



**UTN.BA**  
UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

**Centro de  
e-Learning**

**UNIDAD DIDÁCTICA I**  
**DIPLOMATURA EN PYTHON**

**Centro de e-Learning SCEU UTN - BA.**

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

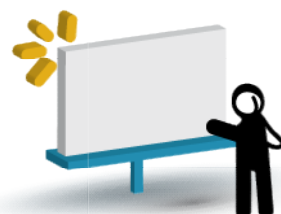
**[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)**

## **Módulo I – Nivel Inicial I**

### **Unidad I – Introducción y estructura.**

**Centro de e-Learning SCEU UTN - BA.**

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148  
**[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)**



## Presentación:

Python es un lenguaje multiplataforma que puede ser programado mediante POO (Programación Orientada a Objetos), su sintaxis es simple difiriendo de otros lenguajes como Java, C, PHP,..etc; en la ausencia de paréntesis y la no incorporación de punto y coma al final de cada instrucción entre otras cosas.

Es un lenguaje muy completo, con soporte para bases de datos relacionales u orientados a objetos, capaz de incorporar ORM's, con frameworks de desarrollo tan completos como Django, entre otros.

Python permite trabajar en desarrollos de escritorio, web, aplicaciones móviles, videojuegos, etc.

En esta unidad vamos a realizar una rápida revisión de los conceptos básicos de instalación y daremos nuestros primeros pasos en este completo lenguaje, sobre la versión 3.7.



## Objetivos:

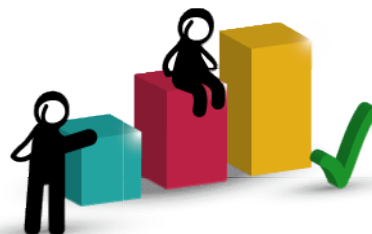
### Que los participantes:

- Logren llevar a cabo la instalación de Python 3.7 sobre la plataforma de windows.
- Comprendan el concepto de variable y asignación dinámica.



## Bloques temáticos:

- 1.- Instalación.
- 2.- Herramientas Útiles.
- 3.- Como funciona Python.
- 4.- Variables y comentarios
- 5.- Asignación dinámica, garbage collection y referencias compartidas.
- 6.- GUI.



## Consignas para el aprendizaje colaborativo

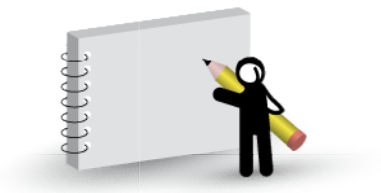
En esta Unidad los participantes se encontrarán con diferentes tipos de actividades que, en el marco de los fundamentos del MEC\*, los referenciarán a tres comunidades de aprendizaje, que pondremos en funcionamiento en esta instancia de formación, a los efectos de aprovecharlas pedagógicamente:

- Los foros proactivos asociados a cada una de las unidades.
- La Web 2.0.
- Los contextos de desempeño de los participantes.

Es importante que todos los participantes realicen algunas de las actividades sugeridas y compartan en los foros los resultados obtenidos.

Además, también se propondrán reflexiones, notas especiales y vinculaciones a bibliografía y sitios web.

El carácter constructivista y colaborativo del MEC nos exige que todas las actividades realizadas por los participantes sean compartidas en los foros.



## Tomen nota

Las actividades son opcionales y pueden realizarse en forma individual, pero siempre es deseable que se las realice en equipo, con la finalidad de estimular y favorecer el trabajo colaborativo y el aprendizaje entre pares. Tenga en cuenta que, si bien las actividades son opcionales, su realización es de vital importancia para el logro de los objetivos de aprendizaje de esta instancia de formación. Si su tiempo no le permite realizar todas las actividades, por lo menos realice alguna, es fundamental que lo haga. Si cada uno de los participantes realiza alguna, el foro, que es una instancia clave en este tipo de cursos, tendrá una actividad muy enriquecedora.

Asimismo, también tengan en cuenta cuando trabajen en la Web, que en ella hay de todo, cosas excelentes, muy buenas, buenas, regulares, malas y muy malas. Por eso, es necesario aplicar filtros críticos para que las investigaciones y búsquedas se encaminen a la excelencia. Si tienen dudas con alguno de los datos recolectados, no dejen de consultar al profesor-tutor. También aprovechen en el foro proactivo las opiniones de sus compañeros de curso y colegas.

## Glosario

**Centro de e-Learning SCEU UTN - BA.**

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

**[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)**

**GUI:** Interfaz gráfica.

**IDLE:** Editor de texto que viene por defecto en cada distribución de python.

**Path:** ruta al ejecutable de python. En Windows la variable path contiene las rutas a los programas principales del sistema operativo.

**Objeto:** Esta palabra se utilizará con dos finalidades. En primer lugar llamaremos objeto a todo elemento que podemos manejar desde python, desde un archivo, una base de datos, o una variable en general. En segundo lugar en el nivel intermedio lo utilizaremos para definir la instancia de una clase, es decir la creación de un objeto de la clase. Estos términos escapan el nivel del curso inicial y lo veremos en el nivel intermedio.

**Objeto inmutable:** Es aquel que sí es modificado se convierte en otro objeto para python.

**Objeto mutable:** Es aquel que puede ser modificado parcialmente y python lo sigue considerando el mismo objeto.

**PVM (Python Virtual Machin):** programa que realiza el papel de Intérprete del lenguaje.

**Variable:** Una variable se puede considerar como un símbolo que puede ser reemplazado o que toma un valor determinado, como puede ser un valor numérico en una ecuación o expresión matemática en general. Las variables se pueden utilizar para guardar datos de diferentes tipos, por ejemplo: enteros, caracteres, listas, arrays, diccionarios u objetos en general.

**Variables de entorno:** variables del sistema operativo Windows que deben ser modificadas durante la instalación para el correcto funcionamiento de python.



## 1. Instalación

La versión de Python que vamos a instalar es la 3.7 lanzada el 27 de junio del 2018, esta versión posee una performance mucho mayor a las anteriores y a la fecha es la última versión estable.

