

Introducción a la programación con Python



Taller de programación #1

Descripción

Este documento contiene información relacionada a las actividades de prácticas del primer taller del curso Introducción a la Programación con Python. En este taller empezaremos con una pequeña revisión de las herramientas que usaremos en este curso. A continuación mostraremos cómo correr un programa básico en Python para luego continuar la explicación de cómo recibir e imprimir información. Por favor note que el taller se ha ejecutado en una computadora con el sistema operativo Windows 10.

1. Revision

1.1 Anaconda

Anaconda es una distribución libre y abierta de los lenguajes Python y R, utilizada en ciencia de datos, y aprendizaje automático (machine learning). Anaconda permite la creación de entorno de desarrollo, la instalación de librerías y el uso de libros jupyter de manera amigable al usuario. Anaconda se puede descargar desde su página original <https://www.anaconda.com/> , anaconda puede ser instalado para Windows, MacOS y Linux.

1.2 Command Prompt (Simbolo del sistema)

Permite al usuario interactuar directamente con el sistema operativo, los comandos son en forma de línea de textos, antiguamente se usaba para controlar el sistema puesto que no existía mouse, control de manejo de ventanas o GUI.

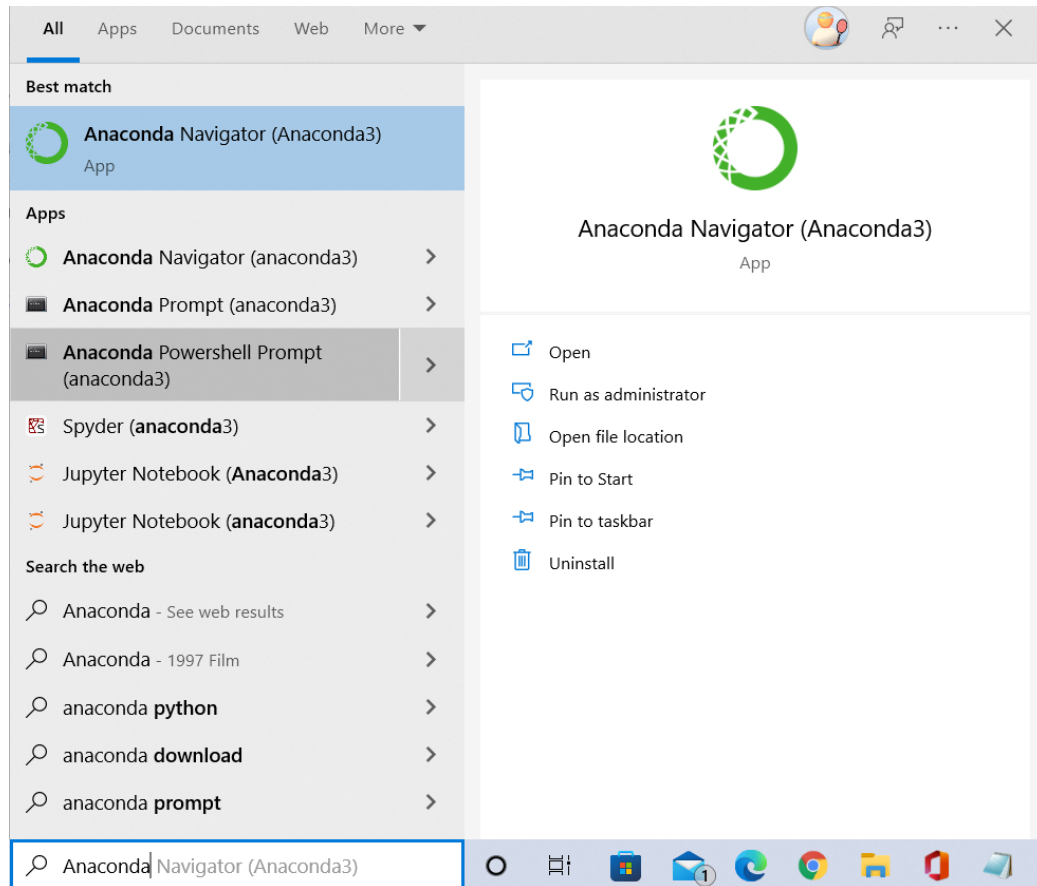
1.3 Editor de texto

Es un programa que permite crear o modificar archivos de forma digital. Pueden decidir usar el editor de texto de su preferencia para este y los próximos talleres. En caso de no haber usado un editor de texto anteriormente les recomendamos atom. Si desean descargarlo pueden hacerlo usando el siguiente enlace: <https://atom.io/>

