

# Introducción a Python



ESET Latinoamérica | [info@eset-la.com](mailto:info@eset-la.com) | [www.eset-la.com](http://www.eset-la.com)



## Contenido

Contenido.....	2
Un poco de Historia.....	3
Scripting y más.....	3
Multiplataforma.....	3
Variables y Tipado.....	4
Paradigmas en Python.....	4
¿Por qué Python?.....	4
Instalando Python.....	4
Python en Windows.....	5
Cómo armar un entorno de desarrollo en Python.....	6
En Windows por ahí.....	7
Interactuando con Python.....	7
La consola.....	7
Los Scripts.....	7
„Hola Mundo“ en Python.....	9
Conclusión.....	9

# Introducción

Aprender a programar no es una tarea fácil, pero si divertida. Como les comentamos anteriormente, el presente curso es una introducción a un lenguaje de programación muy dinámico, práctico y fácil de aprender, **Python**. Este lenguaje de programación se creó a principios de los años noventa y desde entonces ha sido utilizado para un gran número de proyectos incluyendo programación Web, herramientas de seguridad, para scripting, automatización de tareas y demás. Desde el Laboratorio de ESET Latinoamérica buscamos que entusiastas y estudiantes puedan incorporar nuevos conocimientos que los ayuden en su desarrollo profesional.

## Un poco de Historia

Python es un lenguaje de programación que se creó a principios de los años noventa por **Guido Van Rossum**, entre las cualidades más particulares del lenguaje se destaca que cuenta con una sintaxis muy limpia, es potente, dinámico fácil de aprender, en parte por eso lo elegimos para este curso. A través de los años, Python se convirtió en un lenguaje muy adoptado por la industria de la seguridad informática, debido a su simpleza, practicidad y que es tanto un lenguaje interpretado como de scripting.

En la actualidad, podemos encontrar este lenguaje de programación en una gran cantidad de herramientas que se usan a diario dentro del mundo de la seguridad como así también en sistemas basados en UNIX por defecto. Como se pueden crear scripts para automatización de tareas Python puede convertirse en una herramienta muy práctica para evitar tener que realizar tareas repetitivas como configuración de sistemas, análisis básico de malware y otra gran lista de tareas que iremos viendo a medida que avancemos con el curso.

Para todos aquellos curiosos que tengan alguna duda en relación al lenguaje o las capacidades que tiene, los invitamos a buscar en Internet la gran cantidad de recursos que existen para aprender acerca de este lenguaje. Además, la extensa bibliografía disponible acerca de Python complementará todo lo que veamos a lo largo de los siguientes módulos.

Entonces, **¿qué es Python?** Python es un lenguaje de programación, *open source*, optimizado para crear software de calidad asegurando la productividad de los desarrolladores como así también la portabilidad a diferentes plataformas. Su éxito a través de los años ha quedado demostrado no solo por su aplicación en el mundo de la seguridad sino que además está presente en productos de Google, NASA, YouTube y muchas otras empresas más.

## Scripting y más

Como vimos, Python es también un lenguaje de Scripting, esto significa que podemos crear nuestros propios *scripts* y ejecutarlos sin la necesidad de compilar el código, como sucede con otros lenguajes como Java, C++, o C# entre otros. Por otro lado, para contar con esta capacidad, Python relega un poco de velocidad su ejecución en pos de garantizar portabilidad y flexibilidad.

El código que nosotros creamos en Python, se traduce en un *bytecode* al momento de ejecutarse por primera vez y es lo que veremos más adelante como los archivos *.pyc* o *.pyo*. Por eso en los sistemas en los cuales vamos a ejecutar nuestros programas o scripts desarrollados en Python necesitaríamos que este se encuentre instalado.

## Multiplataforma

Python cuenta con un intérprete, y por lo tanto se encuentra disponible en un gran número de plataforma incluyendo diferentes sistemas operativos como Windows, UNIX, Mac OS, Linux y además de poder encontrarlo en computadoras y Laptops existe la posibilidad de ejecutarlo en dispositivos móviles.

Los programas o scripts de Python se van a poder ejecutar en cualquier plataforma a excepción de que se utilice alguna librería específica. Además funcionalidades de librerías que vienen por defecto en Python como el caso de **os** y **sys**, brindan herramientas al desarrollador o especialista de seguridad para maximizar la eficiencia de su código. Hablaremos sobre ambos módulos más adelante en otro de capítulo de este curso.

