

Análisis de datos

Nivel Básico – Explorador

Misión 1

Python

¿Qué es Python?

Python es un “*lenguaje de programación*”, pero un lenguaje con reglas muy bien definidas que deben seguirse al pie de la letra, de manera que permitan que un programador pueda darle instrucciones a una máquina con el fin de manipular información en diferentes formas y cantidades.

a. ¿Cuándo y por qué nace Python?

Python fue creado a finales de los ochenta por el programador holandés *Guido van Rossum* para su uso en las matemáticas y la informática.

b. ¿Por qué se llama Python?

Pese a lo que se pueda creer, Van Rossum no es un amante de las serpientes que estrangulan viva su comida para comérsela. En realidad, el nombre del lenguaje se puso en honor a un grupo humorista del cual era fanático este programador, llamado *Monty Python*.

¿Por qué Python?

Siendo uno de los lenguajes más usados de los últimos años, Python es una opción que se debe tener en cuenta si se busca entrar en el mundo de la programación. Además, Python cuenta con una extensa cantidad de características convenientes para el aprendizaje y el uso rápido y eficaz del mismo.

a. Sintaxis simple

Una *sintaxis* en programación es el conjunto de reglas para el uso de palabras, símbolos y orden de escritura que definen un fragmento de código como correcto. La sintaxis de Python busca ser “elegante” mediante la reducción del uso de símbolos y *palabras clave* complejos o innecesarios, que permiten así leer el código de manera casi tan fluida como si fueran oraciones en inglés.

b. Lenguaje de programación interpretado

Gracias a que Python no requiere un proceso de “traducción” extra para que la máquina lo entienda, no es necesario que el código pase por un proceso en el cual se crea otro archivo antes de ejecutarlo (los lenguajes que sí pasan por este proceso se conocen como lenguaje de programación *compilado*). En cambio, Python traduce el código por instrucciones una a una cada vez que se ejecuta (lenguajes conocidos como lenguaje de programación *interpretado*), permitiendo así ejecutar pequeños pedazos de código para hacer pruebas en ejecución rápidamente.

c. Gran comunidad y documentación

Conformado por una comunidad bastante extensa, Python permite tener una retroalimentación constante con la enorme cantidad de información que se puede encontrar al respecto en la incontable variedad de libros y páginas de internet. Por otro lado, Python es presentado con una documentación muy detallada y precisa sobre todo aquello que lo compone y aquello que se puede hacer en este lenguaje de programación, siempre acompañada por ejemplos acertados y posibles aplicaciones.

¿Cómo Instalar Python?

Python permite ser descargado e instalado en Windows de manera tan sencilla como si se tratase de una aplicación común.

a. Descarga

El primer paso es ir a la página oficial de Python, a la sección de **Descargas**.



<http://www.python.org/downloads/>

Descargue la [versión](#) que presenten al inicio (3.8.1 para el caso), pero si desea descargar alguna otra versión, más abajo en la misma sección puede encontrar todas las versiones de Python disponibles.

Tras finalizar la descarga, se tendrá un archivo llamado `python-x.x.x.exe`, en el que las x serán el número de versión (para el caso, `python-3.8.1.exe`).

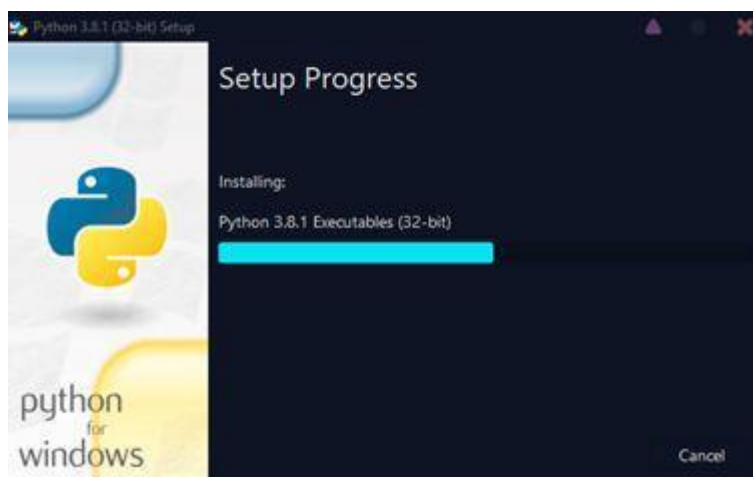
b. Instalación

Al abrir el archivo descargado previamente, el sistema preguntará si se desea ejecutar el archivo; tras aceptar, se abrirá una ventana de instalación.



