

Clase Número: 1

Introducción de la materia

Este curso esta pensado para ser cursado por una persona que no conoce nada de programación y se emplea el lenguaje Python como primer lenguaje para acercarse a la programación de computadoras.

El objetivo fundamental de este tutorial es permitir que el estudiante pueda resolver problemas de distinta índole (matemáticos, administrativos, juegos, contables etc.) empleando como herramienta la computadora.

Hay que tener en cuenta que para llegar a ser un programador se debe recorrer un largo camino donde cada tema es fundamental para conceptos futuros. Es importante no dejar temas sin entender y relacionar.

La programación a diferencia de otras materias que ha estudiado como podría ser la historia requiere un estudio metódico y ordenado (en historia se puede estudiar la edad media sin tener grandes conocimientos de la edad antigua, esto no pasa con el aprendizaje de la programación)

Es bueno tenerse paciencia cuando los problemas no se resuelven por completo, pero es de fundamental importancia dedicar tiempo al análisis individual de los problemas.

¿Qué es un programa?

Programa: Conjunto de instrucciones que entiende una computadora para realizar una actividad.

Todo programa tiene un objetivo bien definido: un procesador de texto es un programa que permite cargar, modificar e imprimir textos, un programa de ajedrez permite jugar al ajedrez contra el ordenador u otro contrincante humano.

La actividad fundamental del programador es resolver problemas empleando el ordenador como herramienta fundamental.

Para la resolución de un problema hay que plantear un algoritmo.

Algoritmo: Son los pasos a seguir para resolver un problema.

Diagrama de flujo

Un diagrama de flujo es la representación gráfica de un ALGORITMO.

Los símbolos gráficos principales a utilizar para el planteo de diagramas de flujo son:



Inicio y fin del diagrama.



Entrada de Datos

(Vamos a considerar que las entradas de datos se realizan siempre por el teclado de la computadora).

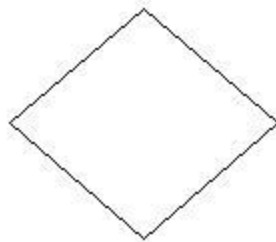


Salida de Datos

(Vamos a considerar que las salidas de datos se realizan siempre por la pantalla de la computadora).



Operación



Condición

Estos son los elementos gráficos esenciales para el desarrollo de un diagrama de flujo.

Planteo de un problema utilizando un diagrama de flujo.

Para plantear un diagrama de flujo debemos tener muy en claro el problema a resolver.

Ejemplo : Calcular el sueldo mensual de un operario conociendo la cantidad de horas trabajadas y el pago por hora.

Podemos identificar:

Datos conocidos:

Horas trabajadas en el mes.

Pago por hora.

Proceso:

Cálculo del sueldo multiplicando la cantidad de horas por el pago por hora.

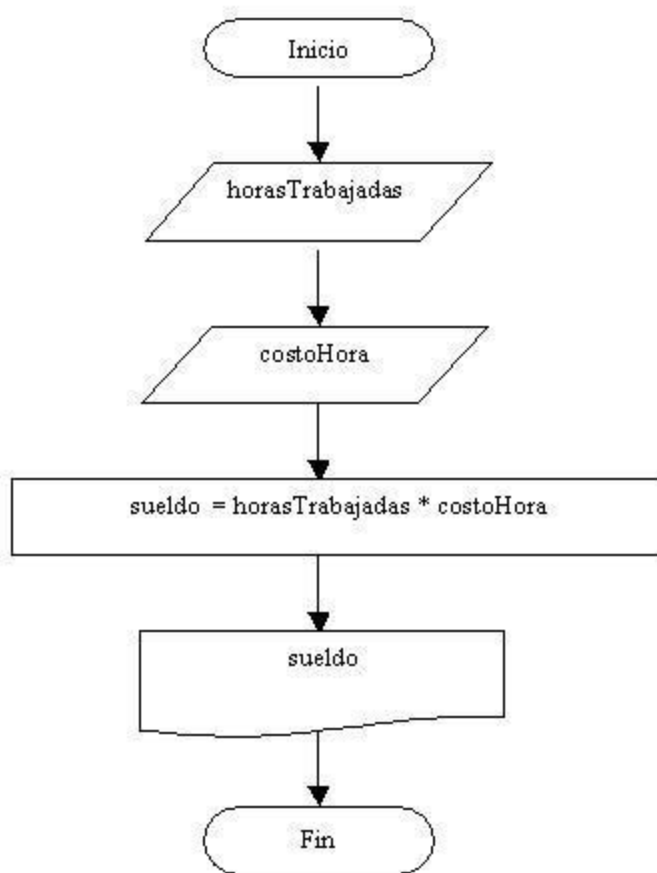
Información resultante:

Sueldo mensual.

Si hacemos un análisis, todo problema está constituido por:

- Datos conocidos: Datos con los que se cuenta al plantear el problema.
- Proceso: Operaciones a realizar con los datos conocidos.
- Información resultante: Es la información que se obtiene del proceso y nos permite resolver el problema.

Esta forma de expresar un problema identificando sus datos conocidos, procesos e información resultante puede llegar a ser engorrosa para problemas complejos donde hay muchos datos conocidos y procesos. Es por eso que resulta mucho más efectivo representar los pasos para la resolución del problema mediante un diagrama de flujo.



Resulta mucho más fácil entender un gráfico que un texto.

El diagrama de flujo nos identifica claramente los datos de entrada, operaciones y datos de salida.