## Variables y Tipado

Si bien todavía no definimos qué es una variable, o para qué las vamos a utilizar es importante definir que en Python cuenta con tipado dinámico. Con esto hacemos referencia a que cada variable que vayamos usando en nuestros *scripts* no es necesario definir un tipo de dato (número, texto, etc). En Python el tipo de una variable se define en **tiempo de ejecución**, y según el tipo de valor que se asigne.

Otro concepto que es importante definir que es Python, además de contar con tipado dinámico, es **fuertemente tipado**. ¿Qué significa esto? En pocas palabras que cuando una variable es de un determinado tipo no podemos tratarla como si fuera otra cosa. ¿Se marearon? Bueno, existen diferentes tipos de datos en los lenguajes de programación y si bien vamos a tratar este tema en detalle en el módulo 2, por ahora vean como que los números se tratan de una manera y las letras de otra. Por lo tanto, no podemos sumar letras ni juntar números para formar palabras.

## Paradigmas en Python

En programación, existen diferentes maneras de ver las cosas, y cada una de estas perspectivas es también conocidas como **paradigmas de programación**<sup>1</sup>. Cada paradigma de programación cuenta con una serie de metodologías o maneras de pensar que llevan a un camino diferente de resolver un mismo problema según el paradigma<sup>2</sup>. En los últimos años uno de los paradigmas más utilizados es el **Orientado a Objetos**<sup>3</sup>. La programación orientada a objetos busca representar mediante clases y objetos las interacciones necesarias para llevar a cabo una determinada tarea.

A lo largo del curso, iremos explicando más en detalle este paradigma y nos dedicaremos en el Módulo III de lleno a explicar en detalle cómo funcionan las clases y los objetos en Python.

## ¿Por qué Python?

Si bien en los párrafos anteriores enumeramos varios de los motivos por los cuales elegimos Python, como así también algunas de las características que forman parte del lenguaje es importante destacar que es un lenguaje fácil de leer. La sintaxis es sencilla, clara y fácil, por lo tanto a medida que pase el tiempo se hará más fácil leer código que hayan escrito ustedes o que busquen en Internet.

Además, Python cuenta con un montón de librerías para tareas tales como análisis del tráfico de red, ingeniería inversa, **pentesting**, análisis de vulnerabilidades, **análisis de malware** y desarrollo de aplicaciones entre otros.

## Instalando Python

Cómo ya lo comentamos varias veces, en tanto en las últimas versiones de OS X de Apple como en las distribuciones de UNIX y GNU Linux Python viene pre instalado por defecto. En particular, y en el caso de distribuciones como Ubuntu y otras, la versión que encontrarán por defecto es la 2.7. Para confirmar que Python se encuentra instalado en su sistema basta con abrir una consola y escribir la palabra “python” y observarán que la ventana es similar a la siguiente:

<sup>1</sup> Paradigmas de programación: [http://es.wikipedia.org/wiki/Paradigma\\_de\\_programaci%C3%B3n](http://es.wikipedia.org/wiki/Paradigma_de_programaci%C3%B3n)

<sup>2</sup> Programming Paradigm: [http://en.wikipedia.org/wiki/Programming\\_paradigm](http://en.wikipedia.org/wiki/Programming_paradigm)

<sup>3</sup> Programación Orientada a Objetos: [http://es.wikipedia.org/wiki/Programaci%C3%B3n\\_orientada\\_a\\_objetos](http://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos)

```
eset@Supernova: ~  
eset@Supernova:~$ python  
Python 2.7.3 (default, Sep 26 2013, 20:08:41)  
[GCC 4.6.3] on linux2  
Type "help", "copyright", "credits" or "license()" for more information.  
>>> 
```

Imagen 1 - Python instalado

En la imagen anterior podemos observar que la versión de Python instalada en Ubuntu 12.04 Python es la 2.7.3. Si tienen ganas de instalar la versión 3 de Python es tan sencillo como ejecutar „sudo apt-get install python3“ desde una consola o utilizar el gestor de software de cualquier versión de Linux.