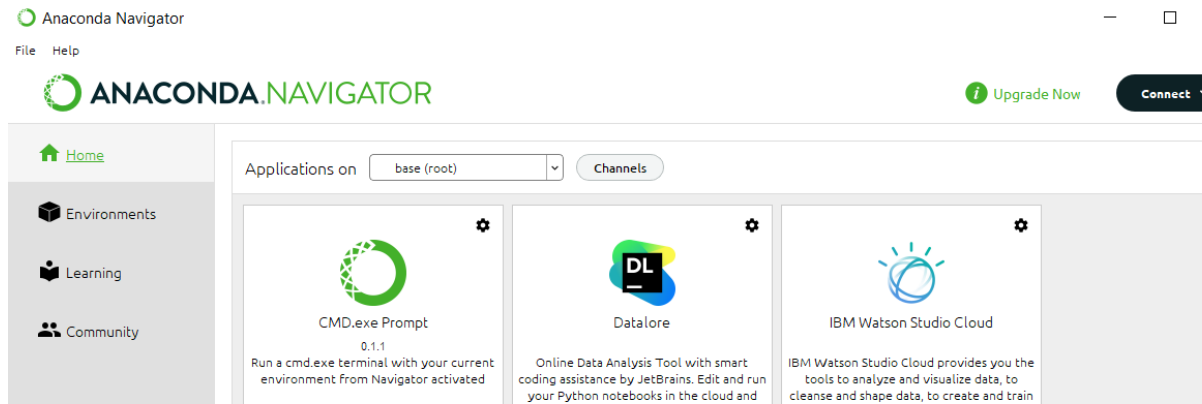
2. Configuración

2.1 Creando un entorno de desarrollo

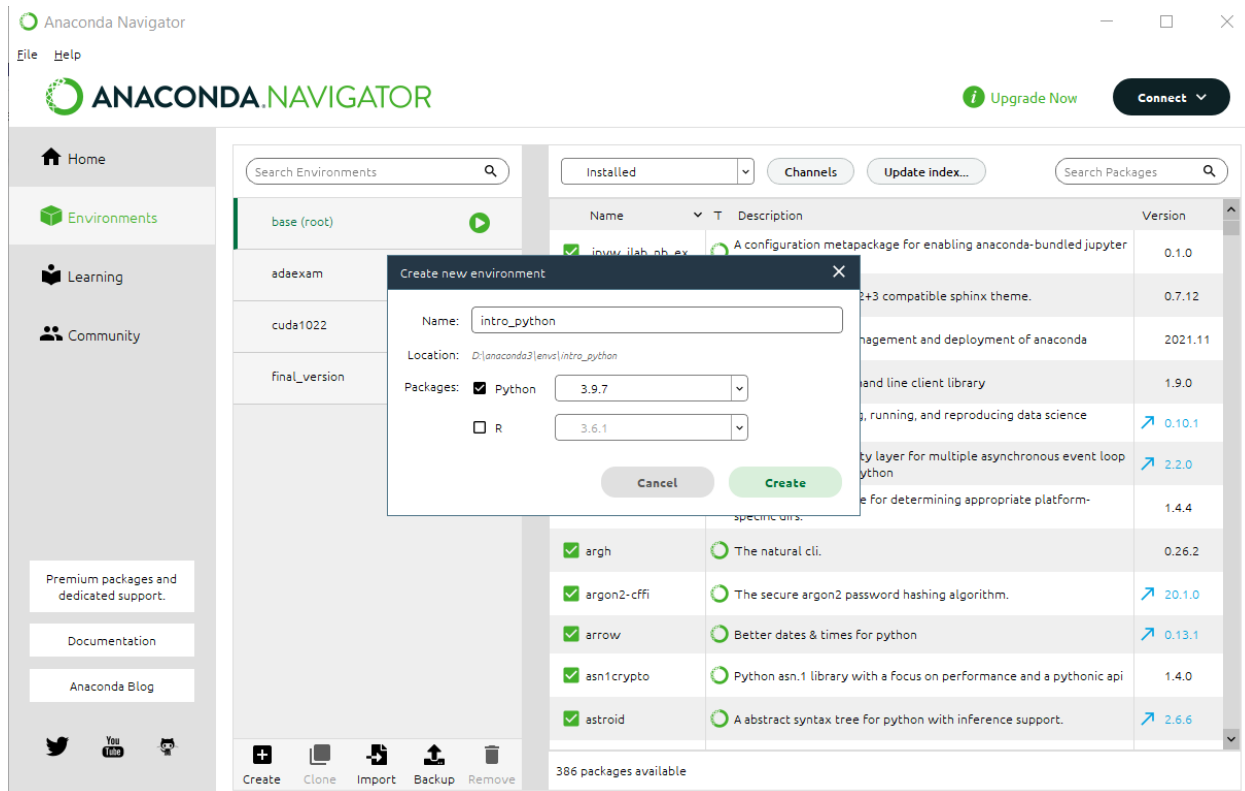
Una vez que Anaconda sea instalada en su computador, diríjase a la barra de herramientas de Windows > barra de búsqueda > escribir “Anaconda”



