



Facultad de Ciencias
de la **Administración**

TECNICATURA
UNIVERSITARIA EN
**DESARROLLO
WEB**



PROGRAMACIÓN I

Unidad II – Python

Ejecución de Programas

Tecnicatura Universitaria en Desarrollo Web

Facultad de Ciencias de la Administración

Universidad Nacional de Entre Ríos

- **Objetivos**

- Conocer las principales características de Python.
- Realizar una instalación completa del ambiente de desarrollo.
- Ejecutar programas simples.

- **Temas a desarrollar:**

- **Python.**
 - Características. Historia. Versiones.
- **Instalación.**
 - Instalación de Python.
 - Instalación y revisión de Entornos de desarrollo y editores de código.
- **Ejecución de programas.**
 - Sentencias simples y definición de Comentarios.

Nuestro primer programa Python

- Es tradición, que el primer programa que escribimos cuando aprendemos un nuevo lenguaje de programación se llame “**Hola Mundo!**”, porque esas son las palabras que se van a mostrar “**Hola Mundo**”.
- En **Python** se hace así:
 - » `print ('Hola Mundo!')`
- Esto es un ejemplo de como la sentencia **print** (imprimir en inglés), aunque no imprime nada en papel, muestra por pantalla los resultados. En este caso los resultados son las palabras
 - **Hola Mundo!**
- Las comillas en el programa marcan el principio y el final de un texto que debe ser mostrado pero no aparecen en el resultado.
- Los paréntesis indican que **print** es una función.
- En **Python 2**, la instrucción **print** no llevaba paréntesis:
 - » `print 'Hola Mundo!'`

Operaciones aritméticas

- Después del '**Hola Mundo!**' el siguiente paso son las matemáticas. 😞
- **Python** nos provee de **operadores**. Se tratan de símbolos especiales que representan computaciones como adición o multiplicación.
- Los operadores **+**, **-**, ***** llevan a cabo suma, resta y multiplicación. En los siguientes ejemplos los vemos en funcionamiento:

» **40 + 2**
42

» **43 - 1**
42

» **6 * 7**
42

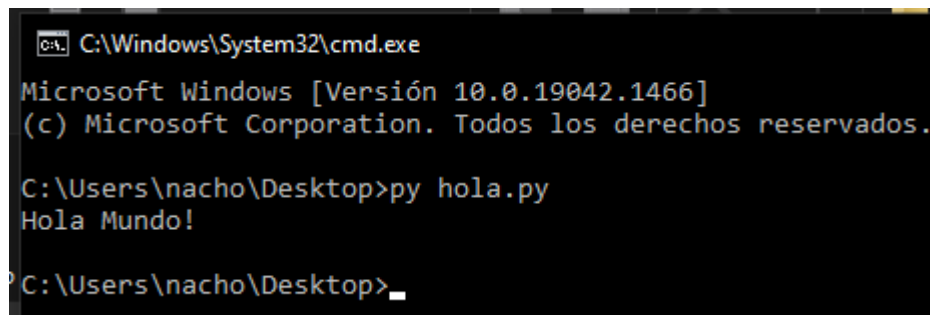
- El operador **/** realiza la división de dos números
 - » **84 / 2**
 - **42.0**
- Vemos que a diferencia de las operaciones anteriores el resultado es **42.0** en vez de **42**. Discutiremos esto más adelante.
- Finalmente, con ****** realizamos exponenciación.
 - » **6** 2 + 6**
 - **42**

Valores y tipos de datos

- **Un valor** es una de las cuestiones más básicas con las que un programa trabaja puede ser una letra o un número. Algunos valores que vimos hasta el momento son **2**, **42.0** y **'Hola Mundo'**.
- Estos valores pertenecen a distintos tipos de datos: **2** es un **entero**, **42.0** es un **número de punto flotante** y **'Hola Mundo'** es un **string o cadena de caracteres** (se llama así porque las letras están enlazadas unas a otras).
- Si no sabemos de qué tipo es un valor la función **type()** nos puede decir:
 - » **type(2)**
 - **<class 'int'>**
 - » **type(42.0)**
 - **<class 'float'>**
 - » **type('Hola Mundo!')**
 - **<class 'str'>**
- En los resultados mostrados anteriormente la palabra **“class”** (clase en inglés) se usa en un sentido de **clasificación** o **categoría**.
 - ¿Qué pasará si ponemos **2** y **42.0** entre comillas simples? Es decir **'2'** y **'42.0'**?

Ejecutando programas Python desde archivos

- Para desarrollar un programa en **Python** no basta con usar nuestros comandos en “modo calculadora”.
- Necesitamos guardar lo que queremos hacer en un archivo y ejecutar el intérprete.
- Para ello:
 - 1) Utilizando un editor de textos, creamos un nuevo archivo con el contenido de nuestro programa y luego lo guardamos con extensión **.py**.
 - 2) Abrimos una consola de comandos y nos posicionamos en el directorio donde quedó el archivo.
 - 3) Ejecutamos **Python** escribiendo en la consola: **python hola.py** o simplemente **py hola.py**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.19042.1466]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\nacho\Desktop>py hola.py
Hola Mundo!

C:\Users\nacho\Desktop>_
```

Comentarios

- En la medida que los programas se vuelven **más grandes y complejos**, se vuelven más **difíciles de leer**.
- Muchas veces vamos a encontrarnos con una porción de código que nos va a llevar tiempo darnos cuenta qué hace o el motivo por lo qué lo hace.
- Por esta razón es una buena idea agregar **notas** a los programas para explicar en lenguaje natural lo que está haciendo el programa. Estas notas son llamadas **comentarios** y comienzan con el símbolo **#**
 - **# calculo la cantidad de milisegundos en una hora**
 - **60 * 60 * 1000**
- Todo lo que aparece a partir del símbolo **#** hasta el final de la línea es ignorado por **Python** y no tiene efectos en la ejecución del programa.

Bibliografía

- Pablo A. García, Marcelo A. Haberman, Federico N. Guerrero: ***“Programación E1201: curso de grado”***. 1Era Edición. Ed. Editorial de la UNLP. 2021.
- Óscar Ramírez Jiménez: ***“Python a fondo”*** 1era Edición. Ed. Marcombo S.L.. 2021.
- Allen Downey. ***“Think Python”***. 2Da Edición. Green Tea Press. 2015.
- Eirc Matthes: ***“Python Crash Course”***. 1era Edición. Ed. No Starch Press. 2016.
- Zed A. Shaw: ***“Learn Python 3 the Hard Way”***. 1era Edición. Ed. Addison-Wesley. 2017.
- Armando E. De Giusti.: ***“Algoritmos, datos y programas con aplicaciones en Pascal, Delphi y Visual Da Vinci”***. Ed. Buenos Aires Prentice Hall; Pearson Educación. 2001.