## Python en Windows

Para aquellos usuarios de Windows que quieran contar con Python instalado en sus sistemas pueden descargar el instalador desde la página oficial <http://www.python.org/download/> e instalarlo según su sistema operativo sea de 32 o de 64 bits. Una vez que hayan descargado el instalador y estén listos para ejecutar. En la siguiente imagen pueden ver la página oficial de descarga de los instaladores de Python en donde van a poder elegir el que más se amolde a sus necesidades:

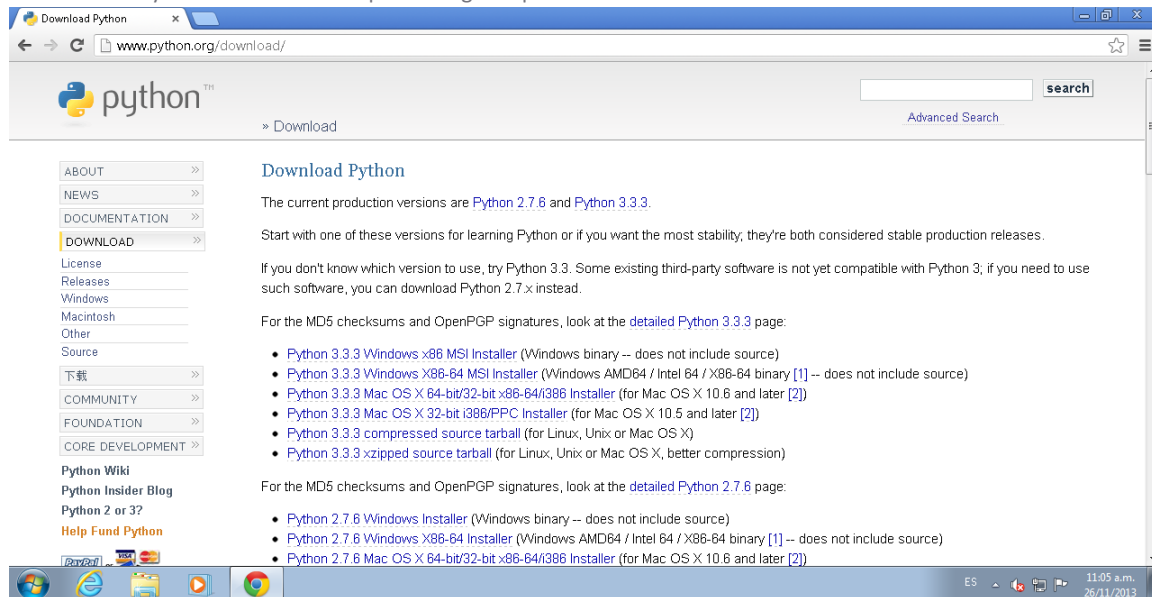


Imagen 2 - Página de descarga del instalador de Python

Se puede observar en la imagen anterior que hay dos versiones de Python que se pueden instalar y no vamos a entrar en las diferencias entre la versión 3.X y la 2.X. Para el presente curso de Introducción a Python no impacta la versión que instalen pero les recomendamos utilizar la versión 2.X ya que como se verá más adelante algunas herramientas utilizan esta versión de Python para extender sus capacidades.

No vamos a entrar en detalles de cómo llevar adelante el proceso de instalación pero una vez que finalicen y abran una ventana de comando (Inicio -> Ejecutar -> cmd) e ingresen la palabra „python“ deberían ver algo similar a lo que se muestra en la siguiente pantalla:

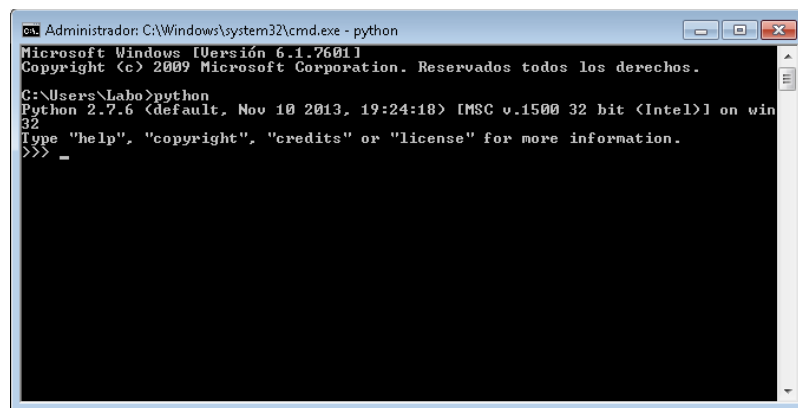


Imagen 3 - Python instalado en Windows

De esta manera se puede asegurar que sea cual sea el Sistema Operativo que decidan utilizar Python ya se encuentra instalado correctamente. Una diferencia a tener en cuenta, es que al momento de instalar la versión de Python para Windows, también podrán observar que se encuentra disponible **IDLE**, un editor o **IDE (Integrated Development Environment)** de Python que nos va a permitir escribir y probar el código.

