

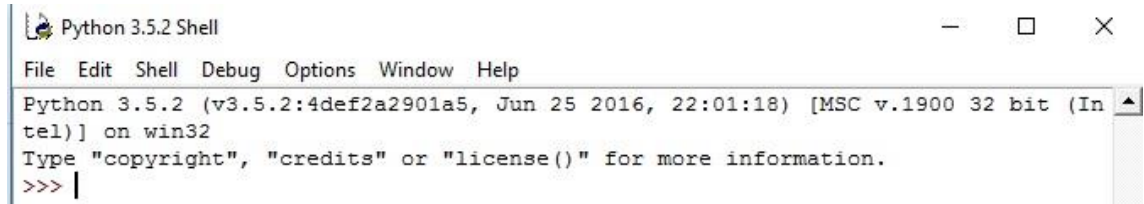


Trabajo Práctico N°1

Uso del IDLE

Ejercicio 1 – Probar IDLE

1. Abrir el Programa IDLE (Python 3.5).
2. Observar que aparece el prompt esperando ingreso de datos.



3. Ingresar una expresión matemática (Ej. 2+2).
4. Presionar Enter.
5. Observar el resultado arrojado.

Ejercicio 2 – Operaciones Algebraicas

1. Abrir el Programa IDLE (Python 3.5).
2. Observar que aparece el prompt esperando ingreso de datos.
3. Solicitar al IDLE que calcule la resta de 2 números.
4. Presionar Enter
5. Observar el resultado arrojado.
6. Haga lo mismo con la multiplicación (*) y división (/).

Ejercicio 3 – Cálculo de módulo

1. Abrir el Programa IDLE (Python 3.5).
2. Observar que aparece el prompt esperando ingreso de datos.
3. Solicitar al IDLE que calcule el módulo de 2 números.
4. Dicha operación se realiza con el operador %.
5. Presionar Enter
6. Observar el resultado arrojado.

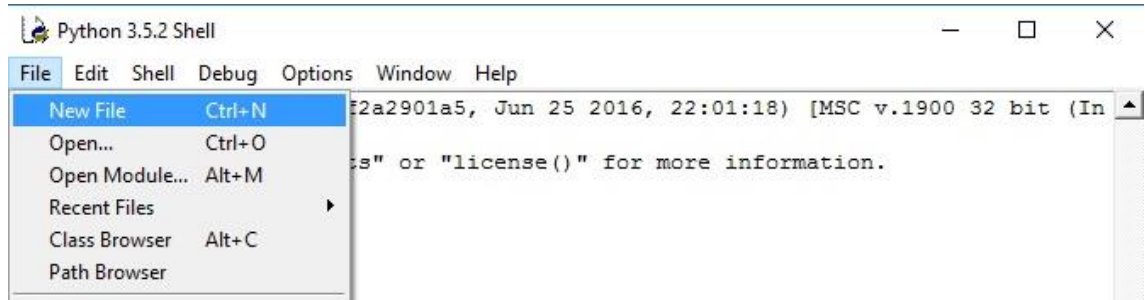
Ejercicio 4 – Cálculo de Potencia

1. Abrir el Programa IDLE (Python 3.5).
2. Observar que aparece el prompt esperando ingreso de datos.
3. Solicitar al IDLE que calcule la potencia de un número elevado a otro (Ej. 2 al cubo).
4. Dicha operación se realiza con el operador ** (doble asterisco).
5. Presionar Enter
6. Observar el resultado arrojado.

