introducción al desarrollo web con Flask

Rodolfo Ferro
rodolfo.ferro@leon.gob.mx
LAB León

Módulo 2:

Python básico y web (pt. I)

¿Qué es Python?

Python es un lenguaje de programación interpretado, de tipado dinámico multiusos que es muy simple y elegante, pero poderoso.

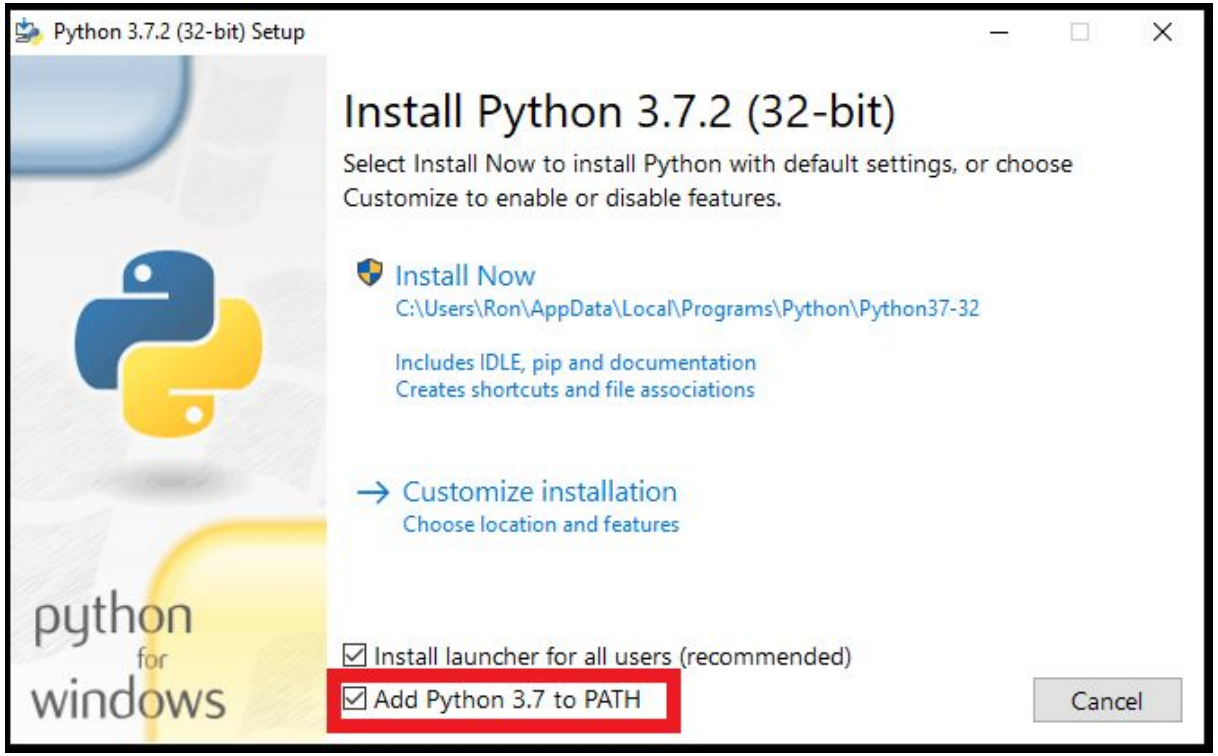
Es por ello que se utiliza en grandes compañías como Google y Dropbox, y en instituciones académicas donde lo utilizan para enseñar cursos que van desde introducción a la programación hasta inteligencia artificial. **Algunas plataformas como Pinterest e Instagram utilizan Python.**

Setup del curso

Para el desarrollo de las prácticas contenidas en el taller, estaremos trabajando sobre scripts de Python, por lo que necesitaremos instalar algunos requerimientos.

La lista de requerimientos es la siguiente:

- Python y PIP
- MongoDB
- Flask y pymongo
- Atom / Sublime Text


Python 3.7.2 (32-bit) Setup
Install Python 3.7.2 (32-bit)
Select Install Now to install Python with default settings, or choose
Customize to enable or disable features.
Install Now
C:\Users\Ron\AppData\Local\Programs\Python\Python37-32
Includes IDLE, pip and documentation
Creates shortcuts and file associations
Customize installation
Choose location and features
python
for
windows
Install launcher for all users (recommended)
Add Python 3.7 to PATH
Cancel

I/O – ¡Hola, mundo!

`input()` y `print()` son las funciones de entrada y salida (respectivamente) de información en Python. `input()` tiene como argumento una cadena de texto que da las instrucciones que se deseas desplegar y lo que retorna (la respuesta) siempre es una cadena de texto; mientras que `print()` tiene como argumentos cadenas de texto o variables que se desean desplegar.

```
# Ejemplo:
nombre = input(“Introduce tu nombre: ”)
print(f“Mucho gusto, {nombre}”)
```

Variables, tipos y conversión

Podemos pensar que una variable es como un contenedor de información. Python puede albergar distintos tipos de datos, como valores numéricos, booleanos y cadenas de texto. Al ser Python de tipado dinámico, no hace falta declarar el tipo de variable, sino que sólo basta con definir sus valores.

```
# Example:
name = “Rodolfo”
age, pi = 26, 3.1416
age = float(age)

print(name, age)
print(type(name), type(age))
```

Operaciones básicas

En esencia hay tres tipos de operaciones:

- **Operaciones aritméticas.** Operan a nivel *matemático-aritmético* sobre los valores de las variables (`+, -, *, /, //, **`).
- **Operaciones de comparación.** Operan a nivel *matemático-de-comaración* sobre los valores de las variables (`<, >, <=, >=, ==, !=`).
- **Operaciones lógicas.** Operan a nivel *matemático-lógico* sobre los valores de las variables (`and, or, not`).

Condicionales

Los condicionales los utilizaremos para validar si se cumple una o más condiciones para ejecutar cierto bloque de código, en esencia nos ayuda a tomar decisiones sobre el flujo de nuestro código.

```
# Example:
if condition:
    # Block of code
elif other_condition:
    # Another block of code
else:
    # Final block of code
```

Ciclos – `while`

Iteración significa ejecutar el mismo bloque de código una y otra vez, potencialmente muchas veces. Una estructura de programación que implementa la iteración se denomina bucle o ciclo.

Para un ciclo `while`, será importante establecer una condición de detención.

```
# Example:
while condition:
    # Block of code
```

Colecciones

Una colección es un conjunto de elementos en una estructura de datos de Python. De manera nativa Python tiene soporte para:

- Listas y tuplas
- Diccionarios y conjuntos

```
# Example:
my_list = [1.0, 2.0, 3.0]
my_tuple = (1.0, 2.0, 3.0)
my_dict = {
     “day”: 6,
     “month”: “August”,
}
```

Ciclos – `for`

Python tiene un ciclo basado en colecciones o en iteradores. Este tipo de ciclo itera sobre una colección de objetos, en lugar de especificar valores o condiciones numéricos, este ciclo se llama ciclo `for`.

```
# Example:
for element in collection:
    # Block of code that iterates over the collection
```

Funciones

La manera en la que empaquetamos procesos que ejecutan un bloque completo de código es a través de funciones, la sintaxis de Python es muy sencilla y eficiente para escribir bloques de código de forma ordenada.

La mayoría de los paquetes en Python están conformados por objetos y funciones.

```
# Example:
def function_name(arguments):
    # Block of code
    # Final block of code
    return something
```


4 BETA
Bootstrap

Siguiente sesión...

Módulo 3:

Introducción a Flask