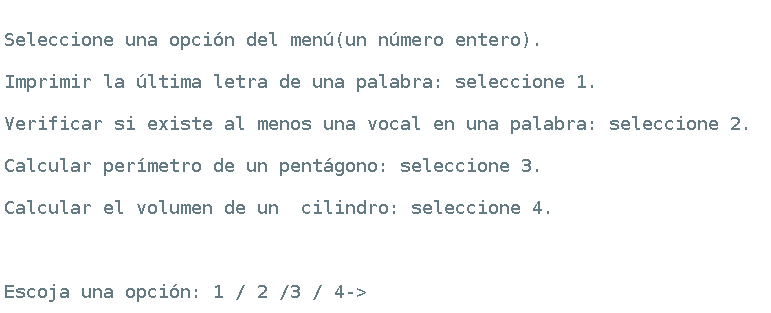
Diagrama de un programa basado en módulos y archivos separados.

Hasta ahora hemos programado de **forma secuencial**, una instrucción detrás de otra;todas las funciones y acciones están contenidas dentro del mismo archivo. Veamos un ejemplo de la forma como hemos programado hasta el momento mediante un ejemplo:

El programa es un menú que permite seleccionar cuatro opciones. Al arrancar el programa, presenta el siguiente menú:



Ahora veamos el código del programa, el cual está escrito en un solo archivo llamado ProgramaMacarron.py:

import sys

import math

print("\t\n")

print("Seleccione una opción del menú(un número entero).\t\n")

print("Imprimir la última letra de una palabra: seleccione 1.\t\n")

print("Verificar si existe al menos una vocal en una palabra: seleccione 2.\t\n")

print("Calcular perímetro de un pentágono: seleccione 3.\t\n")

print("Calcular el volumen de un cilindro: seleccione 4.\t\n")

print("\t\n")

try:

opcion=int(input("Escoja una opción: 1 / 2 /3 / 4-> "))

except:

print("Usted no ha introducido un valor numérico entero al seleccionar una opción.\t\nEl programa terminó")

sys.exit(0)

if opcion < 1 or opcion > 4:

print("Usted no ha introducido un valor entre 1(incluido) y 4(incluido) para seleccionar una opción.\t\nEl programa terminó")

sys.exit(0)

if opcion==1:

palabra= input("Introduzca una palabra: ")

if len(palabra)>0:

print("La última letra es: ",palabra[-1])

else:

print("Usted no ha introducido ninguna palabra")

if opcion==2:

palabra= input("Introduzca una palabra: ")

mensaje="No hay ninguna vocal."

if "a" in palabra:

mensaje="Existe al menos una vocal"

if "e" in palabra:

mensaje="Existe al menos una vocal"

if "i" in palabra:

mensaje="Existe al menos una vocal"

if "o" in palabra:

mensaje="Existe al menos una vocal"

if "u" in palabra:

mensaje="Existe al menos una vocal"

print( mensaje)

if opcion==3:

try:

lado=float(input("Introduzca el valor del lado del pentágono, debe ser mayor a 0: "))

except:

print("\t\nUsted no ha introducido un valor númerico en algún valor usado para la ecuación.\t\nEL programa se detendrá.")

sys.exit(0)

if lado<=0:

print("\t\nUsted ha introducido un valor númerico igual o menor a cero.\t\nEL programa se detendrá.")

sys.exit(0)

perimetro= lado \* 5

print("El perímetro del pentágono es: ",round(perimetro,2))

if opcion==4:

try:

radio=float(input("Introduzca el valor del radio del cilindro, debe ser mayor a 0: "))

altura=float(input("Introduzca el valor de la altura del cilindro, debe ser mayor a 0: "))

except:

print("\t\nUsted no ha introducido un valor númerico en algún valor usado para la ecuación.\t\nEL programa se detendrá.")

sys.exit(0)

if radio<=0 or altura<=0:

print("\t\nUsted ha introducido un valor númerico igual o menor a cero.\t\nEL programa se detendrá.")

sys.exit(0)

volumen= math.pi \* (math.pow(radio,2)) \* altura

print("El volumen del cilindro es: ",round(volumen,2))

print("\t\nFin del programa.")

Arriba, hemos coloreado en diversos colores cuatro porciones del código, cada una de ellas procesa una opción del menú.