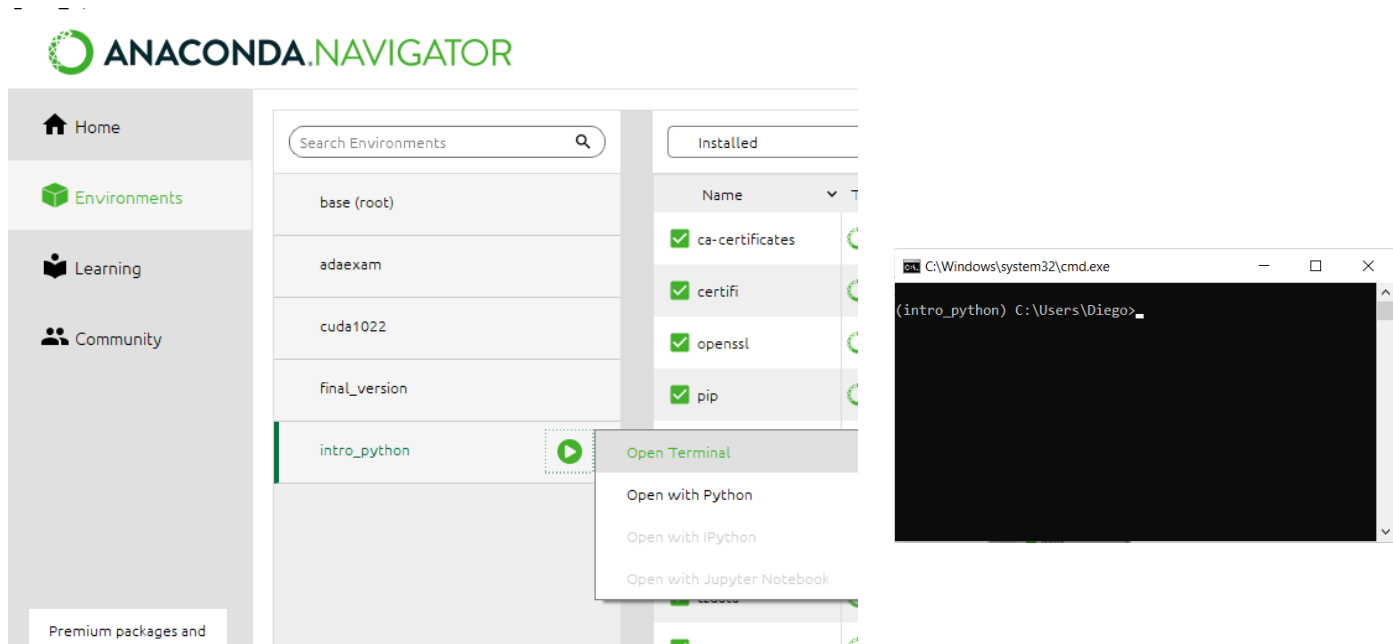
Inicie Anaconda Navigator arrastrando su mouse hasta la aplicación y dando click derecho



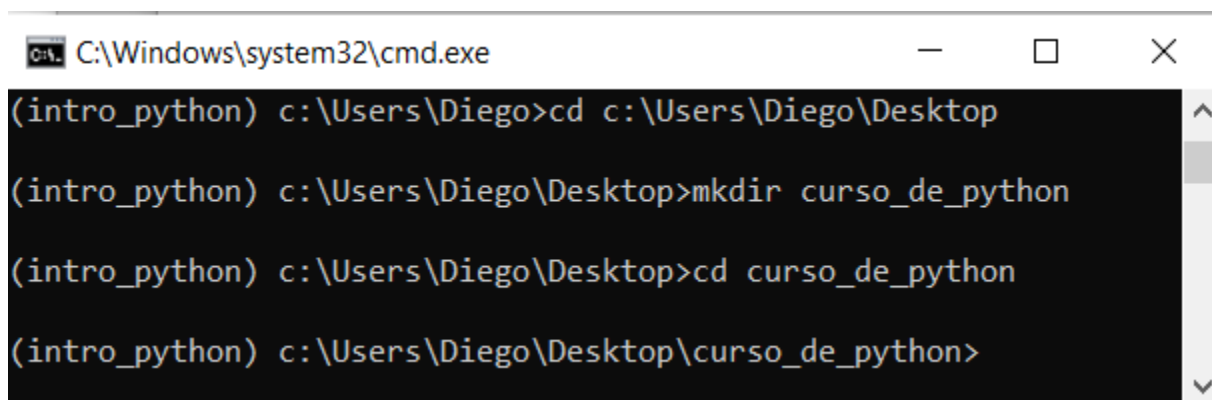
A la derecha de la aplicación, seleccione “Environments” > “Create” > Marque la casilla Python y de un nombre a su entorno de desarrollo.