En el ejemplo tenemos dos datos de entrada: horasTrabajadas y costoHora, a las entradas las representamos con un paralelogramo y hacemos un paralelogramo por cada dato de entrada.

La operación se representa con un rectángulo, debemos hacer un rectángulo por cada operación. A la salida la representamos con la hoja rota.

El diagrama de flujo nos da una idea del orden de ejecución de las actividades en el tiempo. Primero cargamos los datos de entrada, luego hacemos las operaciones necesarias y por último mostramos los resultados.

Variable: Es un depósito donde hay un valor. Consta de un nombre y pertenece a un tipo de dato.

Para el ejemplo planteado la variable `horasTrabajadas` almacena la cantidad de horas trabajadas por el operario. La variable `valorHora` almacena el precio de una hora de trabajo. La variable `suelo` almacena el sueldo a abonar al operario.

En el ejemplo tenemos tres variables.

Tipos de variable.

Una variable puede almacenar:

- Valores Enteros (100, 260, etc.)
- Valores Reales (1.24, 2.90, 5.00, etc.)
- Cadenas de caracteres ("Juan", "Compras", "Listado", etc.)

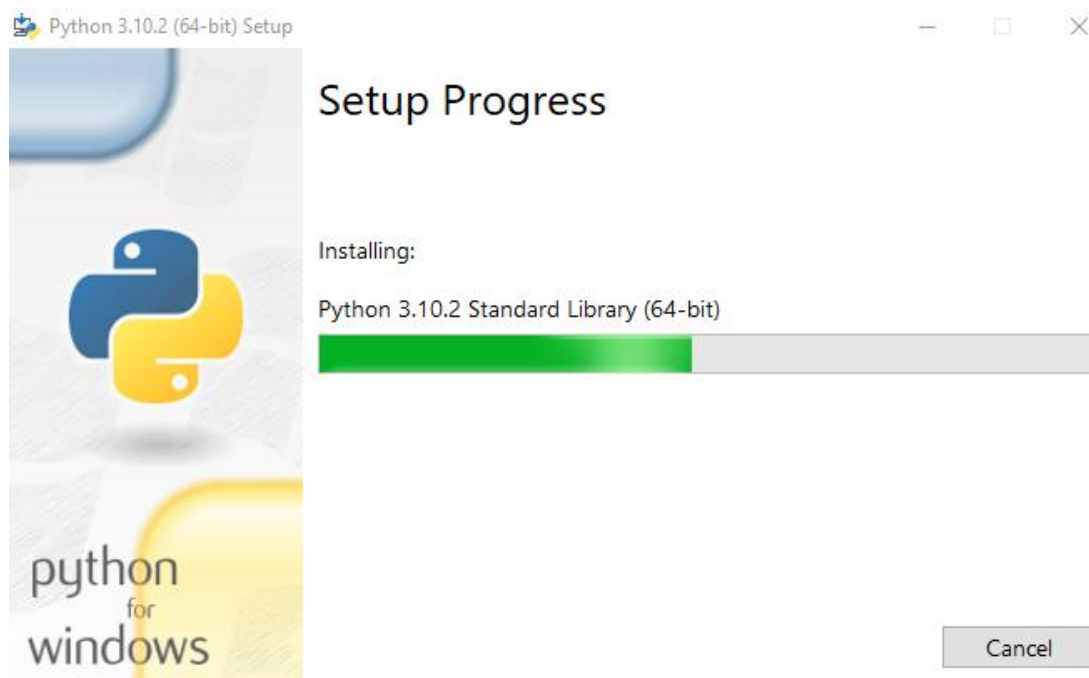
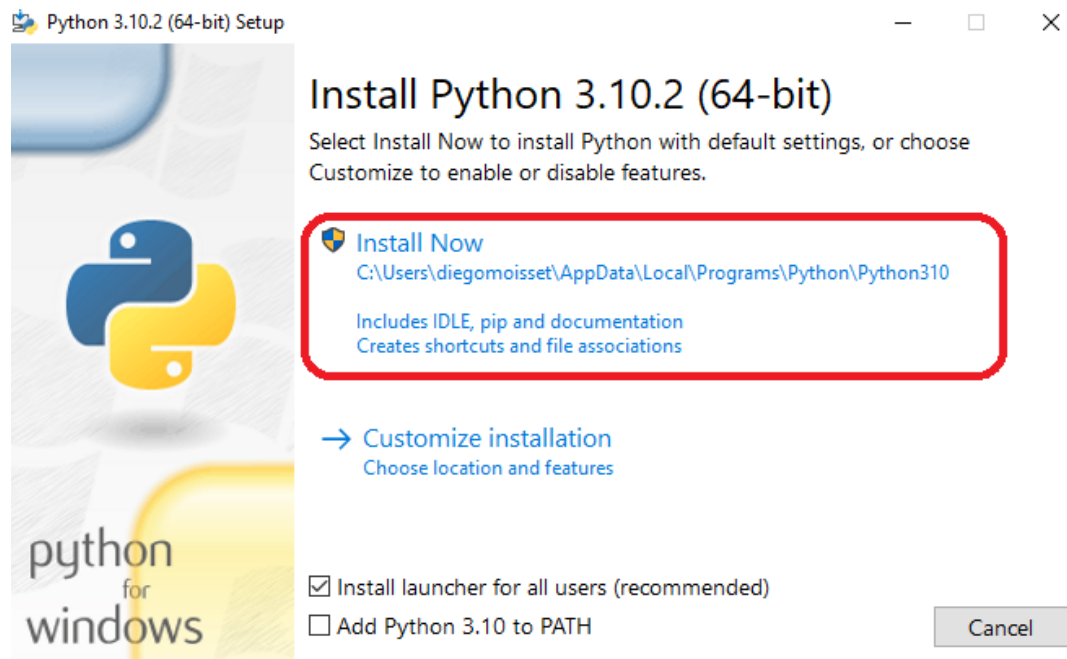
Elección del nombre de una variable:

Debemos elegir nombres de variables representativas. En el ejemplo el nombre `horasTrabajadas` es lo suficientemente claro para darnos una idea acabada sobre su contenido. Podemos darle otros buenos nombres. Otros no son tan representativos, por ejemplo `hTr`. Posiblemente cuando estemos resolviendo un problema dicho nombre nos recuerde que almacenamos las horas trabajadas por el operario pero cuando pase el tiempo y leamos el diagrama probablemente no recordemos ni entendamos qué significa `hTr`

Instalación de Python

Para la descarga del lenguaje Python lo hacemos del sitio: python.org (descargar la versión más actual 3.7)

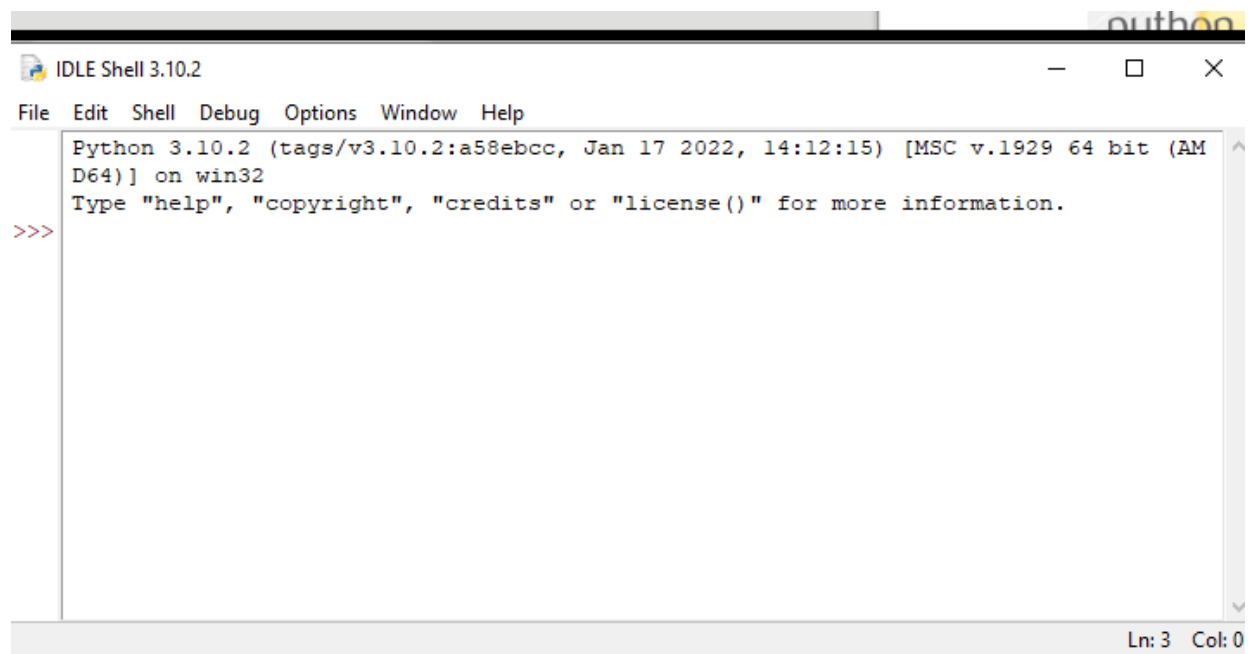
Ejecutamos el programa que descargamos y procedemos a instalarlo (marcamos 'Add Python3.7 to PATH'):



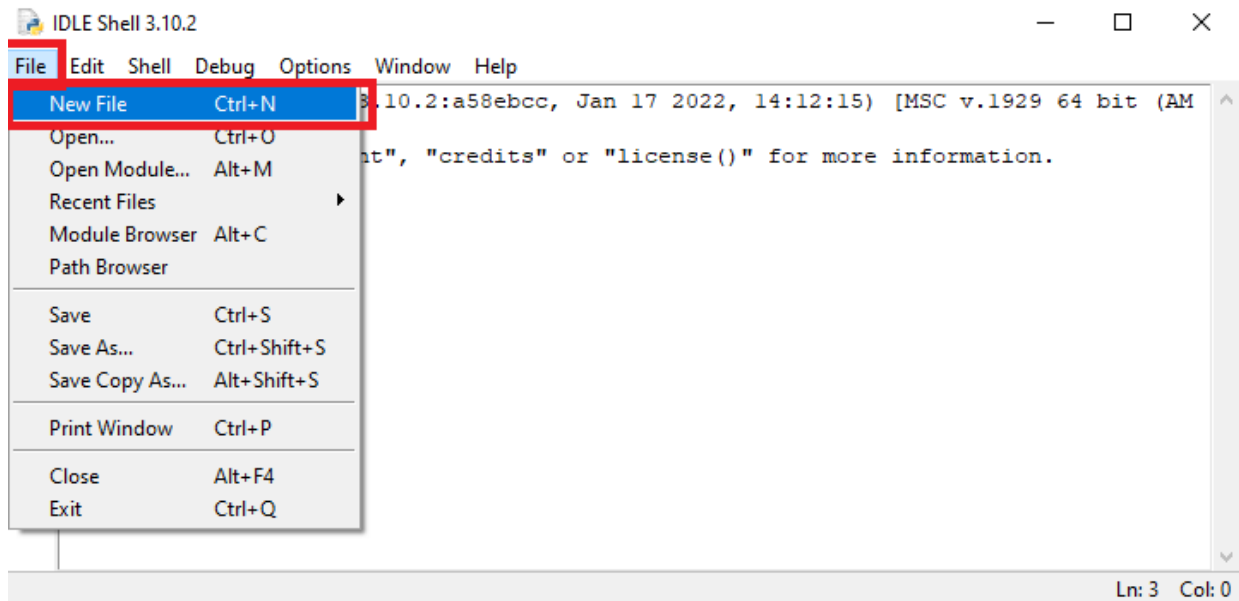
Luego de haberse instalado podemos ejecutarlo desde el menú de opciones de Windows:



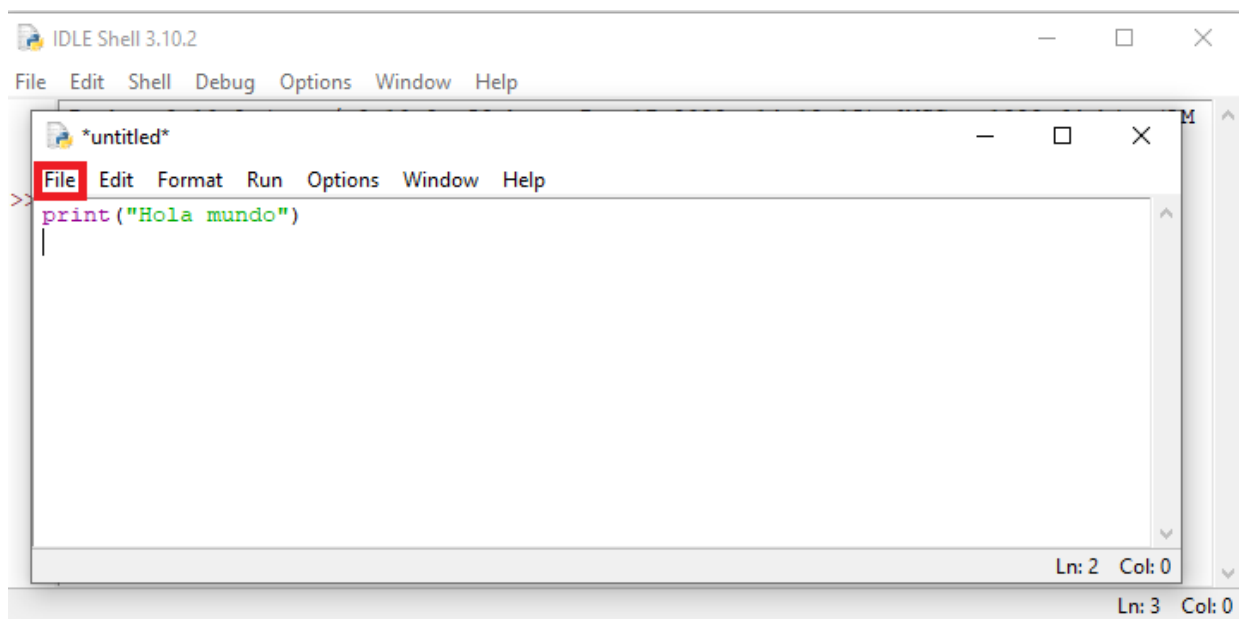
Una vez que iniciamos el Python aparece la siguiente ventana:



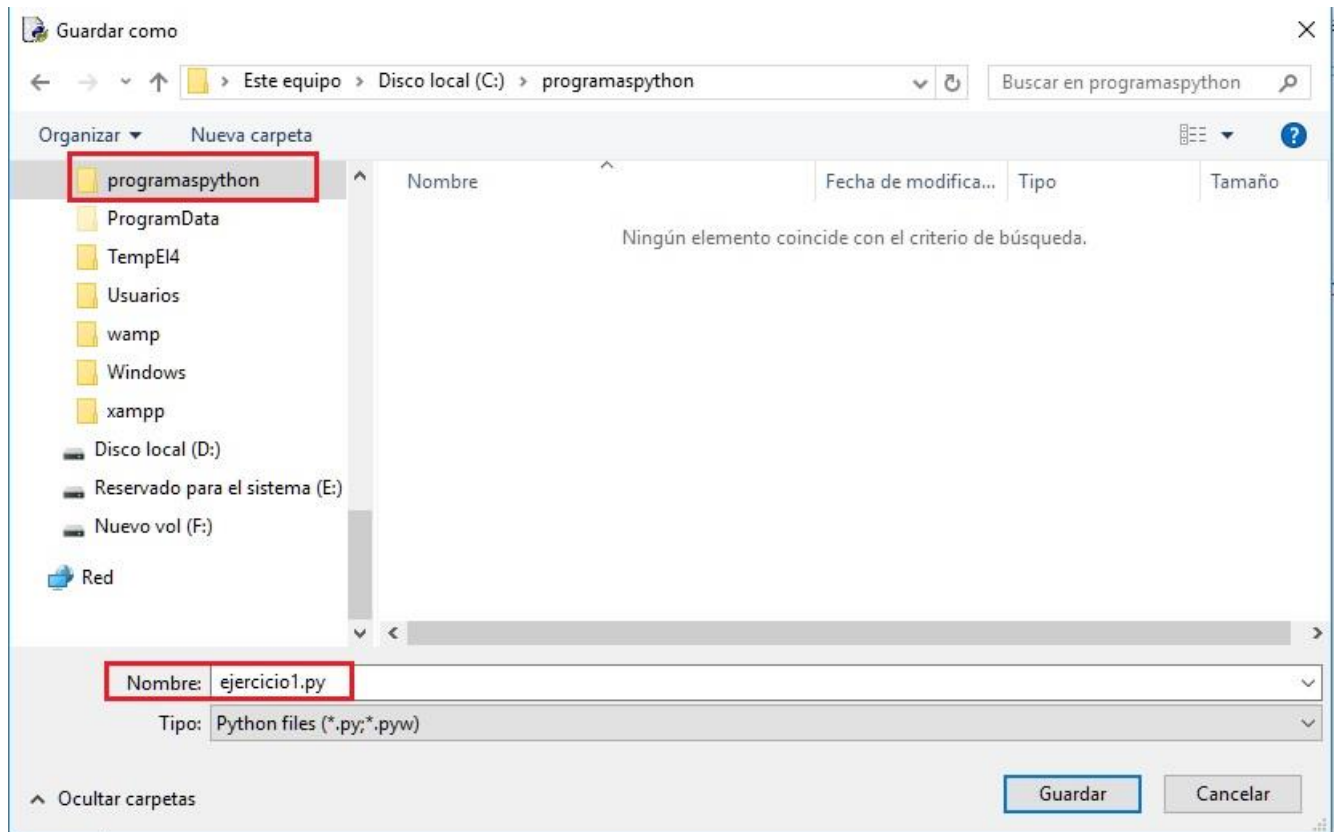
Para crear nuestro primer programa debemos seleccionar desde el menú de opciones "File"->"New File":



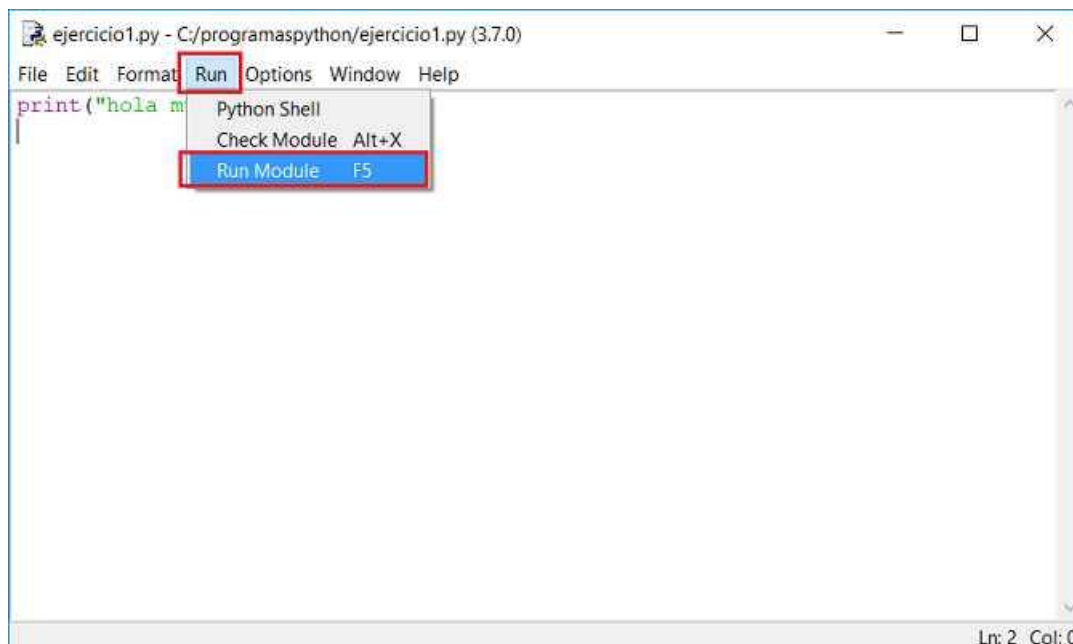
En esta nueva ventana debemos codificar nuestro programa en Python, el programa mínimo es mostrar un mensaje por pantalla:



Procedemos a guardar en el "disco duro" el primer programa en Python seleccionando la opción "File"->"Save", creamos un directorio donde almacenaremos cada uno de nuestros ejercicios que desarrollaremos durante este tutorial:



Para ejecutar el programa que codificamos debemos seleccionar desde la ventana de nuestro editor la opción "Run" -> "Run Module":



Podemos ver en otra ventana el resultado de la ejecución de nuestro primer programa en Python:


```
programa1.py - C:/basura2/programa1.py (3.10.2)
File Edit Format Run Options Window Help
print("Hola mundo")

IDLE Shell 3.10.2
File Edit Shell Debug Options Window Help
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/basura2/programa1.py =====
Hola mundo
>>>
```

Configuración del diagrama de flujo en Python

El diagrama de flujo es nuestra herramienta para poder plantear una solución a nuestro problema.

