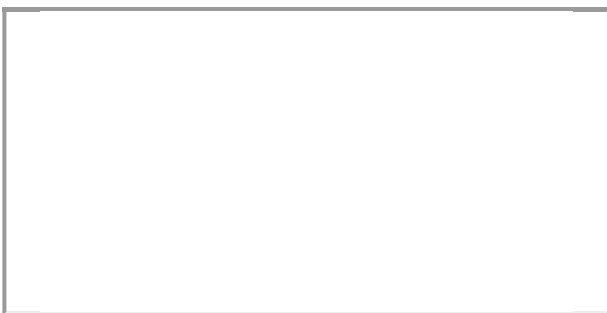


# Módulo 1. Introducción a Python

## Introducción

Esta lectura tendrá como objetivo introducir el lenguaje de programación Python, reconocer sus características principales y sus ventajas de uso. Luego de una presentación teórica en esta primera unidad sobre el lenguaje de programación, durante la siguiente unidad se trabajará en la instalación y preparación del ambiente de desarrollo. A través del contenido presentado nos aseguraremos la correcta instalación del *software*, y reconocer sus componentes principales. Esto será una tarea imprescindible para poder continuar con el resto de la materia. Este contenido proporcionará los conocimientos sobre cómo preparar el entorno de programación, desde descargar el intérprete, hasta configurar el IDE, pudiendo extenderse a cualquier otro lenguaje de programación. Esto último permitirá, por un lado, desarrollar la habilidad de preparar y configurar un ambiente de desarrollo y, por el otro, incorporar las próximas habilidades de programación, ya que sobre este ambiente se realizarán todos los ejercicios correspondientes a cada lectura.

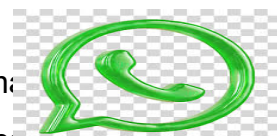
## Video de inmersión



## Unidad 1. Introducción

### Tema 1. Introducción a lenguajes de programación

Previamente a desarrollar código y convertirnos en programadores, para mañana, es necesario comenzar por los fundamentos básicos sobre un lenguaje de programación. Entendiendo este concepto será más fácil reconocer la lógica que existe detrás de cada tarea que cubre esta materia.

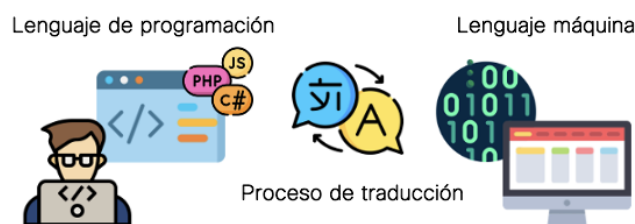


Si bien popularmente se considera a las computadoras unos dispositivos extremadamente inteligentes, la realidad es que ellas no son inteligentes por sí mismas. Ellas solamente tienen la capacidad de realizar operaciones muy sencillas, con la característica de que pueden realizarlo una gran cantidad de veces y de una forma muy rápida.

El hecho de poder resolver situaciones y operaciones complejas solo es posible a través de un **programa** que le permita a este dispositivo interpretar las operaciones a realizar y devolver un resultado o conclusión al respecto.

Existe un pequeño inconveniente: las computadoras no hablan lenguaje humano, sino *lenguaje máquina*. Internamente, estos dispositivos solo procesan unos y ceros. Por otro lado, los humanos tampoco hablamos en lenguaje máquina y realizar esto sería extremadamente complejo para poder indicarle a una computadora todas las operaciones que precisaríamos realizar. Es por esto que necesitamos un intermediario, una especie de traductor entre lo que buscamos instruirle a una computadora en *lenguaje natural*, para que la misma reciba estos comandos en *lenguaje máquina*. De aquí es que surgen los **lenguajes de programación**. Estos son lenguajes que los humanos pueden usar para desarrollar sus programas y que, al mismo tiempo, las computadoras pueden interpretar para ejecutar las instrucciones. Existen cientos de lenguajes de programación. Algunos nombres de aquellos más utilizados hoy en día son Java, C, C++, C#, Python, PHP, SQL y Javascript.