IMPORTANTE: marque la opción en la parte inferior de la ventana, que dice “*Add Python 3.8 to PATH*”; esto permitirá acceder más fácilmente a los comandos de Python.

Para iniciar la instalación basta con hacer clic en **Install Now**, lo cual instalará en una dirección por defecto en la computadora.



Tras finalizar de instalar se presenta una información junto a unos enlaces con datos sobre Python; si se quiere ignorar esto, se puede cerrar la ventana.

La instalación dejará una carpeta en el **menú** de Windows con cuatro archivos: IDLE es una consola para ejecutar el *intérprete interactivo* (explicado a continuación) de Python; el archivo con el nombre de Python y la versión seguida de “(32 bits)” simplemente ejecutará el *intérprete interactivo* en la consola; los últimos dos archivos son un manual y la documentación.

¿Dónde y cómo programar?

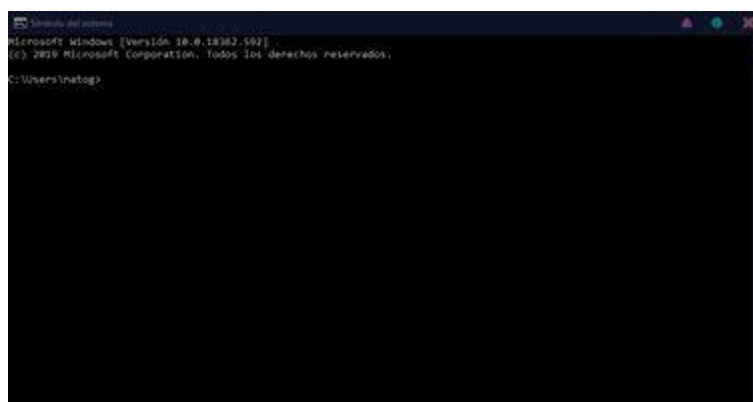
Claramente, para empezar a programar se debe saber dónde programar. Para esto existen múltiples opciones, ya que un lenguaje de programación no es más que un medio de comunicación con la máquina; se puede hablar dando instrucciones independientes una a una o escribir un conjunto de instrucciones en un *archivo fuente*.

a. Terminal de comandos

También conocido como el **símbolo del sistema** o *command prompt* en inglés, es un intérprete de comandos en Windows, es decir, una aplicación que permite comunicarse con el sistema. Con esto en mente, se puede ejecutar Python a través de este.

i. Abrir la terminal

Para abrir el símbolo del sistema en Windows, se debe ir al menú y buscar “cmd”, lo cual dará un resultado con el nombre previamente mencionado: “símbolo del sistema”. Al seleccionar este, se abrirá una ventana completamente negra con letras blancas y unas cuantas líneas de texto sobre versión y derechos de Microsoft.



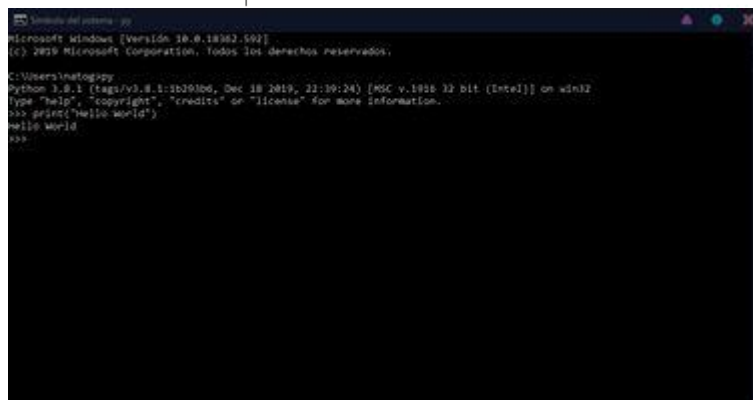
ii. Intérprete interactivo de Python

Dentro del símbolo del sistema escriba la palabra “py” y luego presione la tecla **Enter**, con lo cual se ejecutará el modo interactivo del intérprete de Python, y las líneas de comando ahora iniciarán con un trío de símbolos mayor que “>>>”; esto significa que el comando a ingresar será un comando de Python.