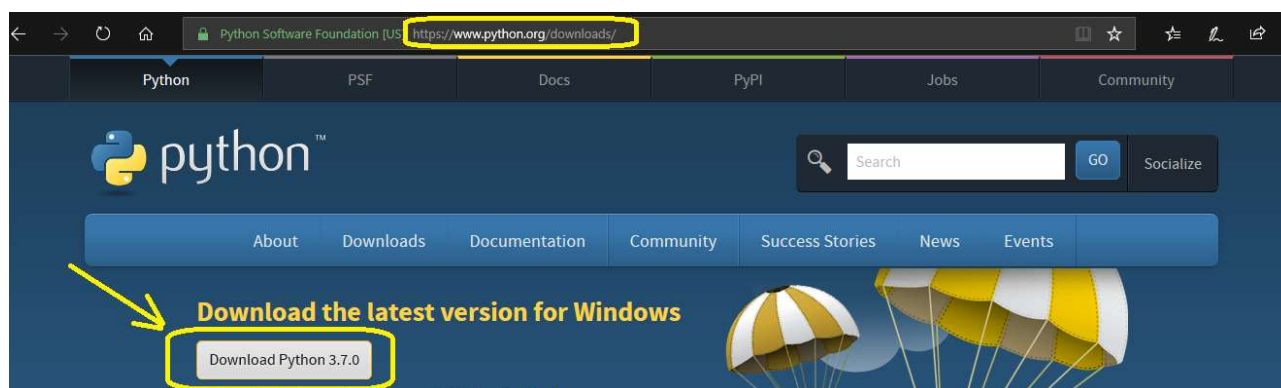
**Nota:** Si en el momento de leer el material el alumno encuentra una actualización, puede descargar la nueva versión y seguir los mismos pasos.

### 1.1. Descargar

Accedemos al sitio:

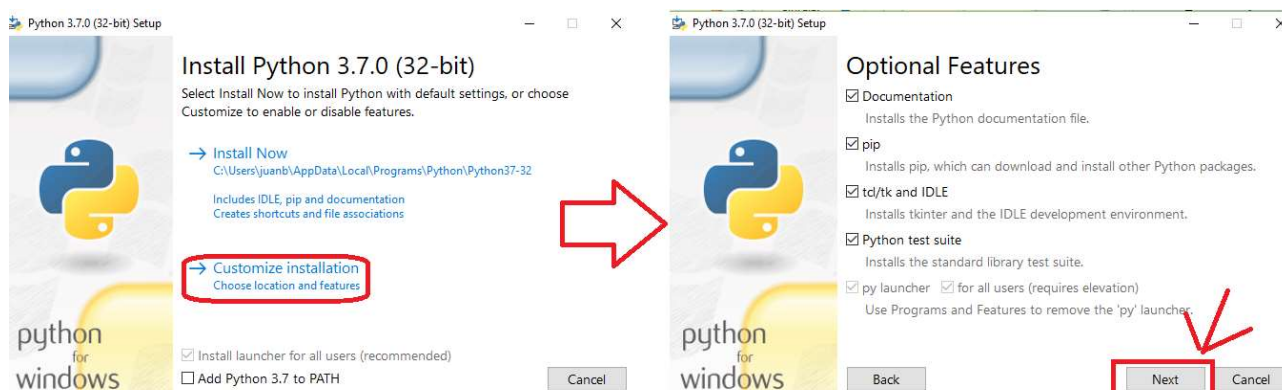
<https://www.python.org/downloads/>

Y descargamos el paquete correspondiente al sistema operativo que tenemos, en este caso para trabajar en facultad, lo haremos sobre windows de 64bit el cual es un ejecutable.

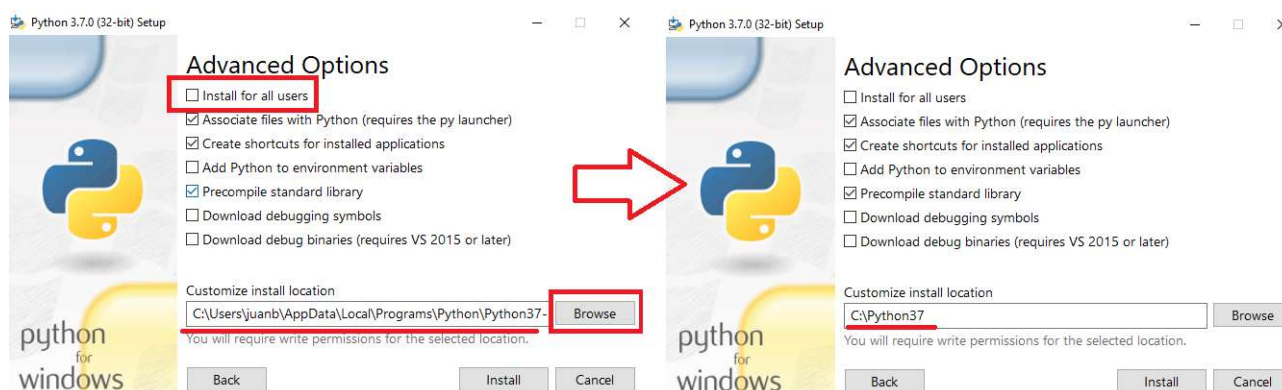


### 1.2. Instalación

A partir de la versión 3.5, la instalación solo requiere que presionemos en “Install Now” y que tildemos el campo de “Add Python 3.X to PATH”, aquí seleccionaremos la opción que nos permite personalizar la instalación para poder seleccionar el directorio de instalación y comprender cómo realizar la instalación en versiones previas. No tildamos el campo para agregar la ruta al path.



Por defecto la opción de instalación lo instala para el usuario actual a no ser que seleccionemos que lo instale para todos los usuarios o introduzcamos la ruta donde queremos instalarlo buscándola con el botón “Browse”.



Debemos tener cuidado en el caso de seleccionar una carpeta a partir de Browse ya que si seleccionamos por ejemplo el disco “C”, agregaría todos los archivos sueltos dentro del disco. Para evitar esto indicamos el nombre de la carpeta que queremos que cree en el momento de la instalación dentro del disco para que todos los archivos se agreguen ahí dentro. Como ejemplo podríamos poner:

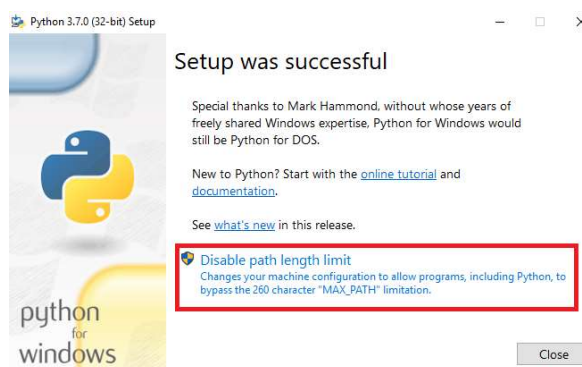
**C:\Python37**

Y se crearía la carpeta “Python37” con la finalidad de contener todos los archivos a instalar.