## Cómo armar un entorno de desarrollo en Python

Ahora que ya cuentan con Python instalado en su sistema, llegó el momento de hablar de cómo desarrollar en Python o en dónde escribir el código para luego ejecutarlo. Como se podrán imaginar las opciones disponibles en el mercado son más que unas cuantas, y en pocas palabras lo único que necesitamos es un editor de texto.

Para todos aquellos que quieran probar diferentes opciones les recomendamos revisar la lista que se encuentra en el sitio oficial de Python<sup>4</sup> en dónde podrán ver según su sistema operativo y sus necesidades. En el marco del presente curso vamos a continuar utilizando IDLE con el objetivo de mantener los requerimientos de software al mínimo ya que lo que nos importa aprender es el lenguaje:

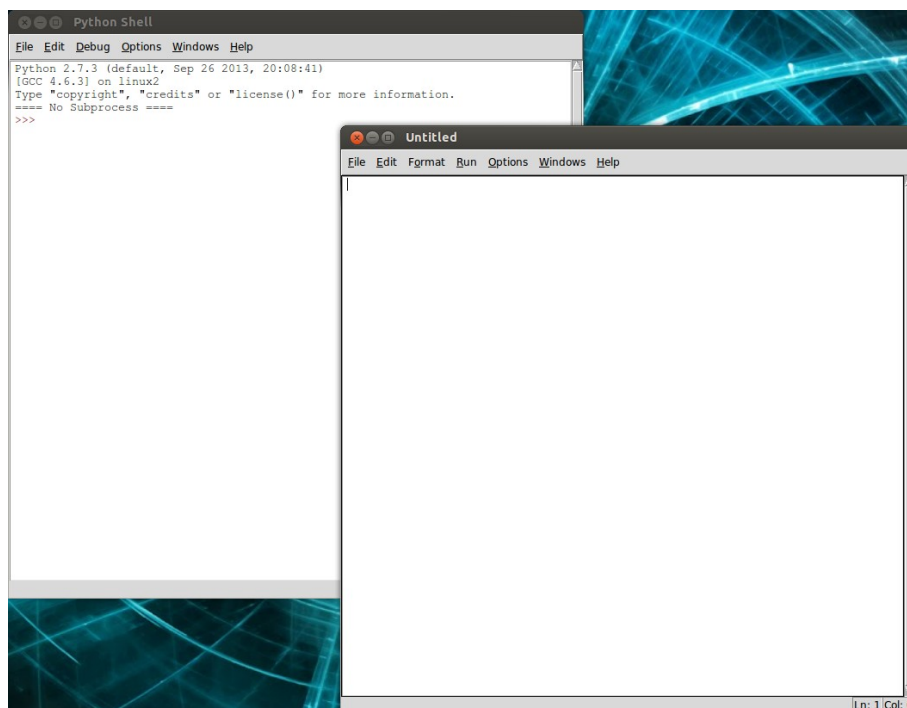


Imagen 4 - IDLE ejecutándose en Linux

<sup>4</sup> IDEs y software para desarrollar en Python: <https://wiki.python.org/moin/IntegratedDevelopmentEnvironments>

En caso de que quieran probar algún otro IDE no va a haber ningún problema, siéntanse libres de instalar el software que prefieran. IDLE puede parecer un poco minimalista pero no deja de ser extremadamente efectivo, liviano y práctico.

A medida que vayamos avanzando con los módulos del curso, irán apareciendo diferentes conceptos y maneras de organizar nuestro código de manera que sea ordenado y práctico para su uso. Les recomendamos crearse una carpeta en algún lugar de su sistema para ir almacenando todos los scripts y archivos que vayamos utilizando a lo largo del curso.