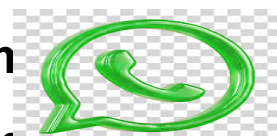
**Figura 1. Lenguaje de programación**



Fuente: Elaboración propia.

## 1. Preguntas de repaso: Verdadero o Falso.

Un lenguaje de programación permite el proceso de traducción de lenguaje máquina a lenguaje



natural para que una computadora pueda interpretar instrucciones.

Verdadero.

Falso.

Justificación

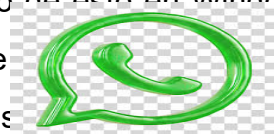
## Tema 2. Lenguaje interpretado vs. lenguaje compilado

Anteriormente se mencionó que el lenguaje de programación cumple un rol de traductor entre las instrucciones deseadas por el programador y el lenguaje máquina que logran interpretar las computadoras para ejecutar las operaciones. En esta sección se comentarán las dos maneras que existen de realizar este proceso de traducción: *interpretación* y *compilación*. Si bien ambas realizan el **proceso de traducción**, cabe destacar que son dos conceptos distintos y no trabajan en conjunto, sino que será una o la otra dependiendo del lenguaje de programación con que se trabaje.

### Proceso de traducción

#### Compilación

Un **lenguaje de programación compilado** pasa por un proceso de –como su nombre lo indica– compilación, a través del cual el *software*, luego de ser analizado en busca de errores en la sintaxis, se empaqueta en un archivo (un ejemplo de esto en Windows son los archivos del tipo .exe). De esta manera, el código fuente se convierte en un archivo de lenguaje de alto nivel al lenguaje máquina y queda listo para ser ejecutado. El hecho de contar con un archivo ejecutable le da portabilidad, pudiendo distribuir el programa sin necesidad de que el receptor cuente con acceso al código. Algunos



ejemplos de lenguajes de programación compilados son C, C++ y Java.

## Interpretación

El **lenguaje de programación interpretado**, a diferencia del compilado, es traducido al momento de ejecución y no hay un archivo ejecutable resultante de este. Una diferencia clara al momento de ejecutar un programa en lenguaje interpretado, es que, en caso de haber un error en el código, se observa que este se ejecuta hasta la línea que presenta el error. Esto no sucede con el lenguaje compilado, que –funcione o no– no permite ejecutar ninguna pieza del código si hay una parte errónea.

## Proceso de traducción

Para poder realizar este **proceso de traducción de un lenguaje** de programación interpretado, es necesario un programa denominado *Intérprete*. A su vez, esto significa que el código fuente no es sencillo de distribuir, pues el usuario que lo consuma también debe contar con un intérprete para poder ejecutarlo.

Algunos ejemplos de lenguajes de programación interpretados son Ruby, *Javascript* y Python.

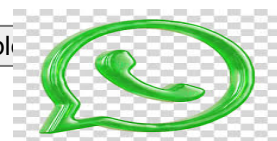
De todos los lenguajes mencionados, en esta materia se hará foco en este último.

## 2. Actividad de repaso: Seleccioná la opción correcta

### En el lenguaje interpretado...

1. El código fuente se traduce cada vez que el archivo es ejecutado a través de un *software* específico.

2. El código fuente se traduce por única vez generando un archivo ejecutable.



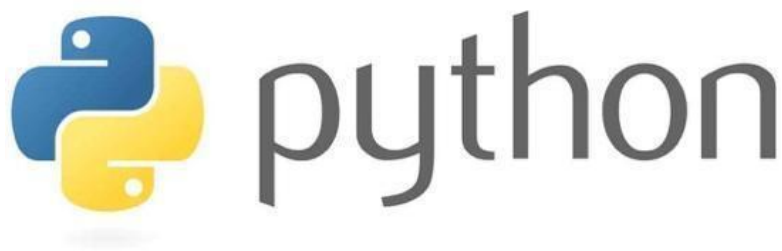
Justificación

## Tema 3. Qué es Python. Historia. Versiones

Habiendo mencionado varios lenguajes de programación, tanto compilados como interpretados, ¿por qué Python? Y, al mismo tiempo, ¿qué es Python?

Se intentará responder a estas preguntas a continuación.