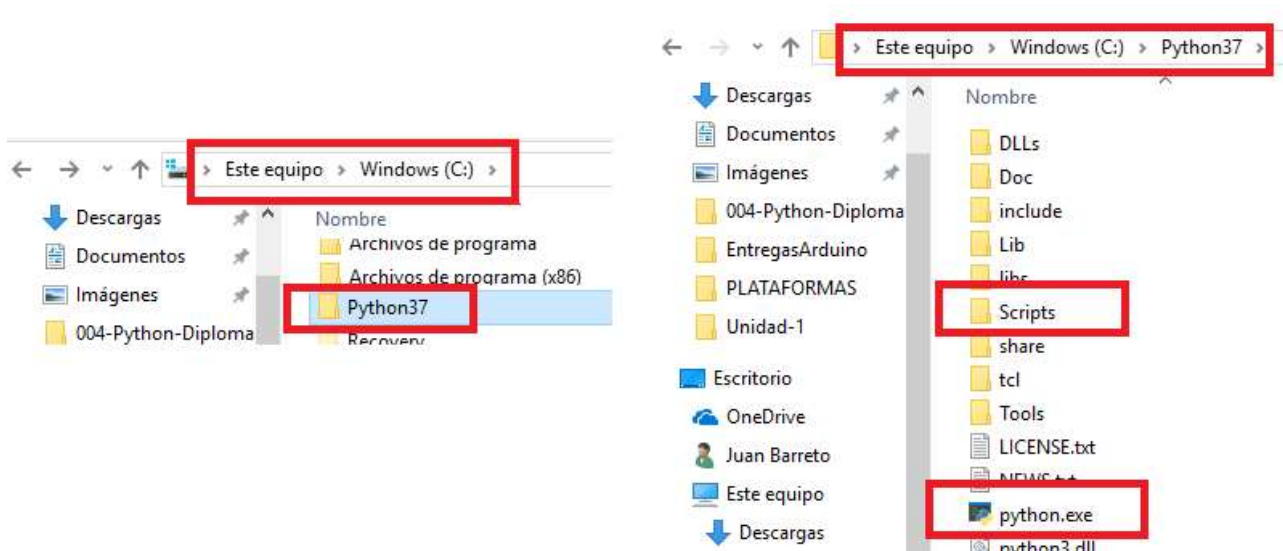
Finalmente presionamos “next” para comenzar la instalación, y deshabilitamos la limitación del path para finalizar.

**Centro de e-Learning SCEU UTN - BA.**

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148  
[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)



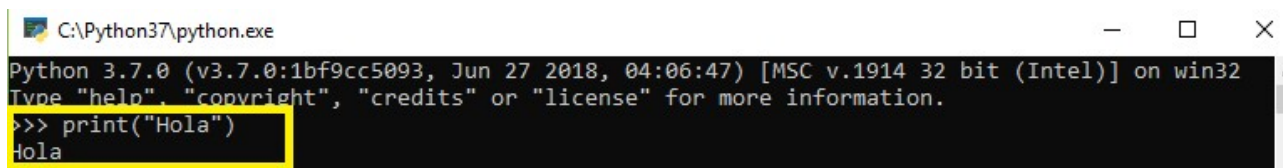
Ya podemos ingresar a la carpeta y observar que dentro se encuentra el ejecutable de python “python.exe” y un directorio muy importante “Script” al cual haremos referencia un poco más adelante en esta unidad.



Si damos doble click sobre el ejecutable se nos abrirá una terminal desde la cual ya podemos utilizar python, a modo de ejemplo escribiremos:

### Print (“Hola”)

Podremos observar cómo nos retorna “Hola”.



**Centro de e-Learning SCEU UTN - BA.**

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148  
[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)



## Configuración.

En nuestro sistema operativo, podríamos tener instalada cualquier versión de python, incluso todas ellas al mismo tiempo, sin embargo la versión que estaría ejecutándose en el sistema sería aquella que se encontrara agregada en las variables de entorno (VER pág 13) del usuario que está logueado. Si abrimos el cmd del sistema y escribimos:

```
python
```

Nos saldría un mensaje diciendo que **“python no se reconoce como un comando interno o externo, programa o archivo por lotes ejecutable”** ya que no reconoce que python esté instalado.

Sin embargo si escribimos:

```
set path=%path%;C:\python37
```

Vemos como ahora podemos escribir python e ingresaríamos a la versión de Python que se encuentra indicada en la ruta resaltada en rojo en la línea de código anterior, pudiendo ahora repetir la línea “print(“Hola”) y se comportaría exactamente igual que al haber abierto el ejecutable que se encuentra dentro de la carpeta de instalación.



```
Simbolo del sistema - python
Microsoft Windows [Versión 10.0.17134.285]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

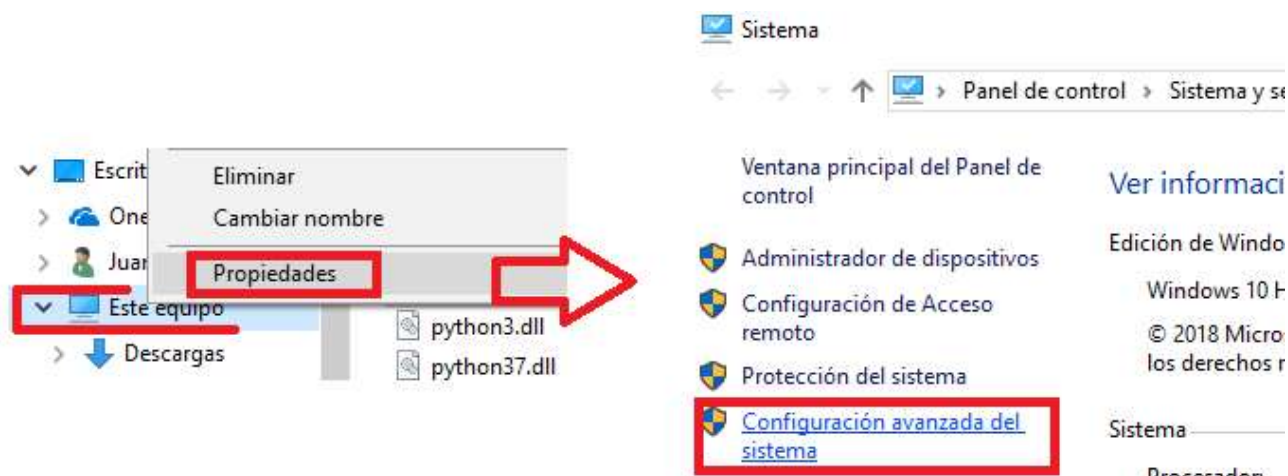
C:\Users\juanb>set path=%path%;C:\python37

C:\Users\juanb>python
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hola")
Hola
```

Lo que hemos realizado fue agregar dinámicamente la versión de python 3.7 a las variables de entorno del sistema operativo para el usuario actual de forma que mientras esta ventana del cmd se encuentre abierta podamos trabajar con ella con python. Si abrimos otra ventana vemos como nuevamente el comando “python” no se reconoce.