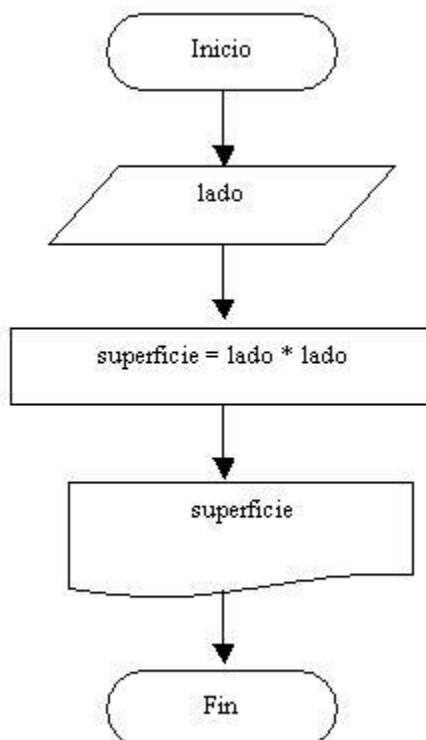
Para poder probar nuestra solución propuesta a un problema mediante un diagrama de flujo lo debemos codificar seguidamente en Python.

Empezaremos con problemas muy sencillos y procederemos a implementar su diagrama de flujo y seguidamente su codificación.

Problema 1

Hallar la superficie de un cuadrado conociendo el valor de un lado.

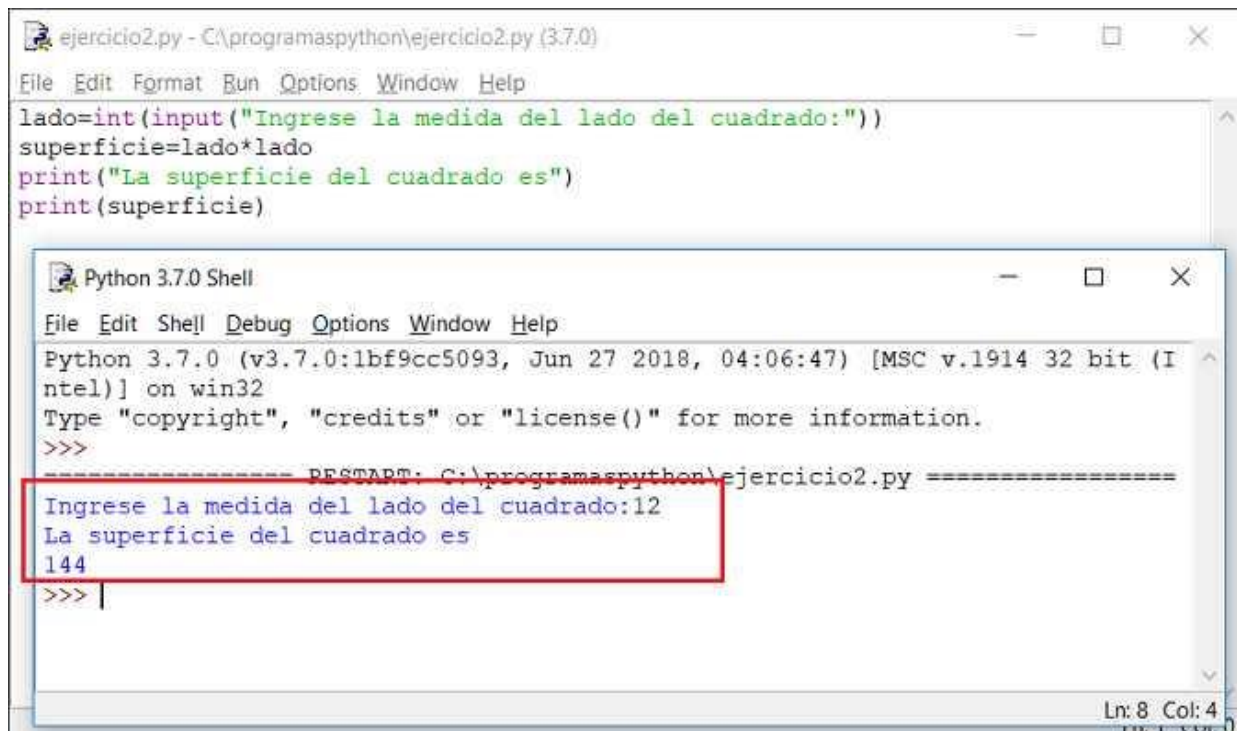
Diagrama de flujo:



Empleando un editor de texto (VSCode – Sublime Text, Notepad++ etc.) de Python procedemos a crear nuestro primer programa "ejercicio1.py", lo grabamos en el disco duro y codificamos en Python la siguiente solución al diagrama de flujo:

```
lado=input("Ingrese la medida del lado del cuadrado:")
lado=int(lado)
superficie=lado*lado
print("La superficie del cuadrado es")
print(superficie)
```

Si ejecutamos el programa "Run" -> "Run Module" podemos comprobar que se solicita el ingreso por teclado de la medida del lado del cuadrado y seguidamente nos muestra la superficie dependiendo del valor ingresado:



```
ejercicio2.py - C:\programaspython\ejercicio2.py (3.7.0)
File Edit Format Run Options Window Help
lado=int(input("Ingrese la medida del lado del cuadrado:"))
superficie=lado*lado
print("La superficie del cuadrado es")
print(superficie)

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
----- RESTART: C:\programaspython\ejercicio2.py -----
Ingrese la medida del lado del cuadrado:12
La superficie del cuadrado es
144
>>> |
```

Para el ingreso de un dato por teclado y mostrar un mensaje se utiliza la función `input`, esta función retorna todos los caracteres escritos por el operador del programa:

```
lado=input("Ingrese la medida del lado del cuadrado:")
```

La variable `lado` guarda todos los caracteres ingresados pero no en formato numérico, para esto debemos llamar a la función `int`:

```
lado=int(lado)
```

Ahora se vuelve a guardar en la variable `lado` el valor que ingresó el operador pero en formato entero que posibilita hacer operaciones matemáticas con el mismo.