**Figura 2: Logo de Python**

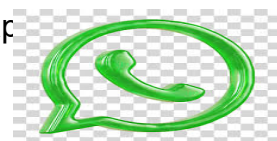


Fuente: Python Software (s. f.). Logo de Python [Logo]. Python Software Foundation.

Muchas personas afirman que este lenguaje de programación interpretado es el más usado en el mundo, otros que se encuentra en el tercer puesto. Lo que sí se puede afirmar es que sin dudas es uno de los lenguajes de programación más populares del momento.

Una particularidad de este lenguaje es que fue creado por una sola persona, Guido van Rossum (quien publicó su primera versión en 1994) y su adopción fue creciendo exponencialmente con el pasar de los años.

Los objetivos que Guido perseguía detrás de este proyecto era desarrollar un lenguaje fácil e intuitivo de utilizar, de manera que sea comprensible como el idioma inglés. Adicionalmente, su idea era que fuera de código abierto, de manera que la comunidad de programadores pueda aportar a su desarrollo. Por último, también buscaba que fuera versátil, es decir, que el tiempo de desarrollo fuera relativamente corto, para que se pudieran realizar tareas cotidianas en este lenguaje de forma rápida.



### **Versiones de Python**

Si bien la primera versión pública de Python fue lanzada en el año 1994, a medida que la comunidad fue creciendo, posteriormente fueron surgiendo nuevas versiones que

incorporaron significativas mejoras y arreglaron errores que se fueron descubriendo con el tiempo. Sin entrar en demasiadas especificaciones técnicas de cada versión, vale la pena destacar que en el año 2000 es publicada la versión 2.0, y una última gran actualización del lenguaje fue la versión 3.0, publicada en el año 2008.

Si bien la versión 2.0 llegó a su fin de vida en el año 2020 (los desarrolladores ya no siguen implementando mejoras sobre la misma), al día de hoy se continúa usando ya que existe una gran cantidad de aplicaciones en circulación desarrolladas en esa versión. No obstante, Python 3.0 incorpora grandes cambios en la sintaxis y esto hace que las versiones anteriores no sean retrocompatibles. Esto significa que no necesariamente un *script* escrito en Python 2 funcionará con un intérprete de Python 3, lo cual implica que para reutilizar un *script* de la versión anterior deberá hacerse un ajuste manual de aquellas líneas de código que presenten una sintaxis diferente.

Si se va a comenzar un nuevo proyecto de Python, se recomienda utilizar Python 3. Al mismo tiempo, esta es la versión que será utilizada durante esta materia.

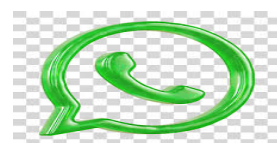
### 3. Pregunta de repaso: Verdadero o falso

**Al comenzar un nuevo programa en Python, es recomendable utilizar la versión 2, dado que es la que más tiempo lleva en el mercado y es la más estable.**

Verdadero.

Falso.

Justificación



## Tema 4. Ventajas y casos de uso

Muchas de las ventajas de este lenguaje de programación están asociadas con los objetivos originales de su creador, que sin dudas fueron cumplidos: desarrollar un lenguaje fácil e intuitivo, de código abierto y rápido de escribir.

Adicionalmente, entre otros beneficios que favorecieron su adopción se puede mencionar los siguientes.

### ≡ Es fácil de aprender

Es un muy buen lenguaje para comenzar en el camino de la programación, ya que su sintaxis permite enfocarse principalmente en la lógica y en lo que se desea hacer sin demasiadas complicaciones.

### ≡ Es fácil de entender

El hecho de que su sintaxis esté directamente relacionada con el idioma inglés permite leer el código de una manera muy simple.

### ≡ Es fácil de utilizar

A diferencia de otros lenguajes en los que hay que desarrollar muchas funciones (se explicará este concepto posteriormente) para realizar una determinada acción, Python cuenta con muchas librerías propias que acortan significativamente el tiempo de desarrollo.

### ≡ Es fácil de obtener e instalar

Python puede ser instalado de manera gratuita en múltiples plataformas.