Una pequeña prueba puede ser escribir el siguiente comando:

```
print("Hello World")
```

Presione luego la tecla **Enter** y vea cómo reacciona el intérprete.



```

Simulador del sistema - py
Microsoft Windows [Versión 10.0.18362.592]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\matgrip>
Python 3.8.1 (tags/v3.8.1:tsb290bb, Dec 18 2019, 22:10:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello World")
Hello World
>>>
  
```

Se puede notar que el intérprete de Python reacciona imprimiendo la frase "Hello World", que es exactamente lo que se le pidió hacer con el anterior comando.

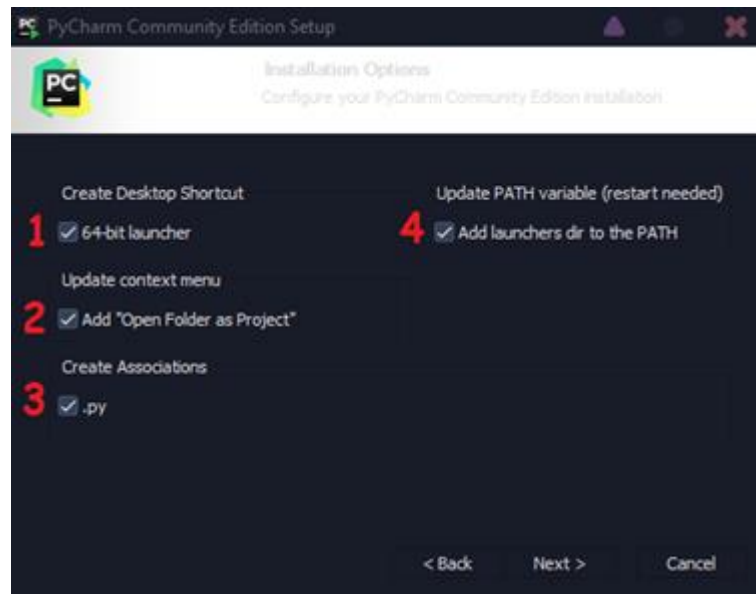
b. Entorno de desarrollo

Escribir todo un programa en la **consola** cada vez que se quiere ejecutar algo puede volverse tedioso. Así que preferiblemente se deben escribir las instrucciones en un archivo que se pueda ejecutar y modificar posteriormente. Para este trabajo es recomendable utilizar un programa destinado para esto, y ahí es donde entran en juego los IDE (Integrated Development Environment), siglas en inglés para "entorno de desarrollo integrado". Es un *software* que proporciona todos los elementos y herramientas necesarias, junto a múltiples servicios, para facilitar la creación de código y la revisión de este. Las características más destacables suelen ser un editor de texto, autocompletado, destacar errores y recomendaciones, entre otras.

i. PyCharm Community

En este curso se recomienda, y se usará, el IDE de uso gratuito PyCharm Community de *JetBrains*.

Para descargarlo se puede ir a este [enlace](#) y hacer clic en **Descargar** de la versión Community. Ejecute el archivo para iniciar la instalación, y tras aceptar las peticiones de ejecución y de administrador se abrirá la ventana de instalación. En la primera sección hay una presentación, en la segunda se escoge la dirección a instalar PyCharm (se recomienda dejar la predeterminada) y en la tercera sección se escogen opciones extras para la instalación; se recomienda marcar todas las opciones.



La primera opción simplemente creará un acceso directo en el escritorio. La segunda opción permite abrir carpetas como **Proyecto**. La tercera opción hará que todos los archivos de Python se abran por defecto con PyCharm. La cuarta opción agrega comandos de PyCharm al Símbolo del Sistema.

La siguiente sección permite escoger una carpeta del menú en el cual guardar todos los accesos directos para PyCharm (se recomienda dejar por defecto). Después de esto, se inicia la instalación y, tras terminar, se debe reiniciar el computador para completar la instalación.