Pues bien, pasemos a trabajar de **forma procedimental**, vamos a convertir cada porción de color en una función y las colocaremos al inicio del programa. El resto del código es el programa principal dentro del cual se llamará alguna de las funciones según la opción seleccionada del menú.

import sys

import math

def ultimaLetra(palabra):

mensaje="La última letra es: " + palabra[-1]

return (mensaje)

def verificarVocal(palabra):

mensaje="No hay ninguna vocal."

if "a" in palabra:

mensaje="Existe al menos una vocal"

if "e" in palabra:

mensaje="Existe al menos una vocal"

if "i" in palabra:

mensaje="Existe al menos una vocal"

if "o" in palabra:

mensaje="Existe al menos una vocal"

if "u" in palabra:

mensaje="Existe al menos una vocal"

return (mensaje)

def perimetro(lado):

perimetro= lado \*5

return (round(perimetro,2))

def volumen(radio, altura):

volumen= math.pi \* (math.pow(radio,2)) \* altura

return ( round(volumen,2))

**#Programa principal**

print("\t\n")

print("Seleccione una opción del menú(un número entero).\t\n")

print("Imprimir la última letra de una palabra: seleccione 1.\t\n")

print("Verificar si existe al menos una vocal en una palabra: seleccione 2.\t\n")

print("Calcular perímetro de un pentágono: seleccione 3.\t\n")

print("Calcular el volumen de un cilindro: seleccione 4.\t\n")

print("\t\n")

try:

opcion=int(input("Escoja una opción: 1 / 2 /3 / 4-> "))

except:

print("Usted no ha introducido un valor numérico entero al seleccionar una opción.\t\nEl programa terminó")

sys.exit(0)

if opcion < 1 or opcion > 4:

print("Usted no ha introducido un valor entre 1(incluido) y 4(incluido) para seleccionar una opción.\t\nEl programa terminó")

sys.exit(0)

if opcion==1:

palabra= input("Introduzca una palabra: ")

if len(palabra)>0:

ultima=**ultimaLetra(palabra)**

print(ultima)

else:

print("Usted no ha introducido ninguna palabra")

if opcion==2:

palabra= input("Introduzca una palabra: ")

resultado= **verificarVocal(palabra)**

print( resultado)

if opcion==3:

try:

lado=float(input("Introduzca el valor del lado del pentágono, debe ser mayor a 0: "))

except:

print("\t\nUsted no ha introducido un valor númerico en algún valor usado para la ecuación.\t\nEL programa se detendrá.")

sys.exit(0)

if lado<=0:

print("\t\nUsted ha introducido un valor númerico igual o menor a cero.\t\nEL programa se detendrá.")

sys.exit(0)

resultado= **perimetro(lado)**

print("El perímetro del pentágono es: ",resultado)

if opcion==4:

try:

radio=float(input("Introduzca el valor del radio del cilindro, debe ser mayor a 0: "))

altura=float(input("Introduzca el valor de la altura del cilindro, debe ser mayor a 0: "))

except:

print("\t\nUsted no ha introducido un valor númerico en algún valor usado para la ecuación.\t\nEL programa se detendrá.")

sys.exit(0)

if radio<=0 or altura<=0:

print("\t\nUsted ha introducido un valor númerico igual o menor a cero.\t\nEL programa se detendrá.")

sys.exit(0)

resultado= **volumen(radio,altura)**

print("El volumen del cilindro es: ",resultado)

print("\t\nFin del programa.")

Arriba, hemos creado cuatro funciones que contienen las operaciones de cada opción. El programa principal se encarga de presentar el menú y te pedir los valores que usarán las funciones, también se encarga de capturar los errores al introducir los valores, y si todo está bien, entonces llama a la función y le pasa los valores para que procese. La función no verifica si el valor es erróneo, asume que el programa principal depura los datos antes de enviarlos a la función. Por tanto, si el usuario ha seleccionado por ejemplo la opción 4, que calcula el volumen de un cilindro, el programa le pedirá al usuario que introduzca los valores del radio y de la altura. El programa verificará si los datos son correctos, y solo entonces invoca la función volumen(radio, altura) ; los parámetros radio y altura contienen cada uno los valores que se introdujeron mediante las instrucciones input. Cuando la función volumen (radio,altura) recibe los valores, no verifica si están correctos, eso debió ocurrir antes de llamar a dicha función. La función volumen procesa y devuelve un resultado mediante el return, el cual es recibido por el programa y asignado a la variable llamada resultado, luego imprime dicho resultado.