Adicionalmente a las ventajas que este lenguaje presenta, encontramos distintos casos de uso:

- **Networking:** en los últimos años, la necesidad de automatizar configuraciones en distintos dispositivos que conforman las redes de telecomunicaciones hizo que Python se vuelva un lenguaje popular dentro de las opciones gracias a sus librerías que permiten interactuar con dispositivos de configuración a gran escala.
- **Data Science y Machine Learning:** sus librerías matemáticas, estadísticas y gráficas hicieron que Python sea uno de los lenguajes preferidos para trabajar con



análisis de datos y su procesamiento.

- Para realizar llamados API e interactuar con diversas plataformas.
- Para automatización de tareas y procesos repetitivos.

Los casos de uso de Python fueron evolucionando a medida que las necesidades del entorno fueron cambiando y a medida que el desarrollo de este lenguaje y sus distintos módulos lo fue permitiendo. Por ejemplo, en sus primeros años era utilizado principalmente para automatización de tareas y procesos repetitivos. Con el pasar de los años, el aporte de la comunidad y el auge de la Inteligencia Artificial y el análisis de datos, su uso se fue expandiendo también hacia este campo, siendo hoy en día una de las herramientas principales utilizadas para análisis y gráficos de *Big Data*.

#### 4. Actividad de lectura: *Multiple choice*

### ¿Cuáles son los casos de uso principales de Python?

- Data Science

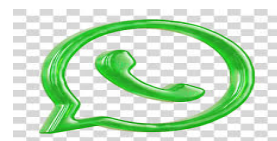
- Automatización de procesos repetitivos

- Desarrollo de páginas web

- Interactuar con APIs

- Manejo de bases de datos

Justificación



## Unidad 2. Preparación de ambiente



Luego de haber presentado el lenguaje de programación con el que trabajaremos durante toda la materia en la unidad anterior y haber comentado sus características principales, el propósito de esta unidad será guiarlos para su correcta instalación y para preparar el ambiente de desarrollo en el que se trabajará a lo largo de la materia.

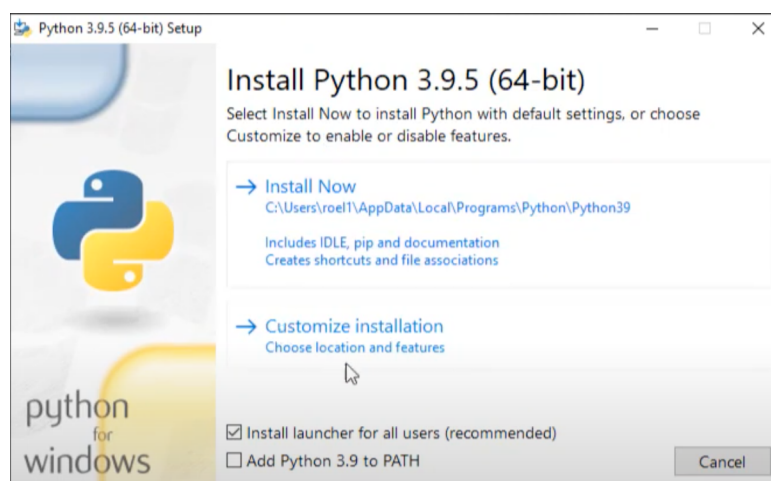
## Tema 1. Descarga e instalación de Python

El primer paso a seguir será la descarga del intérprete de Python, que permitirá a la computadora poder traducir el código a desarrollar en Python a lenguaje máquina. Para esto, nos dirigiremos al sitio oficial <https://www.python.org/downloads/>

Una vez que nos encontramos en este sitio, se debe seleccionar el correcto sistema operativo y la versión del *software*. Se recomienda descargar la última disponible.

Una vez ejecutado el archivo descargado, se dará inicio al instalador de Python que tendrá una vista similar a la siguiente:

**Figura 3: Captura de pantalla**



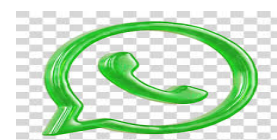
Fuente: Captura de pantalla de *software* de instalación.

Simplemente seleccionar “Install Now” y seguir los pasos de instalación.