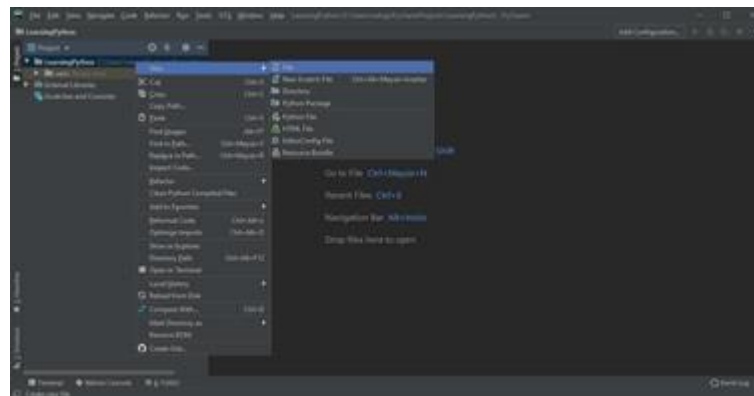
Al abrir PyCharm por primera vez se mostrará una serie de preguntas para la personalización. La primera es seleccionar un tema de colores para la aplicación (la mayoría de los programadores prefieren sus entornos oscuros). Se recomienda ignorar la siguiente sección, ya que consiste en instalar utilidades extras que pueden ser inservibles y complejas para el inicio.

c. Archivo fuente

Ya que todo está perfectamente dispuesto para empezar a programar, se procederá a crear un archivo de Python, el cual se ejecutará con fines de prueba. En PyCharm, la primera vez que es abierto se debe crear un proyecto, al cual se le puede poner cualquier nombre que se desee. Tras

crear este proyecto, se empezará a cargar todo el entorno y se crearán múltiples archivos y carpetas dentro del mismo.

Ahora, en la columna a la izquierda, se puede hacer clic derecho sobre la carpeta con el nombre que se escogió para el proyecto. Luego haga clic en **New** y en **File**.

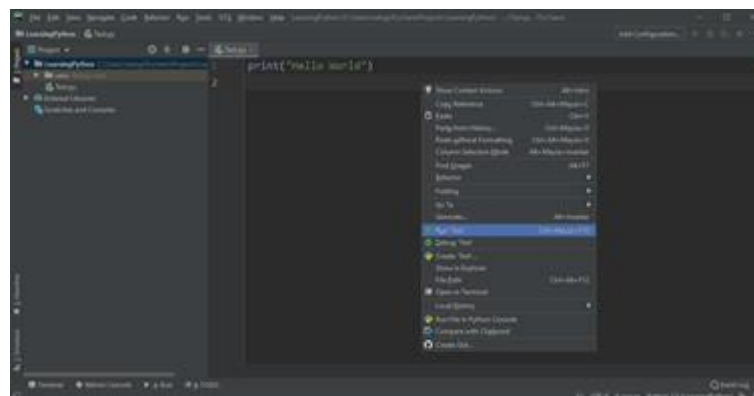


Tras esto hay que darle un nombre al archivo. Se le puede dar cualquier nombre, pero al final debe tener la *extensión* ".py", que lo distinguirá como un archivo ejecutable de Python. Por ejemplo: "Test.py".

Ahora, con un archivo de Python creado, se puede escribir código en él. Escriba algo como:

```
print("Hello World")
```

Posteriormente haga clic derecho en cualquier parte del documento y en **Run**.



Inmediatamente se abrirá una pestaña en la sección inferior de PyCharm, que es en realidad una **consola integrada** que acaba de ejecutar el código y muestra el resultado, el cual debe haber sido imprimir la frase "Hello World".

Tras esto se habrá creado y ejecutado el primer archivo en Python utilizando un IDE y unas cuantas de sus herramientas más básicas.

¿Dónde y cómo programar?

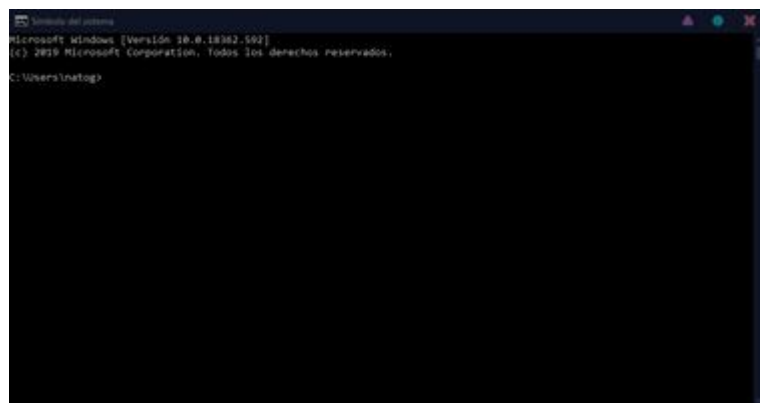
Claramente, para empezar a programar se debe saber dónde programar. Para esto existen múltiples opciones, ya que un lenguaje de programación no es más que un medio de comunicación con la máquina; se puede hablar dando instrucciones independientes una a una o escribir un conjunto de instrucciones en un *archivo fuente*.