## En Windows por ahí...

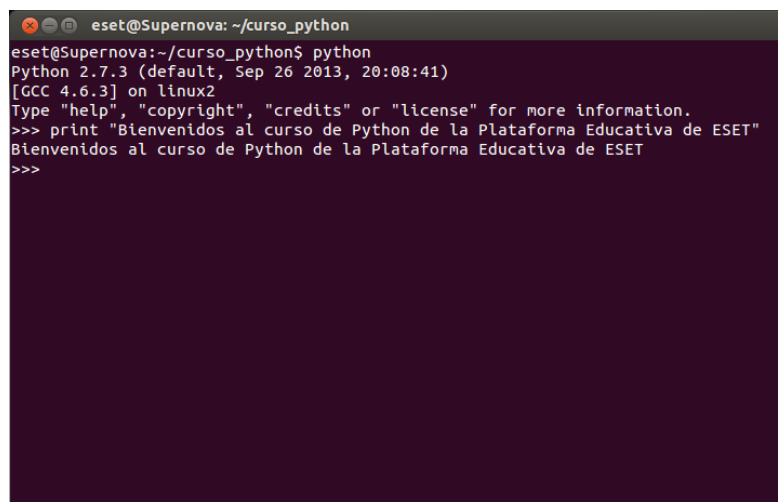
Si llegaron a elegir Windows como el sistema para avanzar con el curso, es posible que tengan que modificar algunas de las variables de entorno, como por ejemplo la variable PATH para que incluya el directorio de instalación de Python. De esta manera, desde cualquier consola van a poder ejecutar sus scripts sin ningún problema.

## Interactuando con Python

A grandes rasgos, existen dos maneras de ejecutar código en Python. La primera de ellas es ejecutar código de manera interactiva utilizando **la consola**, desde allí podrán probar sus algoritmos<sup>5</sup>, revisar funciones y métodos como así también leer la documentación. La segunda opción es a través de la ejecución de **scripts**, estos archivos de extensión **.py** contienen las líneas de código que escribimos y serán ejecutadas por el intérprete.

## La consola

La consola de Python es extremadamente práctica para probar pequeñas porciones de código, cargar nuestros módulos (se verá en detalle en el Módulo II), y ejecutar nuestras funciones. Además es una manera de interactuar con el usuario de una manera muy sencilla y directa para pedir información o para imprimir información:



```
eset@Supernova: ~/curso_python
eset@Supernova:~/curso_python$ python
Python 2.7.3 (default, Sep 26 2013, 20:08:41)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print "Bienvenidos al curso de Python de la Plataforma Educativa de ESET"
Bienvenidos al curso de Python de la Plataforma Educativa de ESET
>>>
```

*Imagen 5 - Bienvenidos al curso desde la consola*

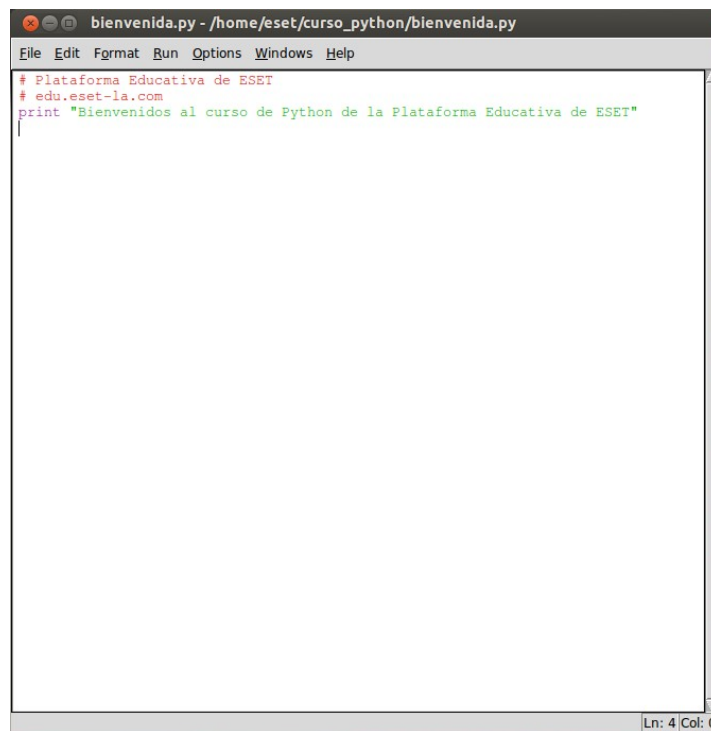
En la imagen anterior, desde una consola de **Python**, imprimimos un mensaje de bienvenida al curso. Para que se muestre la información por pantalla usamos la palabra **print** y entre comillas dobles le pasamos el mensaje a imprimir. Para ver más ejemplos acerca de cómo interactuar con la consola les recomendamos mirar el **Video 1 de este módulo**.