Finalmente, para verificar que la versión descargada haya quedado debidamente instalada, abrir la terminal de comandos (en Windows se debe buscar el programa llamado “cmd”) y escribir el siguiente comando, seguido de Enter:

```
PC C:\Users\Demo> python --version
```

```
Python 3.9.5
```



En caso de recibir un mensaje como el ejemplo anterior indicando la versión de Python

instalada, la instalación se completó correctamente. Caso contrario, se deberá repetir el proceso de instalación anterior.

## 5. Actividad de lectura: Verdadero o falso

**Antes de poder utilizar el lenguaje de programación Python, es necesario descargar e instalar su intérprete desde su página oficial.**

Verdadero.

Falso.

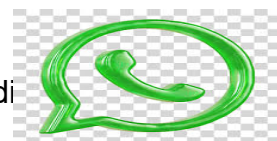
Justificación

## Tema 2. Introducción a IDE (*Visual Studio Code*)

Si bien en el paso anterior se descargó el intérprete y con eso sería suficiente para poder comenzar a desarrollar algún *script* en Python, en el día a día escribir desde el intérprete nativo no es práctico. Esto se debe a que este consta simplemente de un cuadro de texto en el cual se escribe código y no provee ningún tipo de ayuda ni herramientas que faciliten el trabajo al programador.

Es por eso que para trabajar de manera cotidiana desarrollando código, se utiliza un *software* denominado IDE (Integrated Development Environment), o *entorno de desarrollo integrado* por sus siglas en inglés.

Este consta de una interfaz que provee al desarrollador de diversas herramientas muy útiles, enumeradas a continuación.



- Coloreado de palabras clave para distinguir distintas propiedades.

- Detección de comandos y librerías que permiten el autocompletado de la línea, sugiriendo las distintas opciones disponibles. Esto acelera mucho el tiempo de desarrollo.
- Trabajo en simultáneo, permitiendo tener varios *scripts* abiertos e ir cambiando de ventana a ventana.
- Solución de problemas, resaltando en el código alguna línea que contenga un error de sintaxis e indicando una posible resolución.

Este es un ejemplo de cómo se ve un fragmento de código Python en un IDE.

**Figura 4: Captura de pantalla**

```

39  #set this to True if you use a HTTP Proxy. Add the proxy url
40  use_proxy = False
41
42  http_proxy = mytokens.http_proxy
43  proxyDict = {
44      "http" : http_proxy
45  }
46
47  # Function for posting the message to the collaboration client
48  def post(message):
49
50      for key, value in clients.items():
51          clients[key]["data"]["text"] = message

```

Fuente: Captura de pantalla de Python.

## 6. Actividad de lectura: *Multiple choice*

**¿Qué herramientas provee un intérprete para facilitar y hacer más rápido el proceso de programación?**

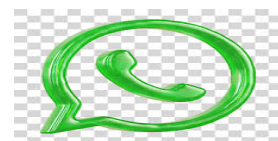
- Corrección de errores de sintaxis

- Sugerencia de bloques de código posibles

- Coloreado de palabras clave

- Detección de comandos y librerías

- Autorresolución de problemas



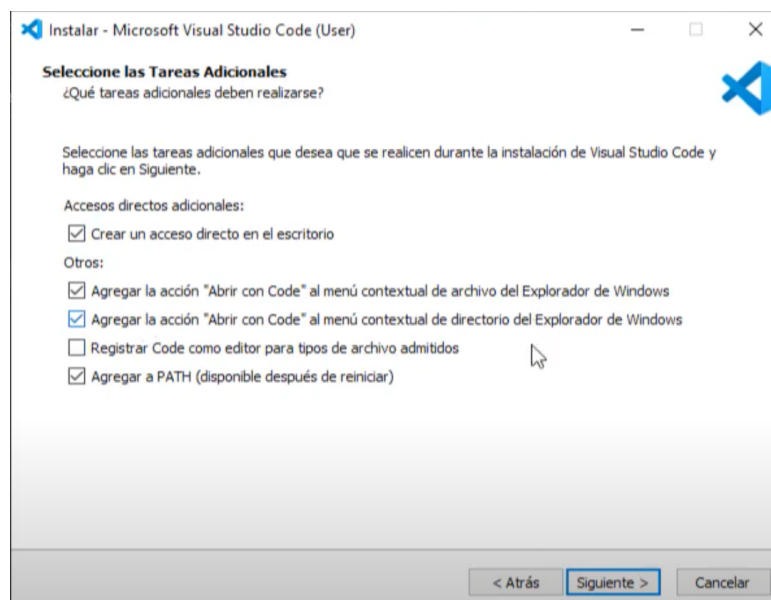
### Tema 3. Descarga e instalación de *Visual Studio Code*