Una vez creado su entorno de desarrollo, inicie presionando el botón play y seleccionando "Open Terminal"

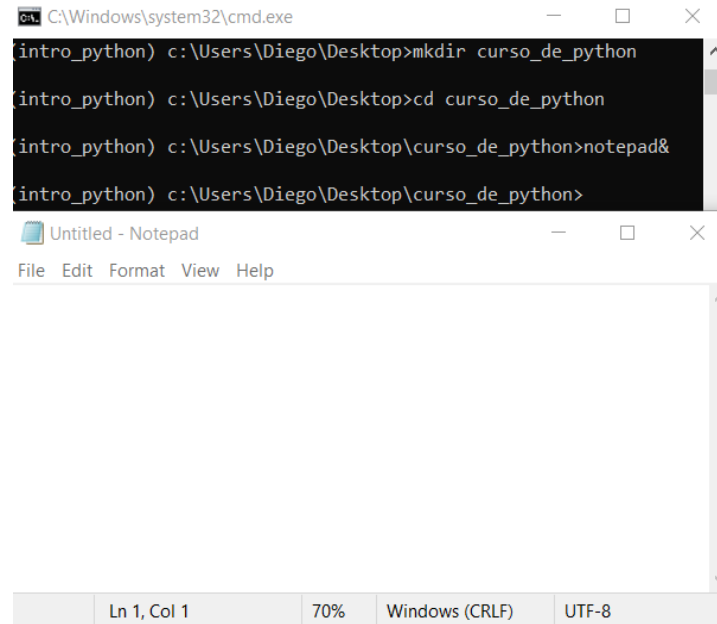


Al hacer esto podrá observar el símbolo de sistema o command prompt en su escritorio. Con el fin de organizar nuestro código, crearemos una carpeta en el escritorio. Para esto ejecute en su símbolo de sistema > cd c:\Users\<nombre_de_usuario>\Desktop > mkdir curso_de_python > cd curso_de_python



3. Mi primer programa

Usaremos el editor de texto "Notepad". Para abrir el programa podemos llamarlo desde el símbolo de sistema con "notepad &"

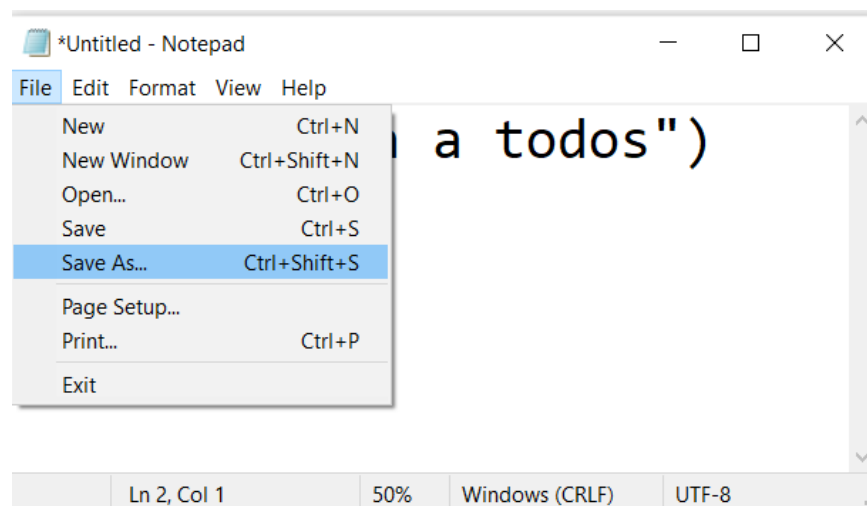


The image shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe" with the following commands entered:
(intro_python) c:\Users\Diego\Desktop>mkdir curso_de_python
(intro_python) c:\Users\Diego\Desktop>cd curso_de_python
(intro_python) c:\Users\Diego\Desktop\curso_de_python>notepad&
(intro_python) c:\Users\Diego\Desktop\curso_de_python>
Below the command prompt is a Notepad window titled "Untitled - Notepad" with a menu bar (File, Edit, Format, View, Help) and a status bar showing "Ln 1, Col 1", "70%", "Windows (CRLF)", and "UTF-8".