## Los Scripts

Les habíamos comentado que existe la posibilidad de escribir nuestras líneas de código en un archivo y luego pedirle al intérprete de Python que las ejecute o que corra nuestra **script**. Estos archivos de texto se guardan con la extensión **.py** y para llamarlos se puede pasarse como parámetro a Python o ejecutar directamente desde el IDE que hayan elegido.

En algún lugar del sistema, creen un archivo nuevo con el nombre **bienvenida.py** y escriban en él las siguientes líneas:

<sup>5</sup> ¿Qué es un algoritmo?: <http://es.wikipedia.org/wiki/Algoritmo>

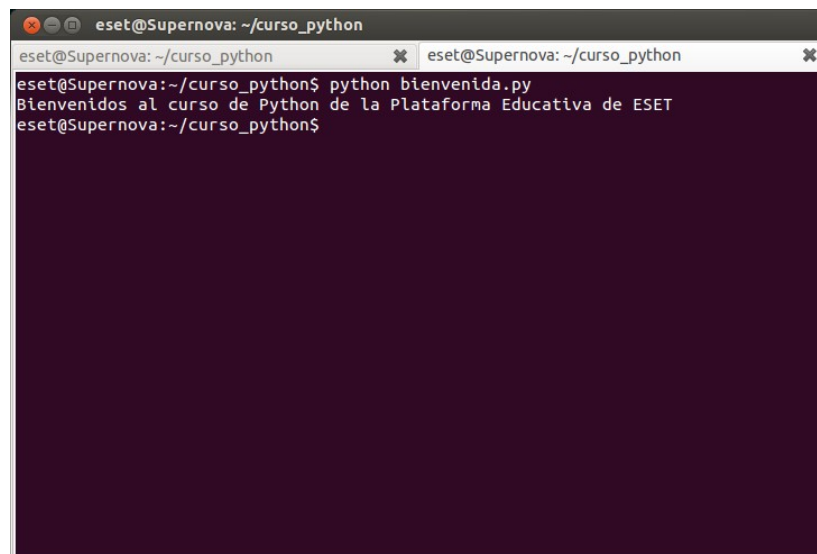


```
# Plataforma Educativa de ESET
# edu.eset-la.com
print "Bienvenidos al curso de Python de la Plataforma Educativa de ESET"
```

*Imagen 6 - Script de Bienvenida*

Una vez que crearon el archivo, escribieron las líneas en él y lo guardaron, estamos listos para ejecutarlo y ver qué es lo que pasa. Si repasamos las líneas del archivo vemos que hay dos líneas que comienzan con un „#“, esto denota que se trata de un comentario, y no se ejecutará nada.

Los comentarios en el código sirven para documentar, explicar y aclarar qué es lo que hace un módulo, script o función y son extremadamente útiles para compartir nuestro código y que otra persona lo pueda leer. Ahora que está todo listo, cerramos el editor y desde una consola, nos ubicamos en el sitio en el cual guardamos nuestro archivo y ejecutamos el comando „python bienvenida.py“, la salida debería ser igual a la siguiente imagen:



```
eset@Supernova: ~/curso_python
eset@Supernova: ~/curso_python$ python bienvenida.py
Bienvenidos al curso de Python de la Plataforma Educativa de ESET
eset@Supernova: ~/curso_python$
```

*Imagen 7 - Ejecutando un script desde la consola*

Tal y como les comentamos, se ejecuto nuestro script *bienvenida.py*, y se mostró por pantalla el saludo de bienvenida pero no los comentarios que introdujimos.



## „Hola Mundo“ en Python

A estas alturas, ya se encuentran en condiciones de escribir el programa más tradicional al momento de aprender un lenguaje. El famoso „Hola Mundo“. Abran una consola de Python e impriman el saludo por pantalla y luego guarden la línea que escribieron en un archivo y ejecutenlo.

## Conclusión

En este módulo, vimos varios de los conceptos más importantes de Python como así también sus principales componentes y características. De ahora en adelante nos iremos adentrando en este lenguaje de programación para conocer de qué manera crear programas o scripts que nos ayuden en nuestras tareas diarias como así también a analizar códigos maliciosos.

Los invitamos a continuar con este curso de la Plataforma Educativa de ESET para continuar aprendiendo acerca de este lenguaje.