Existen diversos IDEs compatibles con Python, entre ellos PyCharm, Atom y Visual Studio Code. Cualquiera de ellos permitirá realizar los ejercicios cubiertos en esta materia. Simplemente con el objetivo de poder explicar los pasos a realizar con los ejercicios solamente en uno de ellos, el IDE elegido para trabajar es el Visual Studio Code. A continuación, se realizará una guía para su instalación y un recorrido por sus componentes principales.

Para comenzar, se debe descargar el archivo instalador desde su sitio web oficial a través del siguiente *link*: <https://code.visualstudio.com/download>

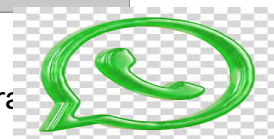
Al ejecutarlo, seguir los pasos indicados por el instalador. Al llegar a la siguiente ventana, seleccionar las opciones que se muestran en la imagen:

**Figura 5: Captura de pantalla de instalador**



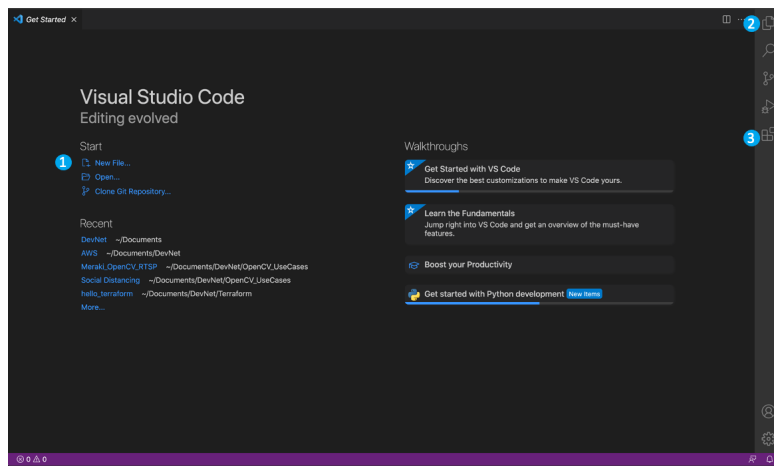
Fuente: Captura de pantalla de instalador de *software*.

Una vez finalizada la instalación, al abrir el programa se encuentra



ntalla:

**Figura 6: Captura de pantalla**



Fuente: Adaptación de captura de pantalla de Visual Studio Code.

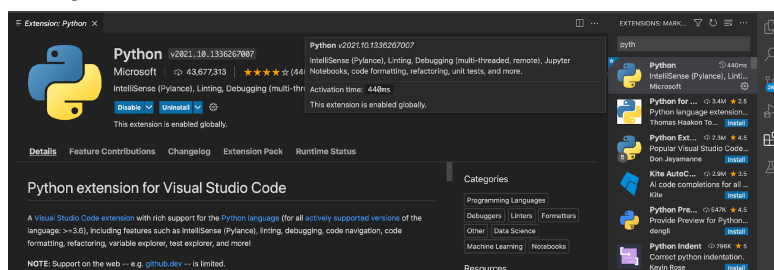
Cabe resaltar algunos puntos que se observan señalados con números en la imagen.

1. En el apartado de Inicio, se encuentran las opciones de crear un nuevo archivo, abrir un archivo o directorio existente, y el de clonar un repositorio de Git.
2. El Explorador permitirá, una vez creado un archivo o directorio, poder navegar entre directorios o distintos archivos dentro de un mismo directorio, por ejemplo, haciendo referencia a un mismo proyecto.
3. El apartado de Extensiones permitirá instalar complementos al *software* que, entre otras cosas, permitirá añadir funcionalidades adicionales al IDE.