a. Terminal de comandos

También conocido como el **símbolo del sistema** o *command prompt* en inglés, es un intérprete de comandos en Windows, es decir, una aplicación que permite comunicarse con el sistema. Con esto en mente, se puede ejecutar Python a través de este.

i. Abrir la terminal

Para abrir el símbolo del sistema en Windows, se debe ir al menú y buscar "*cmd*", lo cual dará un resultado con el nombre previamente mencionado: "símbolo del sistema". Al seleccionar este, se abrirá una ventana completamente negra con letras blancas y unas cuantas líneas de texto sobre versión y derechos de Microsoft.



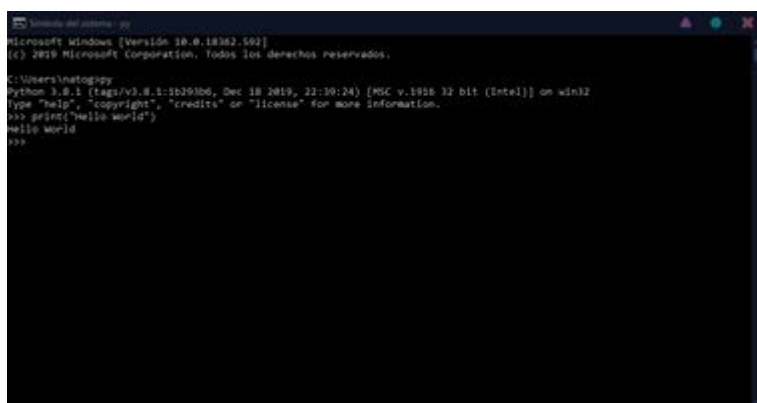
ii. Intérprete interactivo de Python

Dentro del símbolo del sistema escriba la palabra "py" y luego presione la tecla **Enter**, con lo cual se ejecutará el modo interactivo del intérprete de Python, y las líneas de comando ahora iniciarán con un trío de símbolos mayor que ">>>"; esto significa que el comando a ingresar será un comando de Python.

Una pequeña prueba puede ser escribir el siguiente comando:

```
print("Hello World")
```

Presione luego la tecla **Enter** y vea cómo reacciona el intérprete.



```

Microsoft Windows [Versión 10.0.18362.592]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\inetology>python
Python 3.8.1 (tags/v3.8.1:1b200b0, Dec 18 2019, 22:10:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello World")
Hello World
>>>
  
```

Se puede notar que el intérprete de Python reacciona imprimiendo la frase "Hello World", que es exactamente lo que se le pidió hacer con el anterior comando.