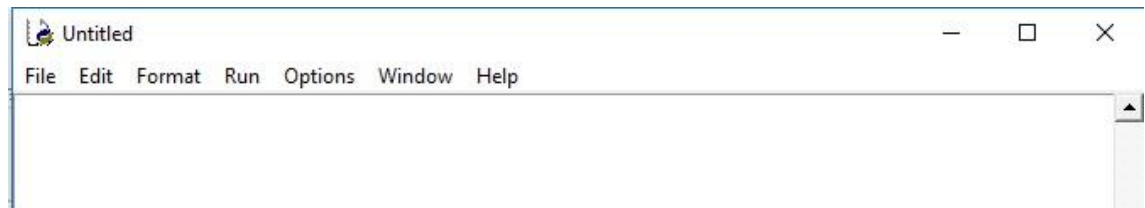
Lectura y Escritura

Ejercicio 6 – Programa saludo

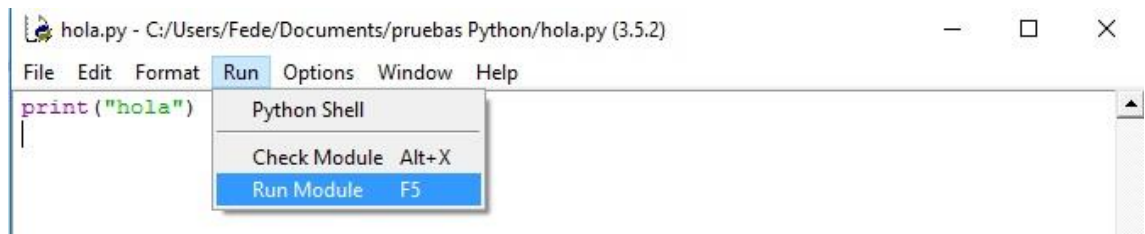
1. Abrir el Programa IDLE (Python 3.5).
2. Observar que aparece el prompt esperando ingreso de datos.
3. Ingresar al menú **File** → **New File** o **Ctrl+N**



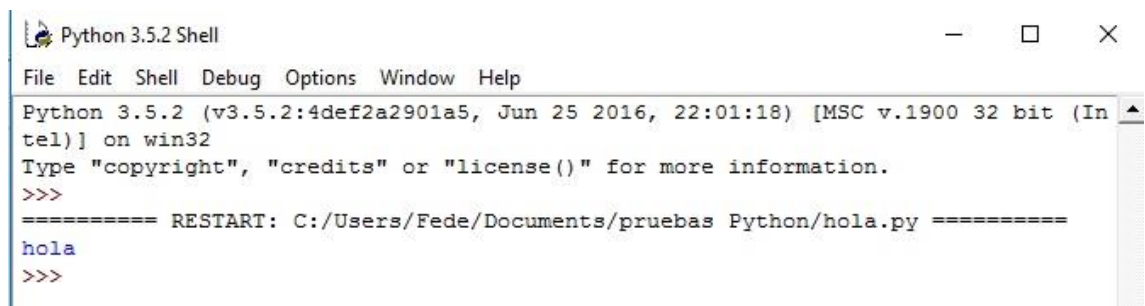
4. Se abrirá una nueva pantalla en blanco



5. Dentro de esa nueva pantalla agregue la siguiente línea de código escribir Hola a la salida estándar.
`print("Hola")`
6. Guarde el código del programa con **File → Save o Ctrl+S** con el nombre Hola.py (recuerde que el programa debe tener extensión .py o Tipo "Python Files", de lo contrario no funcionará).
No se olvide de seleccionar la carpeta donde desee que se guarde el archivo (**Recomendación:** Modificar la que aparece por defecto).
7. Ejecute el programa con **F5** o con **Run → Run Module**



8. Se abrirá nuevamente la pantalla principal del IDLE (donde figura el prompt >>>).
9. A continuación se mostrará la salida de su programa de color azul.



Ejercicio 7 – Programa Intercambio

1. El objetivo del programa es intercambiar los valores de dos variables.
2. Cree un nuevo archivo, llamada Intercambio.



3. Dentro del cuerpo del programa agregue las siguientes líneas para declarar 2 variables de tipo numérico.

```
a = 4  
b = 7
```
4. Para intercambiar los valores, es necesario una variable auxiliar c

```
c = a
```
5. Agregue la siguiente línea para mostrar el valor de a

```
print(A)
```
6. Agregue otra línea para mostrar el valor de b
7. Intercambie los valores con las siguientes líneas

```
a = b;  
b = c;
```
8. Muestre nuevamente los valores de a y b para comprobar que están intercambiados.
9. Guarde el archivo con el nombre intercambio.py.
10. Ejecute.