Un formato simplificado para ingresar un valor entero por teclado y evitarnos escribir las dos líneas anteriores es:

```
lado=int(input("Ingrese la medida del lado del cuadrado:"))
```

Procedemos a efectuar el cálculo de la superficie luego de ingresar el dato por teclado y convertirlo a entero:

```
superficie=lado*lado
```

Para mostrar un mensaje por pantalla tenemos la función `print` que le pasamos como parámetro una cadena de caracteres a mostrar que debe estar entre simple o doble comillas:

```
print("La superficie del cuadrado es")
```

Para mostrar el contenido de la variable superficie no debemos encerrarla entre comillas cuando llamamos a la función print:

```
print(superficie)
```

Algunas consideraciones

Python es sensible a mayúsculas y minúsculas, no es lo mismo llamar a la función input con la sintaxis: Input.

Los nombres de variables también son sensibles a mayúsculas y minúsculas. Son dos variables distintas si en un lugar iniciamos a la variable "superficie" y luego hacemos referencia a "Superficie"

Los nombres de variable no pueden tener espacios en blanco, caracteres especiales y empezar con un número.

Todo el código debe escribirse en la misma columna, estará incorrecto si escribimos:

```
lado=input("Ingrese la medida del lado del cuadrado:")
lado=int(lado)
superficie=lado*lado
print("La superficie del cuadrado es")
print(superficie)
```

Hay más restricciones que iremos aprendiendo a medida que avance el curso.

Errores sintácticos y lógicos

Modificaremos el problema anterior y agregaremos adrede una serie de errores tipográficos. Este tipo de errores siempre son detectados por el intérprete de Python, antes de ejecutar el programa.

A los errores tipográficos, como por ejemplo indicar el nombre incorrecto de la función, nombres de variables incorrectas, falta de paréntesis, palabras claves mal escritas, etc. los llamamos errores SINTACTICOS.

Un programa no se puede ejecutar por completo sin corregir absolutamente todos los errores sintácticos.

Existe otro tipo de errores llamados ERRORES LOGICOS. Este tipo de errores en programas grandes (miles de líneas) son más difíciles de localizar. Por ejemplo un programa que permite hacer la facturación pero la salida de datos por impresora es incorrecta.

Problema 1

Hallar la superficie de un cuadrado conociendo el valor de un lado.

Programa correctamente codificado:

```
lado=int(input("Ingrese la medida del lado del cuadrado:"))
superficie=lado*lado
print("La superficie del cuadrado es")
print(superficie)
```

Programa con un error sintáctico:

```
lado=int(input("Ingrese la medida del lado del cuadrado:"))
superficie=lado*lado
print("La superficie del cuadrado es")
print(Superficie)
```

Es incorrecto la impresión de una variable nunca inicializada: "Superficie" (debemos respetar como la iniciamos en las líneas anteriores)

```
ejercicio2.py - C:\programaspython\ejercicio2.py (3.7.0)
File Edit Format Run Options Window Help
lado=int(input("Ingrese la medida del lado del cuadrado:"))
superficie=lado*lado
print("La superficie del cuadrado es")
print(Superficie)

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Ingrese la medida del lado del cuadrado:12
La superficie del cuadrado es
144
>>>
===== RESTART: C:\programaspython\ejercicio2.py =====
Ingrese la medida del lado del cuadrado:10
La superficie del cuadrado es
Traceback (most recent call last):
  File "C:\programaspython\ejercicio2.py", line 4, in <module>
    print(Superficie)
NameError: name 'Superficie' is not defined
>>>
```

Programa con un error lógico:

```
lado=int(input("Ingrese la medida del lado del cuadrado:"))
superficie=lado*lado*lado
print("La superficie del cuadrado es")
print(superficie)
```

Como podemos observar si ejecutamos el programa no presenta ningún error sintáctico, pero luego de ingresar el valor del lado del cuadrado (por ejemplo el valor 10) obtenemos como resultado un valor incorrecto (imprime el 1000), esto debido que definimos incorrectamente la fórmula para calcular la superficie del cuadrado:

```
superficie=lado*lado*lado
```

Estructuras de programación secuencial

Cuando en un problema sólo participan operaciones, entradas y salidas se la denomina una estructura secuencial.

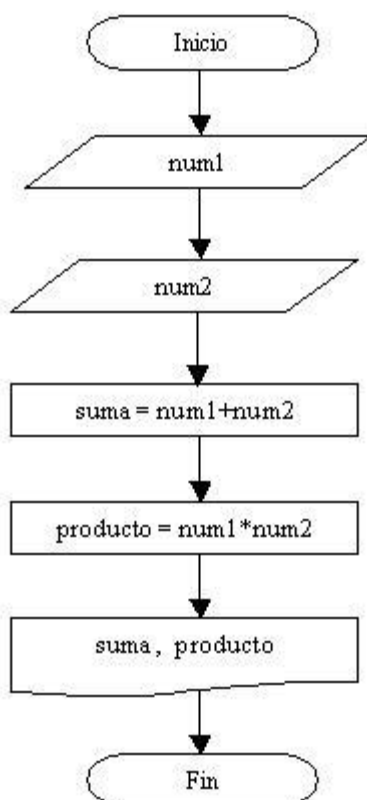
Los problemas diagramados y codificados previamente emplean solo estructuras secuenciales.