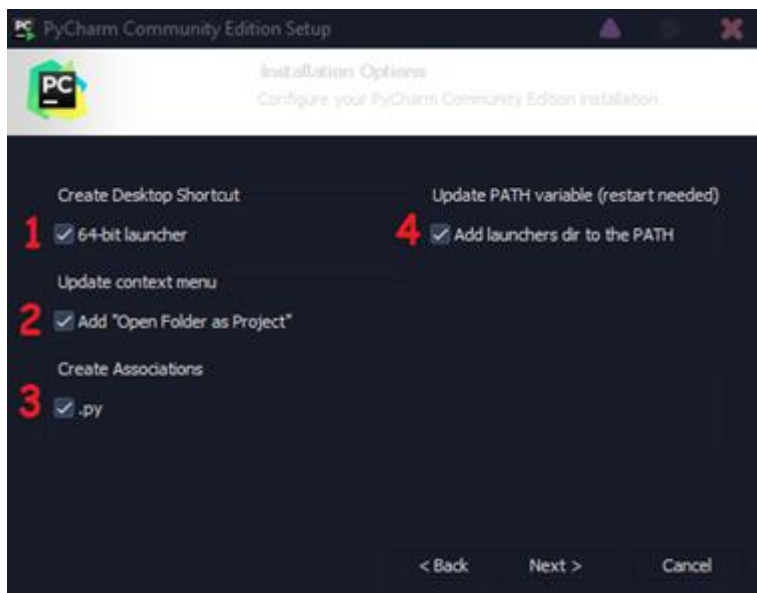
b. Entorno de desarrollo

Escribir todo un programa en la **consola** cada vez que se quiere ejecutar algo puede volverse tedioso. Así que preferiblemente se deben escribir las instrucciones en un archivo que se pueda ejecutar y modificar posteriormente. Para este trabajo es recomendable utilizar un programa destinado para esto, y ahí es donde entran en juego los IDE (Integrated Development Environment), siglas en inglés para "entorno de desarrollo integrado". Es un *software* que proporciona todos los elementos y herramientas necesarias, junto a múltiples servicios, para facilitar la creación de código y la revisión de este. Las características más destacables suelen ser un editor de texto, autocompletado, destacar errores y recomendaciones, entre otras.

i. PyCharm Community

En este curso se recomienda, y se usará, el IDE de uso gratuito PyCharm Community de *JetBrains*.

Para descargarlo se puede ir a este [enlace](#) y hacer clic en **Descargar** de la versión Community. Ejecute el archivo para iniciar la instalación, y tras aceptar las peticiones de ejecución y de administrador se abrirá la ventana de instalación. En la primera sección hay una presentación, en la segunda se escoge la dirección a instalar PyCharm (se recomienda dejar la predeterminada) y en la tercera sección se escogen opciones extras para la instalación; se recomienda marcar todas las opciones.



La primera opción simplemente creará un acceso directo en el escritorio. La segunda opción permite abrir carpetas como **Proyecto**. La tercera opción hará que todos los archivos de Python se abran por defecto con PyCharm. La cuarta opción agrega comandos de PyCharm al Símbolo del Sistema.

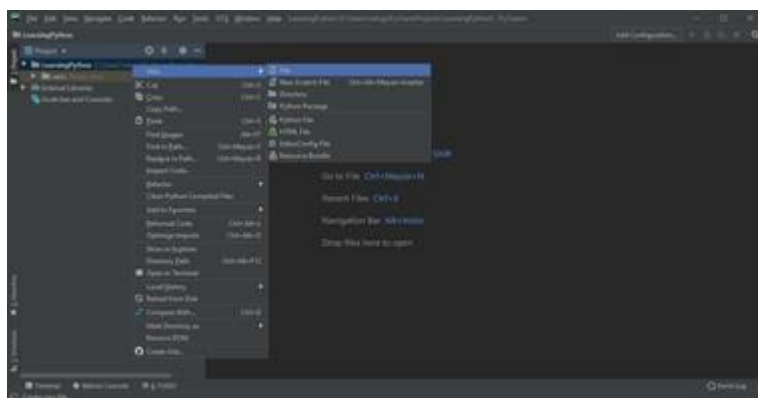
La siguiente sección permite escoger una carpeta del menú en el cual guardar todos los accesos directos para PyCharm (se recomienda dejar por defecto). Después de esto, se inicia la instalación y, tras terminar, se debe reiniciar el computador para completar la instalación.

Al abrir PyCharm por primera vez se mostrará una serie de preguntas para la personalización. La primera es seleccionar un tema de colores para la aplicación (la mayoría de los programadores prefieren sus entornos oscuros). Se recomienda ignorar la siguiente sección, ya que consiste en instalar utilidades extras que pueden ser inservibles y complejas para el inicio.

c. Archivo fuente

Ya que todo está perfectamente dispuesto para empezar a programar, se procederá a crear un archivo de Python, el cual se ejecutará con fines de prueba. En PyCharm, la primera vez que es abierto se debe crear un proyecto, al cual se le puede poner cualquier nombre que se desee. Tras crear este proyecto, se empezará a cargar todo el entorno y se crearán múltiples archivos y carpetas dentro del mismo.

Ahora, en la columna a la izquierda, se puede hacer clic derecho sobre la carpeta con el nombre que se escogió para el proyecto. Luego haga clic en **New** y en **File**.