Habiendo mencionado al Explorador, antes de comenzar con el primer programa en Python, se debe instalar la extensión de Python que añade diversas funcionalidades de gran utilidad a la hora de programar en este lenguaje.

Para realizar esto, se debe hacer *click* en el icono de extensiones, buscar “Python” e instalar la extensión.

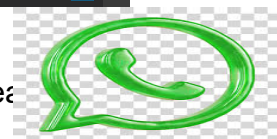
**Figura 7: Captura de pantalla**



Fuente: Captura de pantalla de Visual Studio Code.

Habiendo completado este paso, ya estamos listos para crear un programa en Python.

## 7. Actividad de lectura: *Multiple choice*



## ¿Cuáles de las siguientes opciones corresponden a distintos nombres de IDE disponibles?

a) PyCharm

b) Visual Studio Code

c) Git

d) Python 3.9


Justificación

### Tema 4. Primer programa

Con el objetivo de tener el primer acercamiento al lenguaje de programación Python, así como también al funcionamiento del IDE Visual Studio Code, se procederá a desarrollar el primer programa.

Este ejemplo se trata del programa “Hola Mundo” y consiste en mostrar en pantalla un texto conteniendo esas palabras. Este programa es popularmente conocido por ser el primero que se realiza al estar aprendiendo un lenguaje de programación nuevo.

Para comenzar, se debe crear un nuevo archivo. Para esto, en la pantalla inicial se puede seleccionar la opción “New file”, o bien seleccionar en el menú “File -> New file”.

Una vez creado un directorio donde se guardarán los archivos nuevo archivo con el nombre “holamundo.py”. Notar qu guardar el gramas desarrollados en Python deben terminar con la extensión “.py”.

Habiendo abierto el editor de texto, es momento de escribir la línea de código que

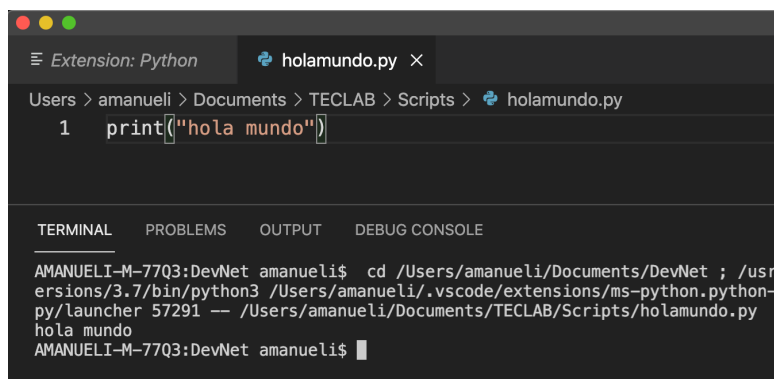
realizará la operación deseada:

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

Para ejecutar esta acción, se debe seleccionar la opción del menú “Run -> Run without debugging”.

Posteriormente, notarán que la terminal mostrará unas líneas de texto, y entre ellas el resultado de nuestra operación.

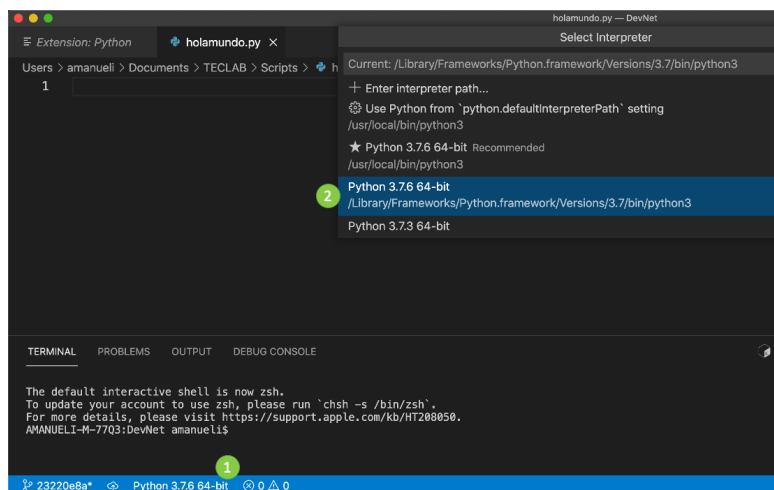
**Figura 8: Captura de pantalla**



Fuente: Captura de pantalla de Visual Studio Code.