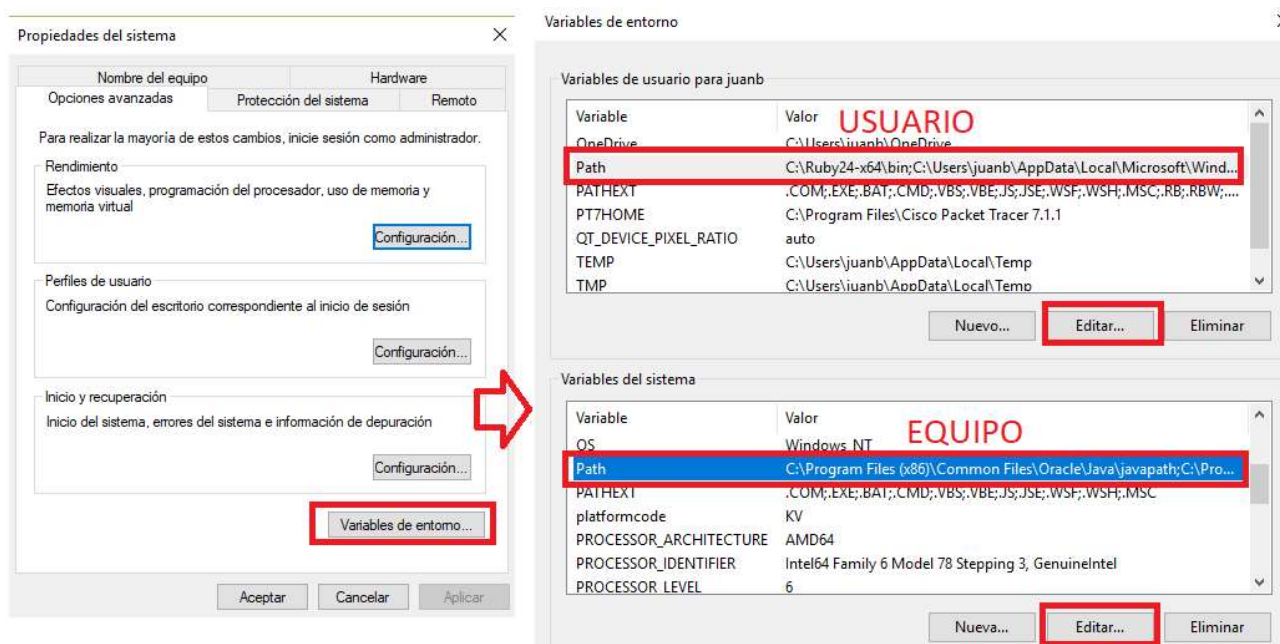
### ¿Cómo se agrega de forma permanente una versión de python?

Cómo comentamos un poco más arriba, es posible tener varias versiones de python instaladas pero solo hacer accesible una versión desde nuestro sistema operativo, para ello abrimos una carpeta cualquiera en Windows y hacemos click derecho sobre el ícono de “Equipo” y luego seleccionamos “Propiedades”. En la nueva ventana seleccionamos “Configuración avanzada del sistema”





Accedemos a una ventana emergente desde la cual debemos ir a “Variables de entorno” para acceder a las secciones desde las cuales poder agregar a la ruta “Path” mediante el botón “Editar”, la ruta a la versión de python que queremos utilizar ya sea para el usuario actual o para todos los usuarios del equipo.



**Nota:** Solo la versión de python agregada en la ruta “Path” es utilizada en el sistema operativo, por lo que cuando descarguemos paquetes de terceros para extender nuestras funcionalidades de python, estos paquetes se instalarán en la versión que se encuentre indicada en la variable “Path”.

Las rutas que debemos agregar son:

**C:\Python37**  
**C:\Python37\Scripts**

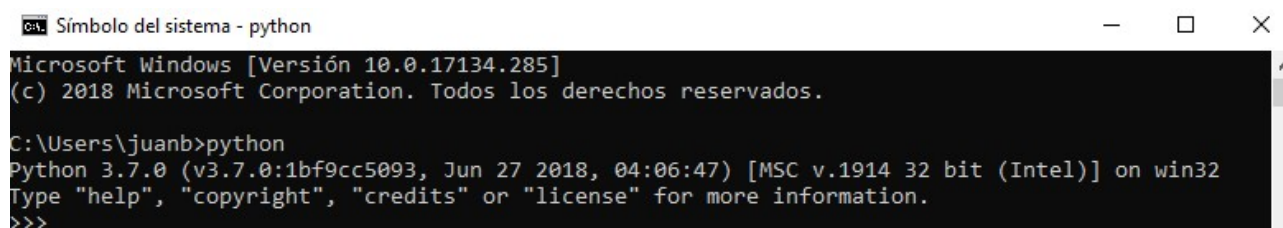
En Windows 10 es más fácil porque podemos agregarlas desde la ventana emergente de a una, sin embargo en versiones previas de Windows todas las rutas venían en una sola línea separadas por punto y coma (;) y debíamos agregarlas al final de la línea de esta forma:

**Contenido-del-path;C:\Python37;C:\Python37\Scripts;**

**Nota:** Debemos tener mucho cuidado de no borrar ninguna de las rutas de la variable “Path” porque podríamos generar grandes problemas en la ejecución de nuestro sistema operativo, si creemos que por accidente modificamos algo, debemos cancelar la operación.

### 1.3. Evaluar si la instalación se realizó correctamente.

Para verificar que todo esté funcionando correctamente, abrimos el cmd y tipeamos “python”. Si todo está bien debemos ver un mensaje como el siguiente:



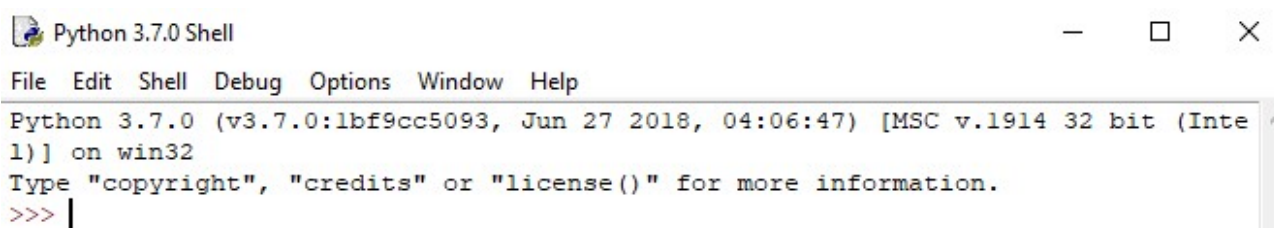
```
Símbolo del sistema - python
Microsoft Windows [Versión 10.0.17134.285]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\juanb>python
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

## 2. Herramientas Útiles

### 2.1. Uso del IDLE.

Dentro de las cosas que se han instalado en nuestra máquina, se encuentra el IDLE (Editor de texto que viene por defecto con cada distribución de python), lo abrimos y debería de presentar una apariencia como la de la siguiente imagen:



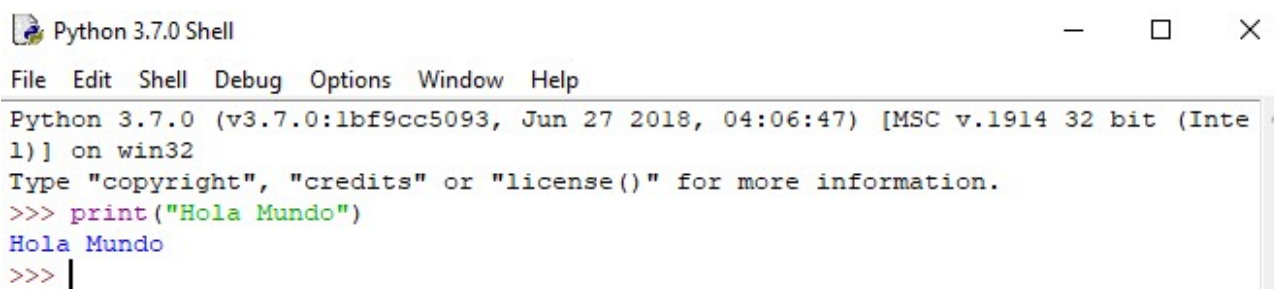
Mediante el IDLE de python podremos evaluar el correcto funcionamiento de nuestros scripts y obtener un detalle de los posibles errores.

Como ejemplo podemos realizar nuestro primer “Hola Mundo” escribiendo:

```
print(“Hola Mundo”)
```

#### Notar que:

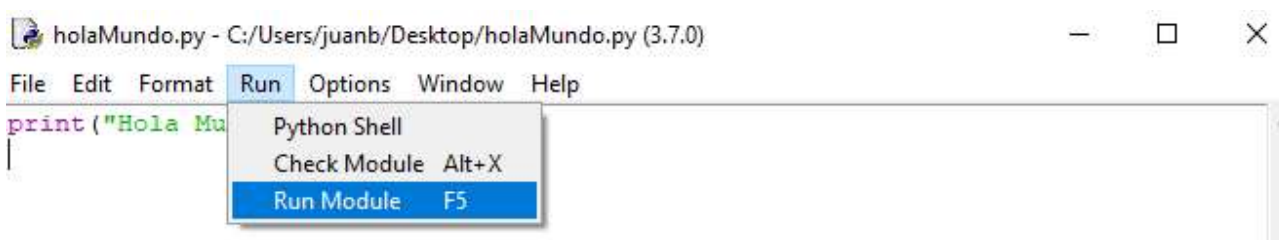
- 1.- En el código, no se agrega punto y como al final de print, como en muchos de los programas actuales (php, javascript, .....)
- 2.- El IDLE asigna un código de colores, el cual puede ser editable.



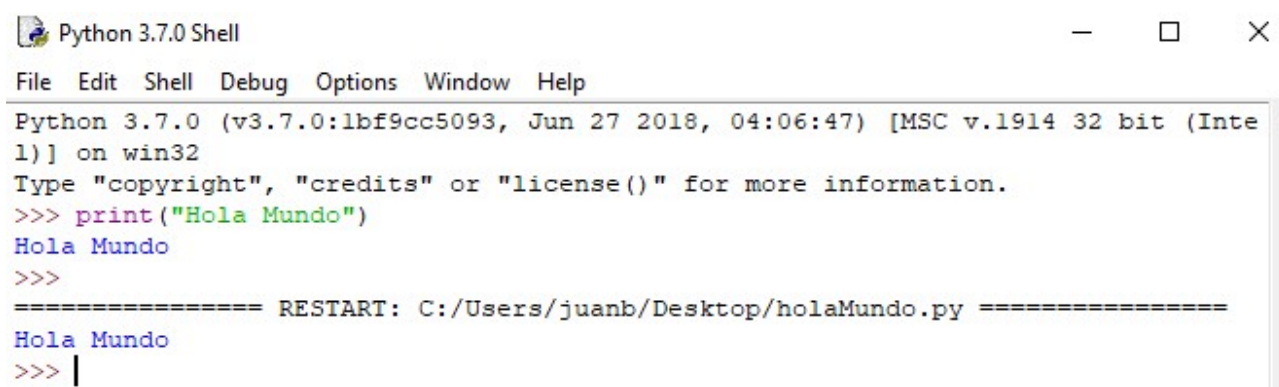


### 2.1.1. Crear y ejecutar script con el IDLE

Si vamos a File en el margen superior izquierdo del IDLE podemos crear un documento nuevo, al cual podemos salvar con un nombre. Si ahora dentro del archivo generado copiamos nuestro `print("Hola Mundo")`, guardamos el archivo y vamos a Run > Run Module F5. Se abre otra ventana del IDLE que nos muestra el resultado de la ejecución y de existir errores un mensaje con el detalle de los mismos.



Visualización en la nueva ventana:



### 2.1.2. Recobrar código en el IDLE

#### **En Windows y linux**

Alt-P for Previous y Alt-N for Next

#### **En Mac**

Ctrl-P y and Ctrl-N.

**Centro de e-Learning SCEU UTN - BA.**

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148  
[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)

### 2.1.3. Ejecutar script en Windows desde el cmd.

Cuando queremos ejecutar un script en windows desde el cmd, debemos pararnos en el directorio en el cual se encuentra y escribir “python nombreArchivo” y este se ejecutará.

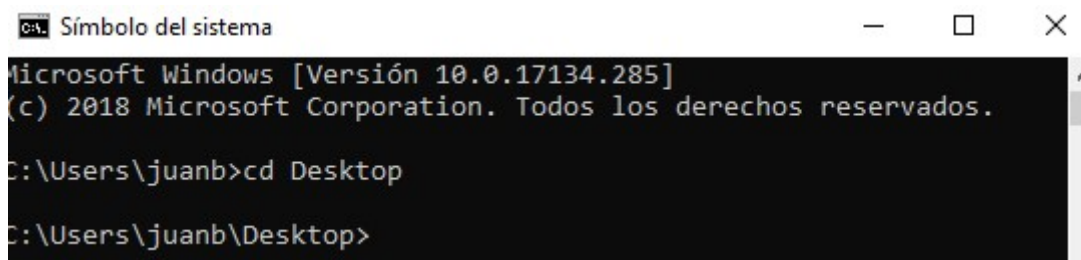
Probemos con el ejemplo anterior, yo he guardado en el escritorio el archivo “holaMundo.py” el cual contiene la línea de código:

```
holaMundo.py
```

```
print(“Hola Mundo”)
```

Para ejecutarlo primero abro el cmd, sabiendo que para ingresar a un directorio lo hago con la palabra “cd” seguida del nombre del directorio y que para ir hacia atrás se utiliza “..”. Dado que en mi caso el cmd se abre desde el usuario actual y que el archivo se encuentra en el escritorio, escribo:

```
cd Desktop
```



A screenshot of a Windows Command Prompt window titled "Símbolo del sistema". The window shows the following text: "Microsoft Windows [Versión 10.0.17134.285]", "(c) 2018 Microsoft Corporation. Todos los derechos reservados.", "C:\Users\juanb>cd Desktop", and "C:\Users\juanb\Desktop>". The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

Con la instrucción anterior ingreso al escritorio y ahora ejecuto:

```
python holaMundo.py
```

Podemos ver cómo nos aparece en pantalla el resultado de la ejecución.



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.17134.285]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

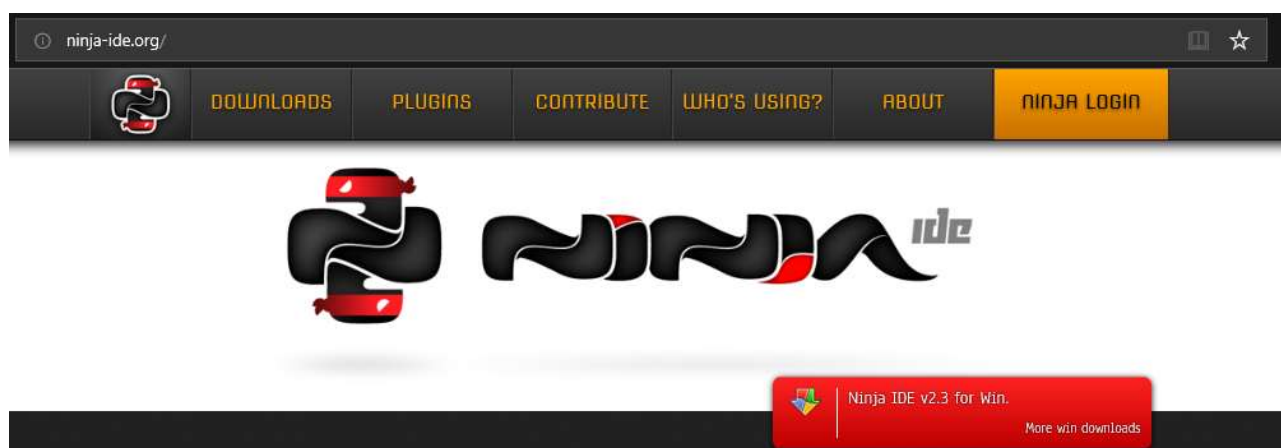
C:\Users\juanb>cd Desktop

C:\Users\juanb\Desktop>python holaMundo.py
Hola Mundo
```

## 2.2. Uso de editor Ninja.

Cuando desarrollamos aplicaciones en un lenguaje, es útil contar con un entorno de desarrollo, que entre otras cosas nos comente rápidamente la posible fuente de errores que se pueda dar en nuestro código y nos ayude a organizar nuestro trabajo. Una herramienta muy útil es el editor para python “Ninja”, el cual podemos descargar de:

<http://ninja-ide.org/>



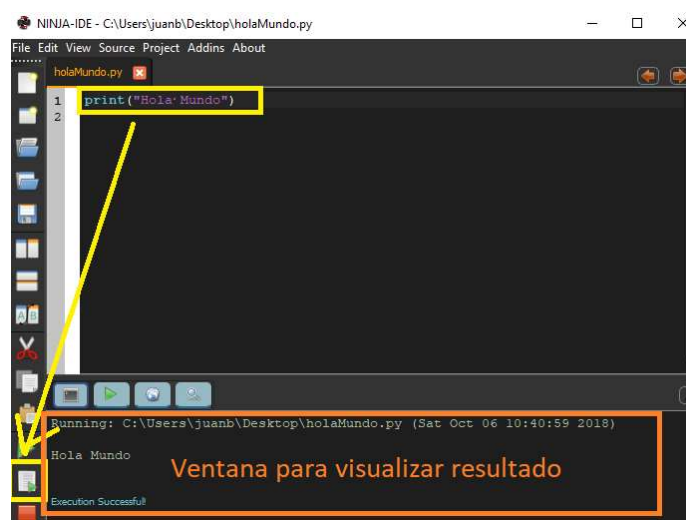
Es muy simple de utilizar y si bien más adelante veremos plataformas más completas, también serán más complejas, por lo que para no complicarnos a esta altura, no necesitaremos más que este editor para poder comenzar a aprender.

**Nota:** El alumno es libre de probar cualquier tipo de editor, como puede ser; Notepad++, SublimeText, Pycharm, Netbeans, etc.

La primera vez que se instala ninja, reconoce la versión de python que tenemos instalada en las variables de entorno y trabaja sobre esa versión, si quisiéramos cambiar la versión que utiliza, podríamos ir a Edit > Preferences y en la nueva ventana seleccionar la solapa "Execution", desde ahí podremos indicar donde se encuentra el ejecutable de la versión de python que queremos utilizar.



Para utilizarlo, abrimos el archivo desde "File" en el margen superior izquierdo o creamos uno nuevo y presionamos en el botón que parece una hoja de papel en blanco con un triángulito en verde y listo, el resultado de la aplicación nos figura en la ventana al pie de la aplicación en la cual se puede visualizar el resultado.

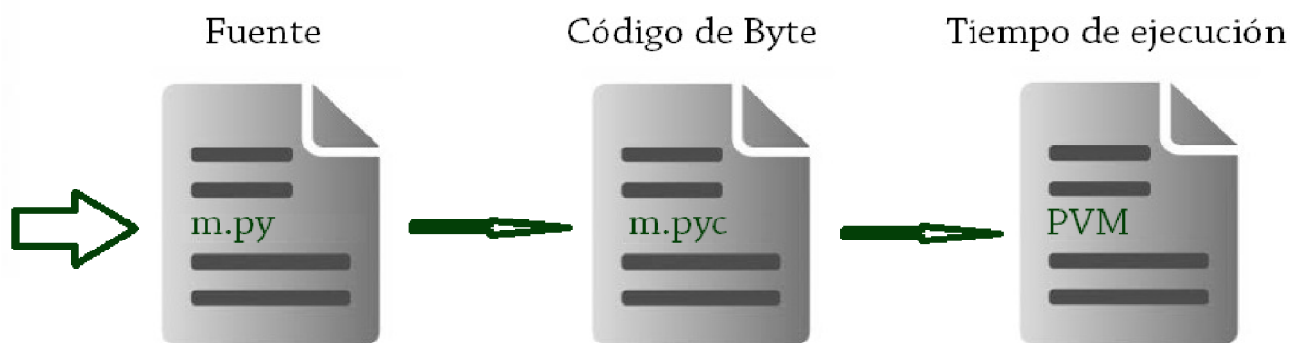




### 3. Como funciona Python

Python es tanto un lenguaje compilado como interpretado, en el cual:

- 1.- Python compila el código, o sea lo pasa a código de máquina, permitiendo que se ejecute más rápido.
- 2.- El código compilado se presenta con extensión .pyc con la estructura `_pycache_subdirectorio`. Una vez que se genere el archivo .pyc si no existe modificaciones se ejecuta este archivo en lugar del .py
- 3.- El intérprete luego interpreta el código de byte línea por línea.