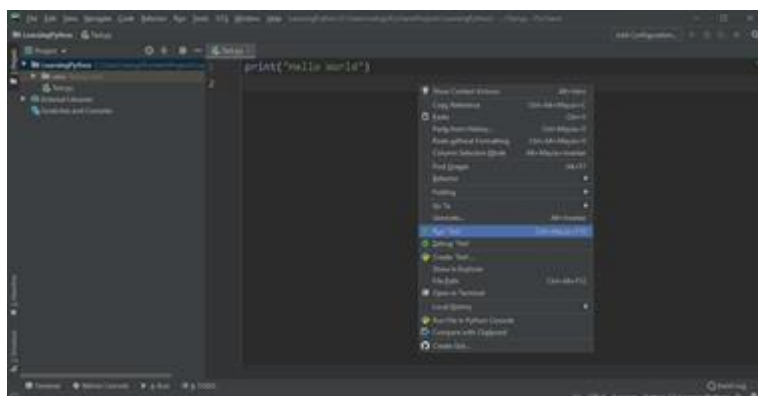


Tras esto hay que darle un nombre al archivo. Se le puede dar cualquier nombre, pero al final debe tener la *extensión* ".py", que lo distinguirá como un archivo ejecutable de Python. Por ejemplo: "Test.py".

Ahora, con un archivo de Python creado, se puede escribir código en él. Escriba algo como:

```
print("Hello World")
```

Posteriormente haga clic derecho en cualquier parte del documento y en **Run**.



Inmediatamente se abrirá una pestaña en la sección inferior de PyCharm, que es en realidad una **consola integrada** que acaba de ejecutar el código y muestra el resultado, el cual debe haber sido imprimir la frase "Hello World".

Tras esto se habrá creado y ejecutado el primer archivo en Python utilizando un IDE y unas cuantas de sus herramientas más básicas.

A tener en cuenta

Es prudente hacer unas aclaraciones antes de sumergirse dentro del aprendizaje de la programación. La mayoría son prevenciones, recomendaciones o curiosidades para tener en cuenta desde el inicio o durante todo el proceso, pero es de vital importancia exponerlas para tener un desarrollo óptimo a través del uso de lo aprendido.

a. Uso de un IDE

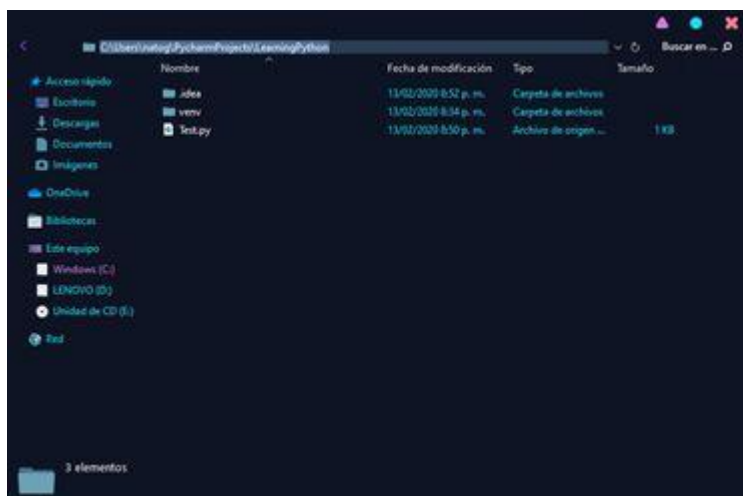
Cualquiera que haya sido el IDE de elección para el desarrollo del curso, no hay que preocuparse por lo complejo que pueda parecer este, ya que los IDE suelen ser aplicaciones muy pesadas para trabajos más profesionales. Contienen muchas opciones y herramientas, pero se recomienda empezar a aprender con estos para familiarizarse con lo que se usa en el mundo real, así que no hay que dejarse asustar por la cantidad de opciones y pestañas que pueden llegar a tener. No obstante, puede ser conveniente empezar a explorar lo que ofrecen y así moverse más ágilmente a través de estos.