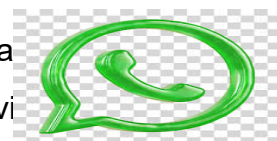
En caso de que arroje un mensaje de error, asegurarse de que la versión correcta del lenguaje se encuentre seleccionada haciendo *click* en (1), y posteriormente seleccionando la última versión instalada en (2).

**Figura 9: Captura de pantalla**



Fuente: Adaptación de captura de pantalla de Visual Studio Code.

La función que permite esta operación es *print()*. Se trata de una Python cuyo objetivo es mostrar en la terminal aquello que sea provisto (los parámetros que se especifican entre paréntesis luego de la palabra *print*). Es de gran utilidad principalmente para entregar información al desarrollador, ya sea de resultados de operaciones que realice el código, o de textos informativos durante el tiempo de ejecución



del programa.

Se cubrirán las funciones internas de Python en mayor detalle en un módulo posterior.

Habiendo completado todos los pasos hasta este punto, ya se cuenta con un ambiente listo para poder avanzar con los ejercicios de los módulos posteriores, habiendo incluso ejecutado el primer programa en Python.

## 8. Actividad de lectura: Actividad de opción correcta

**¿Qué extensión debe tener un programa escrito en el lenguaje de programación Python?**

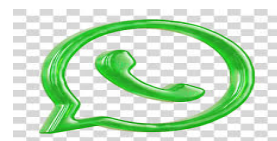
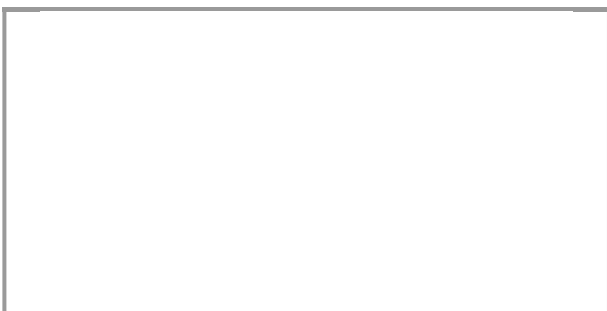
- .python

- .code

- .py

- .ide

### Video de habilidades



### Pregunta de habilidades



**1. ¿Cuál es el nombre del IDE que se utiliza en el video?**

a. PyCharm

b. Python 3.0

c. Visual Studio Code

d. Intérprete

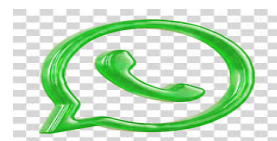
Justificación

**2. Para imprimir un mensaje en pantalla, se debe utilizar la función integrada show(),**

Verdadero.

Falso.

Justificación



**3. La extensión de un archivo de Python es “.p”,**

Verdadero.

Falso.

Justificación

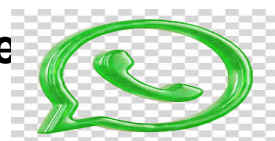
**4. Para imprimir un texto en pantalla, es posible especificar dentro de la función print() directamente el texto, o bien la variable conteniendo el mismo.**

Verdadero.

Falso.

Justificación

**5. ¿A través de qué símbolo se puede crear un comentario en un script de Python?**



a. &

b. #

c. %

d. @

Justificación

## Cierre

El aprendizaje de estos conceptos nos permite entender la diferencia entre lenguaje compilado e interpretado y reconocer a Python dentro de esta última categoría, sabiendo definir sus características. Se guio al alumno para la preparación y configuración del ambiente de desarrollo y esto es imprescindible para los módulos siguientes donde se incorporará la habilidad práctica y todos los *scripts* serán testeados sobre este entorno recientemente instalado.

## Glosario



## Referencias

Python Software (s. f.). Logo de Python [Logo]. Python Software