La programación requiere una práctica ininterrumpida de diagramación y codificación de problemas.

Problema 2:

Realizar la carga de dos números enteros por teclado e imprimir su suma y su producto.

Diagrama de flujo:



Tenemos dos entradas num1 y num2, dos operaciones: la suma y el producto de los valores ingresados y dos salidas, que son los resultados de la suma y el producto de los valores ingresados. En el símbolo de impresión podemos indicar una o más salidas, eso queda a criterio del programador, lo mismo para indicar las entradas por teclado.

Programa: ejercicio2.py

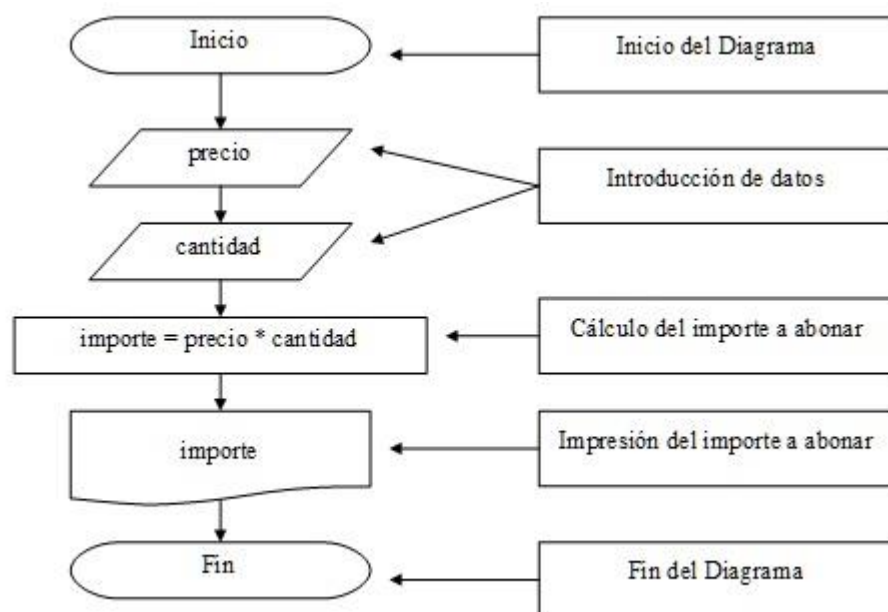
```
num1=int(input("ingrese primer valor:"))
num2=int(input("ingrese segundo valor:"))
suma=num1+num2
producto=num1*num2
print("La suma de los dos valores es")
print(suma)
print("El producto de los dos valores es")
print(producto)
```

Probar de ejecutar el programa y verificar que los resultados de salida sean correctos.

Problema 3:

Realizar la carga del precio de un producto y la cantidad a llevar. Mostrar cuanto se debe pagar (se ingresa un valor entero en el precio del producto)

Diagrama de flujo:



Tenemos

dos entradas: precio y cantidad, una operación para calcular el importe y una salida.

Programa: ejercicio4.py

```
precio=int(input("Ingrese el precio del producto:"))
cantidad=int(input("Ingrese la cantidad de productos a llevar:"))
importe=precio*cantidad
print("El importe a pagar es")
print(importe)
```


Problemas propuestos

- 1 - Realizar la carga del lado de un cuadrado por teclado, mostrar por pantalla el perímetro del mismo (El perímetro de un cuadrado se calcula multiplicando el valor del lado por cuatro)
- 2 - Escribir un programa en el cual se ingresen cuatro números, calcular e informar la suma de los dos primeros y el producto del tercero y el cuarto.
- 3 - Realizar un programa que lea cuatro valores numéricos e informar su suma y promedio.
- 4 - Calcular el sueldo mensual de un operario conociendo la cantidad de horas trabajadas y el valor por hora.

Soluciones a los problemas

Esta sección solo se debería leer luego de haber intentado por un largo tiempo la resolución en forma personal de los problemas propuestos.

Problemas 1

```
lado=int(input("Ingrese el lado del cuadrado:"))
perimetro=lado*4
print("El perimetro del cuadrado es")
print(perimetro)
```

Problemas 2

```
num1=int(input("Ingrese primer valor:"))
num2=int(input("Ingrese segundo valor:"))
num3=int(input("Ingrese tercer valor:"))
num4=int(input("Ingrese cuarto valor:"))
suma=num1+num2
producto=num3*num4
print("La suma de los dos primero valores es")
print(suma)
print("El producto del tercer y cuarto valor es")
print(producto)
```

Problemas 3

```
num1=int(input("Ingrese primer valor:"))
num2=int(input("Ingrese segundo valor:"))
num3=int(input("Ingrese tercer valor:"))
num4=int(input("Ingrese cuarto valor:"))
suma=num1+num2+num3+num4
promedio=suma/4
print("La suma de los cuatro valores es")
print(suma)
print("El promedio es")
print(promedio)
```

Problemas 4

```
horastrabajadas=int(input("Ingrese la cantidad de horas trabajadas:"))
valorhora=int(input("Importe a pagar por hora:"))
sueldo=horastrabajadas*valorhora
print("Sueldo a pagar")
print(sueldo)
```