Programming Python 5th Edition – Mark Lutz – O'Reilly 2013

**NOTA 1:** A diferencia de C o C++ el código corre inmediatamente luego de que es escrito, no existe la etapa de construcción.

**NOTA 2:** En Python la PVM (Python Virtual Machin), no el chip del CPU es el que realiza el papel de Intérprete por lo que Python no funciona tan rápido como C o C++. Por otro lado Python no re analiza cada línea de código, el efecto es que corre a una velocidad intermedia entre los lenguajes compilados y los interpretados.



## 4. Variables y comentarios.

### Variable

Una variable se puede considerar como un símbolo que puede ser reemplazado o que toma un valor determinado, como puede ser un valor numérico en una ecuación o expresión matemática en general.

Las variables se pueden utilizar para guardar datos de diferentes tipos, por ejemplo: enteros, caracteres, listas, arrays, diccionarios, objetos, etc. Todos estos términos seguramente son desconocidos a esta altura, e iremos hablando de ellos en el transcurso de las unidades. De esta forma si queremos hacer que una variable se encuentre relacionada con un valor entero podríamos escribir algo como:

```
variable1 = 7
```

En el ejemplo anterior el nombre de la variable es “variable1” y el valor que toma la variable es el entero = 7.

Pero si el 7 lo ponemos entre comillas simples o dobles en este caso la variable1 estaría relacionada con el carácter “7” o como lo trabajaremos en python el string (alfanuméricos) “7”.

```
variable1 = “7”
```

### Comentarios

En python podemos adicionar a nuestro código comentarios que podemos escribir en una línea anteponiendo el símbolo de numeral “#” o en varias líneas entre comillas triples como se muestra a continuación.

```
# Esto es un comentario
"""
Hola curso,
Esto también es un comentario pero multilínea.
"""
```

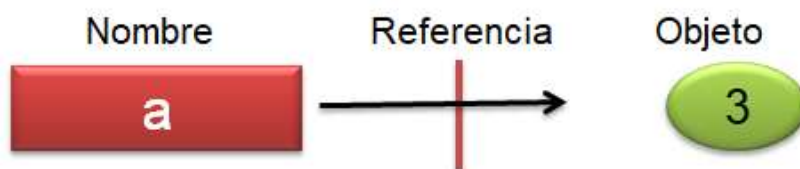


## 5. Asignación dinámica, garbage collection y referencias compartidas.

### 5.1. Asignación dinámica.

Cada vez que creamos una variable:

- 1 – Se genera un registro en una tabla
- 2 – Se crea un objeto
- 3 – Se establece la ruta desde la variable al objeto



El objeto creado posee:

- 1 – Un indicador de tipo de objeto.
- 2 – Un contador de referencia para establecer cuándo se puede reclamar el objeto.

Ejemplo:

La variable "a" cambia de tipo, de entero a cadena de caracteres y luego a un número flotante.

```
a = 3          # Es un entero
a = 'Manzana' # Ahora es un string
a = 1.23       # Ahora es un número flotante
```

En este caso la variable no tiene tipo, ya que este es un dato asociado al objeto , “a” simplemente está referenciando a diferentes objetos. Lo único que podemos decir de una variable es que ésta hace referencia a un determinado objeto en un determinado espacio de tiempo.





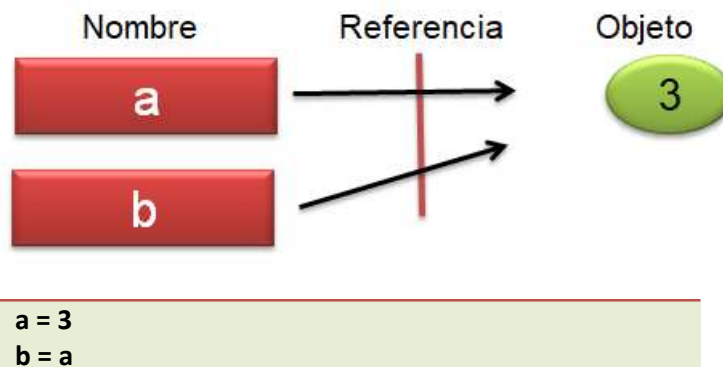
## 5.2. Garbage collection (Cuando se destruye un objeto)

En Python en el momento que un nombre (variable) es asignado a un nuevo objeto, el espacio de memoria ocupado por el nuevo objeto es reclamado si éste no es referenciado por ningún otro nombre u objeto.

```
x = 42
x = 'juan'      # Reclama 42
x = 3.1415      # Reclama juan
x = [1, 2, 3]   # Reclama 3.1415
```

## 5.3. Referencias compartidas

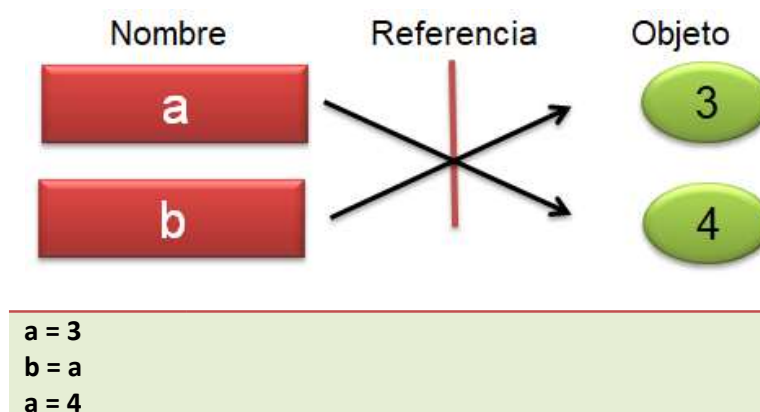
En este caso ambos nombres de variables, tanto “a” como “b” se encuentran asociados al mismo espacio de memoria



En Python, no hay forma de hacer que una variable haga referencia a otra variable, en lugar de esto la nueva variable, hace referencia al mismo objeto.

Ojo, pues según esto al modificar el valor de a, b sigue haciendo referencia al objeto asignado antes





A diferencia de otros lenguajes, en **Python las variables señalan siempre a objetos**, no a espacios de memoria. Al setear una variable con un nuevo valor el objeto original no es alterado, sino que hace referencia a un nuevo objeto.

Con este punto debemos tener especial cuidado y lo desarrollaremos un poco más en la siguiente unidad ya que existen situaciones que nos pueden conducir a un error de interpretación, al trabajar con los tipos de datos “Listas” y “Diccionarios”.

La siguiente tabla presenta algunos de los tipos de objetos que comenzaremos a analizar, debemos prestar especial atención a la primera columna en donde se indica si el objeto es inmutable o no. En la siguiente unidad desarrollaremos este tema, hasta aquí solo diremos que lo dicho antes se cumple para objetos del tipo inmutables.