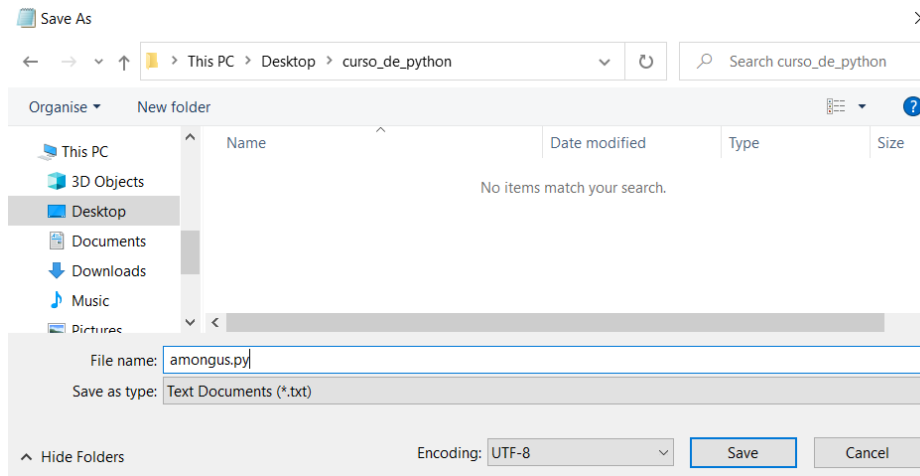
Dentro del editor podemos escribir sintaxis de Python. Para nuestro primer programa haremos que la consola muestra un mensaje de bienvenida, para esto usaremos una palabra especial en python llamada print (print nos permite enviar parámetros e imprimirlos linea por linea)

```
print ("Hola a todos")
```

Ahora necesitamos guardar este documento en la carpeta que creamos anteriormente, para esto dirígete al editor de texto File > Save As



Usando el navegador, guarda tu archivo con el siguiente nombre `tu_nombre.py`



Desde el símbolo de sistema, llamaremos al interpretador de Python usando `python` `tu_nombre.py`

```
C:\Windows\system32\cmd.exe
(intro_python) c:\Users\Diego\Desktop\curso_de_python>python amongus.py
Hola a todos
```

4. Saludando desde el simbolo de sistema

Recibir datos desde el símbolo del sistema es posible en python, gracias a la ayuda de `input()`, de esta manera obtendremos un dato de tipo **string**, la cual puede ser guardada en un variable, finalmente modificaremos `print` para imprimir esta.

```
nombre = input("Ingresa tu nombre: ")
print ("Hola ", nombre)
```

```
C:\Windows\system32\cmd.exe
(intro_python) c:\Users\Diego\Desktop\curso_de_python>python amongus.py
Ingresa tu nombre: Amongus
Hola Amongus
(intro_python) c:\Users\Diego\Desktop\curso_de_python>
```


5. Calculadora

Ahora te toca a ti, usando lo ha aprendido en este día desarrolla una calculadora que obtenga el siguiente resultado en el símbolo del sistema:

```
(intro_python) c:\Users\Diego\Desktop\curso_de_python>python amongus.py
Ingrese operador: +
Ingrese primer numero: 10
Ingrese segundo numero: 5
15.0

(intro_python) c:\Users\Diego\Desktop\curso_de_python>python amongus.py
Ingrese operador: -
Ingrese primer numero: 10
Ingrese segundo numero: 5
5.0

(intro_python) c:\Users\Diego\Desktop\curso_de_python>python amongus.py
Ingrese operador: *
Ingrese primer numero: 10
Ingrese segundo numero: 5
50.0

(intro_python) c:\Users\Diego\Desktop\curso_de_python>python amongus.py
Ingrese operador: /
Ingrese primer numero: 10
Ingrese segundo numero: 5
2.0

(intro_python) c:\Users\Diego\Desktop\curso_de_python>python amongus.py
Ingrese operador: LOL
Ingrese primer numero: 10
Ingrese segundo numero: 10
Operador invalido
```

Por ahora, no te preocupes de aquellos casos donde el usuario ingresa datos que no son válidos para la calculadora.