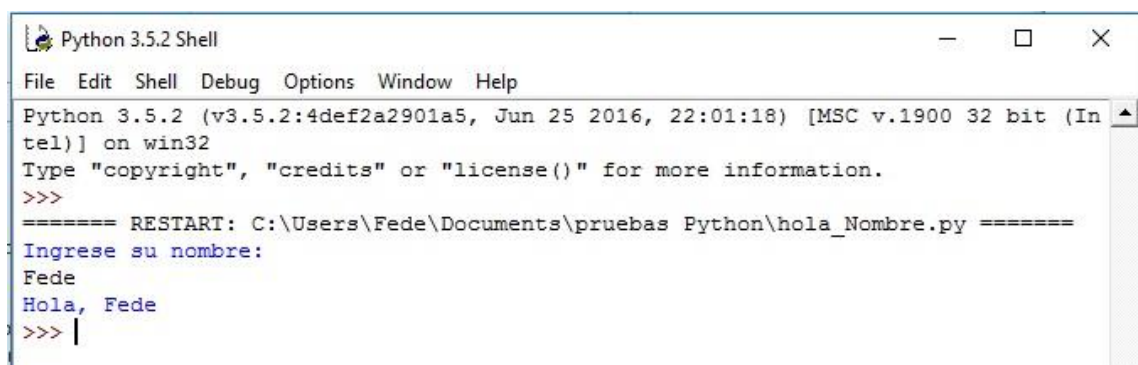
Ejercicio 8 – Programa Saludo

1. El objetivo del programa es leer el nombre del usuario y luego mostrar un saludo personalizado.
2. Cree un nuevo archivo, llamado saludo.
3. Dentro del nuevo archivo, agregue la siguiente línea que permite al programa leer desde la entrada estándar (teclado).

```
print("Ingrese su nombre:")
```
4. Esa instrucción muestra por pantalla la solicitud al usuario que ingrese su nombre.
5. Guarde lo ingresado por el usuario en una variable con la siguiente línea:

```
nombre=input()
```
6. Muestre el saludo al usuario con el siguiente código

```
print("Hola, ", nombre)
```
7. Guarde el archivo con el nombre saludo.py
8. El programa mostrará el saludo personalizado.
9. Ejecute.
10. En la ventana principal del programa se podrá observar lo siguiente:



```
Python 3.5.2 Shell  
File Edit Shell Debug Options Window Help  
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: C:\Users\Fede\Documents\pruebas Python\hola_Nombre.py =====  
Ingrese su nombre:  
Fede  
Hola, Fede  
>>> |
```

Ejercicio 9 – Nombre y Apellido

1. El objetivo del programa es leer el nombre y el apellido del usuario y luego mostrar un saludo personalizado.
2. Cree un nuevo archivo, llamado nombreApellido.
3. Dentro del nuevo archivo, agregue la siguiente línea que permite al programa leer desde la entrada estándar (teclado).

```
print("Ingrese su nombre:")
```



4. Esa instrucción muestra por pantalla la solicitud al usuario que ingrese su nombre.
5. Guarde lo ingresado por el usuario en una variable llamada nombre con la siguiente línea:
`nombre=input()`
6. Solicite ahora que ingrese el apellido.
7. Guárdelo en una variable llamada apellido.
8. Muestre el saludo al usuario con el siguiente código
`print("Hola, ", nombre, apellido)`
9. Guarde el archivo con el nombre nombreApellido.py
10. El programa mostrará el saludo personalizado.
11. Ejecute.
12. En la ventana principal del programa se podrá observar lo siguiente:

```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/Fede/Documents/pruebas Python/hola_Nombre_apellido.py ==
Ingrese su nombre:
Fede
Ingrese su apellido:
Brest
Hola, Fede Brest
>>> |
```

Aritmética

Ejercicio 10 – Suma

1. El objetivo del programa es solicitar al usuario dos números y calcular su suma
2. Cree un nuevo archivo, llamado suma.
3. Dentro del archivo agregue una línea para mostrar desde pantalla.
4. Solicite al usuario que ingrese un número:
`print("ingrese un número: ")`
5. Lea la entrada del usuario con el siguiente código, que guarda el valor entero en una variable x
6. De forma predeterminada, la función `input()` convierte la entrada en una cadena. Si se quiere que Python interprete la entrada como un número entero, se debe utilizar la función `int()` de la siguiente manera:
`numero1 = int(input())`
7. Solicite al usuario el segundo número
8. Lea la entrada del usuario y guárdela en otra variable y de tipo int
9. Calcule la suma de ambos números y guárdela en una nueva variable resultado de tipo int mediante la siguiente línea:
`suma = (numero1+numero2)`
10. Muestre el valor de resultado.
11. Guarde el archivo.
12. Compile y ejecute el programa.
13. El resultado final mostrado será:



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Fede/Documents/pruebas Python/suma.py =====
ingrese un número:
1
ingrese otro número:
2
La suma de 1 y 2 es 3
>>> |
```

Ejercicio 11 – División decimal

1. El objetivo del programa es solicitar al usuario dos números decimales y calcular su división
2. Cree un nuevo archivo, llamado division.
3. Dentro del archivo agregue una línea para mostrar desde pantalla.
4. Solicite al usuario que ingrese un número:
`print("ingrese un número: ")`
5. Lea la entrada del usuario con el siguiente código, que guarda el valor entero en una variable x
6. De la misma manera que el ejercicio anterior, para que Python interprete la entrada como un número decimal, se debe utilizar la función `float()` de la siguiente manera:
`numero1 = float(input())`
7. Solicite al usuario el segundo número
8. Lea la entrada del usuario y guárdela en otra variable y de tipo float.
9. Calcule la división de ambos números y guárdela en una nueva variable resultado de tipo float mediante la siguiente línea:
`división = (numero1/numero2)`
10. Muestre el valor de resultado.
11. Guarde el archivo.
12. Compile y ejecute el programa.
13. El resultado final mostrado será:

Ejercicio 12 – Conversión de Grados

1. El objetivo del programa es realizar un programa que pida una temperatura en grados Celsius y que escriba esa temperatura en grados Fahrenheit.
Se recuerda que la relación entre grados Celsius (C) y grados Fahrenheit (F) es la siguiente: **$F = 1,8 * C + 32$**
2. Cree un nuevo archivo, llamado conversion.
3. Dentro del archivo agregue una línea para mostrar desde pantalla.
4. Ahora ahorraremos líneas de código y unificaremos mostrar por pantalla, el ingreso y la conversión de tipo.
5. Solicite al usuario que ingrese un número decimal, lea la entrada del usuario y realice a conversión mediante la siguiente línea:
`celsius = float(input("Escriba una temperatura en grados Celsius: "))`
9. Calcule la conversión y guárdela en una nueva variable resultado de tipo float mediante la siguiente línea:
`fahrenheit = 1.8*celsius + 32`
10. Muestre el valor de resultado.
11. Guarde el archivo.
12. Compile y ejecute el programa.



Cadenas de Texto

Ejercicio 13 – Longitud de cadena de texto

1. El objetivo del programa es mostrar la longitud de un texto ingresado por el usuario.
2. Cree un nuevo archivo .py, llamado LongitudCadena
3. Solicite al usuario que ingrese un texto cualquiera (puede ser más de una palabra) y guárdelo en una variable llamada texto.
4. Agregue la siguiente línea para calcular la longitud del texto, utilizando la función len() y guárdelo en una variable llamada longitud
`longitud=len(texto)`
5. Muestre la longitud de la cadena por pantalla.
6. Guarde el archivo.

Ejercicio 14 – Cortar cadena

1. El objetivo del programa es cortar por la mitad una cadena de texto ingresada por el usuario.
2. Agregue un nuevo archivo llamado CortarCadena
3. Solicite al usuario que ingrese un texto y guárdelo en una variable llamada texto,
4. Obtenga la longitud de la cadena con la función len() y guárdelo en una variable llamada longitud.
5. Obtenga la mitad de la longitud dividiendo por 2 el valor obtenido en el paso anterior y guárdelo en una variable llamada mitad.
6. Podemos cortar una cadena en partes colocando corchetes luego del nombre de la variable, indicando la posición por la que queremos realizar el corte. El primer argumento es el inicio del corte y el segundo argumento es la posición donde termina el corte.
7. Agregue el siguiente código para cortar la cadena por la mitad y almacenar esa mitad en una variable llamada subtexto.
`subtexto=texto[0:mitad]`
- 8.
9. Agregue el siguiente código para cortar la segunda mitad de la cadena
`subtexto=texto[mitad:longitud]`
10. Muestre ambas mitades por separado.
11. Guarde el archivo.
12. Pruebe con cadenas de diferente longitud.