b. Ejecutar archivos desde la consola

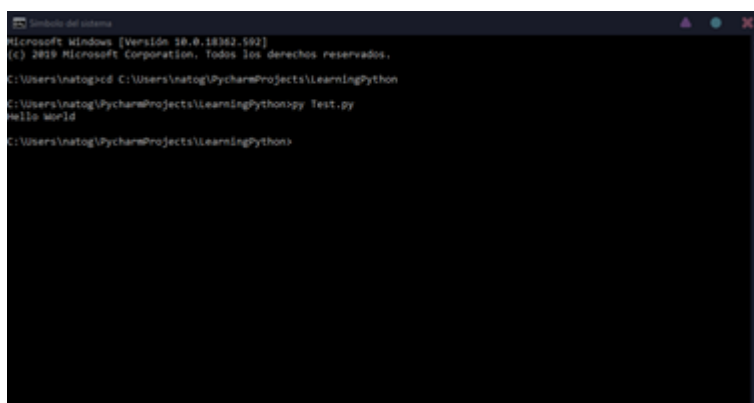
Un archivo de código realmente se puede crear desde cualquier editor de código, incluso el **Bloc de Notas**, lo cual no es recomendable, pero si se tiene un archivo fuente suelto en algún lugar y se desea ejecutar, hay que abrir la consola y navegar en los directorios hasta llegar a la posición del archivo (con fines prácticos se recomienda abrir el bloc de notas, escribir `print("Hello World")` y guardar el archivo bajo el nombre "Test.py"; luego abrir la carpeta donde está el archivo).

Para navegar en el Símbolo del Sistema de Windows se utiliza el comando "cd" seguido de un espacio y la dirección de la carpeta que

contiene el archivo (esto se puede obtener haciendo clic en la barra de navegación del **Explorador** de Windows).



Tras acceder a dicho directorio a través de la consola, se debe escribir la palabra `py`, seguida de un espacio y el nombre EXACTO del archivo (las mismas mayúsculas e incluyendo el `.py`); esto ejecutará el archivo.



Sin embargo, hay que agregar que el uso de un IDE facilita todo este proceso y permite ejecutar el archivo dentro del mismo programa.

c. Usar la documentación

En el mundo de la programación, una actividad muy importante es la búsqueda de información respecto a lo que se necesita utilizar o lo que se está utilizando; de este modo, siempre se pueden encontrar mejores maneras para hacer las cosas. Por ende, se hace indispensable el uso de la documentación que presta Python, empezando por el [PEP8](#), que es

una colección de reglas para el estilo de escritura de Python, que, aunque no sea obligatorio, permite crear código más legible y ordenado.

Finalmente, Python es instalado junto a un **manual**, el cual contiene toda la información que se pueda necesitar para aprender a programar en Python y mucho más. Este manual puede parecer muy agobiante y complejo, así que siempre se puede optar por buscar pequeñas secciones de información en internet o incluso usar la [documentación online](#), siendo así más directo y específico el trabajo de encontrar lo que se requiere.

d. Interprete interactivo vs. ejecutar archivo fuente

Un detalle importante que se debe tener en cuenta respecto al uso del intérprete interactivo es el hecho de que este imprimirá en la consola todo valor que se pueda obtener a través de un comando, independientemente de que se indique imprimir el dato o no. Por otro lado, el usar un **archivo fuente** corresponde a ejecutar un programa, por lo cual Python omite imprimir cualquier dato a menos que se indique imprimirlo.

e. Comentar código

Una de las prácticas más usuales e importantes a realizar en la programación es comentar el código. Esto se refiere a dejar fragmentos de texto que no tienen efecto alguno en el código, pero son de uso para el programador, indicando la función de alguna línea, sirviendo de recordatorio o incluso retirando una línea de código sin eliminarla.

Existen dos maneras de dejar comentarios en Python. La primera solo comentará todo lo que tenga hacia la derecha en una sola línea, y consiste en utilizar un símbolo numeral “#” seguido de lo que se desea comentar.

```
# Esto es un comentario
print("Hello World")
```

La otra manera de comentar no es realmente un formato de comentario en Python, sino que en realidad es un **tipo de dato** llamado *string*, que permite escribir texto (posteriormente se verá esto), pero si no se este en algo más, se puede utilizar como comentario. Este modo de



comentar permite hacer comentarios que abarquen múltiples líneas; lo que se desea comentar se encierra entre tres comillas.

```
""" Esto es  
otro comentario """  
print("Hello World")
```