Inmutalbe/N o Inmutable	Tipos de Objetos	Ejemplo
Inmutable	Números	1234, 3.14, 3+4j, 0b111, Decimal()
Inmutable	Strings	'Pera', b'a\x01c'
No inmutable	Listas	[1, ['tres', 'dos']], list(range(10))
No inmutable	Diccionarios	{'nombre': 'Pedro', 'edad':23, }, dict(horas=10)
Inmutable	Tuplas	(1, 'pera', 4), tuple('pera')
	Archivos	open('archivo.txt')
No inmutable	Sets	set('abc'), {'a', 'b', 'c'}
	Otros	Booleans, types, None

## 6.- GUI

Trabajar con el CMD puede ser muy útil en muchas ocasiones pero muy aburrido, por lo que vamos a introducir una interfaz gráfica llamada “Tkinter” la cual nos va a permitir crear aplicaciones de escritorio en Linux, Mac y Windows.

### 6.1. Primera aproximación – ejemplo1.py.

Comencemos por realizar un hola mundo para comprender un poco el funcionamiento. Lo primero que tenemos que hacer es incorporar la plataforma para lo cual incluiremos como primer línea la llamada a la librería:

```
from tkinter import *
```

La librería utiliza objetos propios que llamaremos widgets los cuales nos permiten agregar botones, cajas, entradas de texto, etc. Por ejemplo el widget “Label()” nos permite agregar texto dentro de la ventana emergente, por lo que un primer ejemplo podría ser:

#### **ejemplo1.py**

```
from tkinter import *  
Label(text='Hola').pack()  
mainloop()
```

Cada widget debe agregar el manejador de geometría pack() de forma de que los mismos se presenten en pantalla y el script debe finalizar con mainloop() el cual da inicio a los procesos de ejecución. Al ejecutar el script aparece en pantalla una ventana emergente con el contenido del atributo “text” del widget “Label”.





## 6.2. Algo un poco más interesante – ejemplo2.py.

Algo mucho más interesante podría ser presentar en pantalla el resultado de una ecuación, por ejemplo el resultado de multiplicar dos variables, pero en lugar de presentar el resultado como texto plano, indicarlo dentro de un campo del tipo entrada de texto editable. Veamos el código, la ejecución y el análisis.

### **ejemplo2.py**

```
from tkinter import *

root = Tk()
e = Entry(root)
e.pack()
e.focus_set()
a = 5
b = 2
c = a * b
var = IntVar()
e.config(textvariable=var)
var.set(c)

mainloop()
```

Al ejecutarlo nos daría:



Como podemos ver el resultado  $c = a * b$  aparece en una ventana emergente editable correspondiente al widget “Entry()” en el cual para agregarlo a la pantalla le pasamos como primer parámetro la ventana “root”, de esta forma estamos indicando que el campo debe posicionarse dentro de la ventana.

Para agregar el resultado dentro del campo creamos una variable con valor “IntVar()” el cual nos permite guardar en la variable un valor que “Tkinter” entiende como una variable del tipo entera y luego le seteamos el valor con “set()” utilizando lo que se denomina notación de punto para agregarle el resultado a la variable (hablaremos de este tipo de notación más adelante).



Finalmente nuevamente utilizamos la notación de punto para pasarle a la entrada el valor mediante el atributo “textvariable” y la rutina “config()”.

**Nota:** En las siguientes unidades comenzaremos a desarrollar el tema de notación de punto, rutinas, funciones, y tipos de datos.

## TRABAJO PARA LA APROBACIÓN DEL CURSO

Cada unidad presenta un trabajo a entregar, la aprobación del nivel inicial se realiza mediante la entrega de todos los trabajos propuestos (8 en total).

Todos los ejercicios serán explicados pasadas dos semanas de la fecha de entrega para darle tiempo a los alumnos a que los resuelvan por ellos mismos, salvo:

- El correspondiente a la primera unidad que será explicado luego de la primera semana.
- El correspondiente a la última unidad, el cual será tomado como entrega final.

## ACLARACIÓN

En los siguientes ejercicios se tomará como punto de partida el ejercicio “ejemplo2.py” de esta unidad. La finalidad no es aprender tkinter (esto lo comenzaremos a realizar de a poco a partir de la unidad 3) solo utilizaremos el formato para presentar la información en pantalla, por lo que la estructura en gris no debe ser modificada, solo se modificará lo que está en letras negras:

```
ejemplo2.py
```

```
from tkinter import *
```

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)



```
root = Tk()
e = Entry(root)
e.pack()
e.focus_set()
a = 5
b = 2
c = a * b
var = IntVar()
e.config(textvariable=var)
var.set(c)

mainloop()
```

### Comentario previo

Los strings o cadena de caracteres como veremos en más detalle en la unidad 2 se definen entre comillas, por ejemplo: `variable = "6"`, los mismos se concatenan utilizando el signo (+) (concatenar es representar un string a continuación del otro)

## Ejercicio 1

Para pasar de un string a un número entero se utiliza el método `int()`, como se muestra a continuación:

```
variable1 = "9"

variable2 = int(variable1)
```



Realice un programa con Tkinter que tome dos valores, uno entero y el otro un string, realice la suma como enteros y lo presente en pantalla.

## Ejercicio 2

Para pasar de un entero a un string se utiliza el método `str()`. Realice un programa con Tkinter que tome dos valores uno entero y el otro un string, concatene ambos como un string y presente el resultado en pantalla

## Ejercicio 3

La librería `random` la cual puede ser agregada en nuestros scripts de la siguiente manera:

```
import random
```

Permite obtener números aleatorios.

### Tarea 1:

Realice un programa partiendo del Ejercicio 1 que multiplique el resultado por un número aleatorio y lo presente en pantalla.

### Tarea 2:

Presente un ejemplo y explique el funcionamiento de cómo se utiliza el siguiente método:

```
random.randrange()
```

**Nota:** Podemos encontrar una referencia al método `randrange()` en la siguiente ruta del manual de python:

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148  
[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)



<https://docs.python.org/3/library/random.html?highlight=randrange#random.randrange>

Basándose solo en esta referencia trate de realizar el ejercicio propuesto. Aquí se le van a presentar varias preguntas que analizaremos juntos en la próxima clase virtual, como por ejemplo:

**¿Qué es un método?**

**¿Qué significa el punto en la expresión anterior?**

**NOTA:** Se pide que el alumno intente realizar ejercicios aun cuando no los pueda resolver completamente, anote todas las dudas para consultar en la próxima clase online y comparta sus dudas en la semana mediante el foro de la unidad.



## Bibliografía utilizada y sugerida

### Libros

Programming Python 5th Edition – Mark Lutz – O'Reilly 2013

Programming Python 4th Edition – Mark Lutz – O'Reilly 2011

### Manual online

<https://docs.python.org/3.7/tutorial/>





## Lo que vimos

En esta unidad realizamos la instalación de python y comenzamos a describir su funcionamiento.

---



## Lo que viene:

En la siguiente unidad comenzaremos a trabajar con algunos tipos de datos.