Por tanto las funciones son cortas y se concentran en una sola acción, en esta caso calcular el volumen, no se dedican a otras cosas como: pedir al usuario que introduzca los valores mediante input o verificar que los datos son correctos (por ejemplo que no sean letras).

Esta forma de dividir el programa en pedazos, permite que se pueda mejorar, corregir y entender mejor. La idea es dividir el programa en funciones, las cuales serán usadas por el programa principal cuando las necesite y tantas veces como sea necesario.

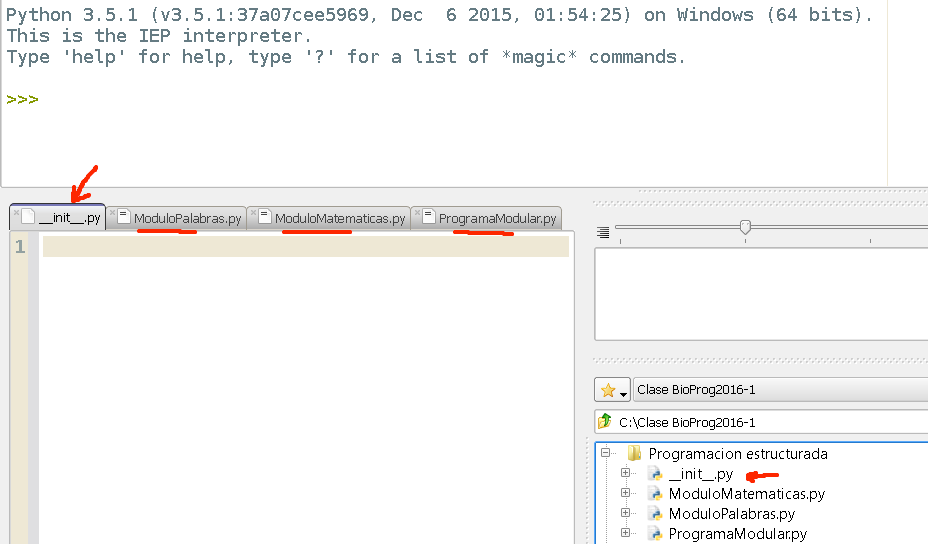
Ahora pasemos a un nivel aún más estructurado y dividido: **los módulos** (**programación modular**).

Esta forma también crea funciones separadas que se llamarán desde el programa principal, pero las funciones se escriben en un archivo diferente al que contiene el programa inicial. La ejecución arrancar desde el archivo que contiene nuestro programa principal , o sea, el que llama y usa las funciones para ejecutar ciertas acciones o cálculos. Por tanto tendremos un archivo principal y otros archivos que contienen solo funciones (módulos). Cada módulo agrupa diversas funciones por su similitud, por ejemplo: funciones de matemáticas, funciones de conversión o funciones de base de datos.

Ahora bien, la pregunta es: ¿cómo se llaman o se pueden usar las funciones que están escritas en archivos diferentes al archivo principal desde dicho archivo principal? Para ello utilizaremos la instrucción **import**, esta instrucción permite indicar al archivo principal el nombre de un módulo y posibilita entonces que se pueda leer su contenido y usar sus funciones como si estuviesen escritas dentro del archivo principal. Para que funcione la instrucción import, dicha inistrucción debe poder identificar las carpetas que contienen los módulos, y la manera de “marcar” una carpeta que contiene módulos es mediante la creación de un archivo vacío llamado **\_\_init\_\_.py** ( dos guiones bajos antes de la palabra init, y dos guiones bajos después de la palabra init) , este archivo no contiene nada, pero cuando python se percata de su existencia en una carpeta, se da cuenta que esa carpeta está capacitada para contener archivos (módulos ) cuyos contenidos pueden ser leídos desde otros archivos.

Veamos la manera de convertir nuestro programa anterior -que hemos creado en forma procedimental- en varios archivos modulares y uno principal ( un archivo principal no se identifica de ninguna manera en especial, es el que ejecutamos al lanzar el programa y dentro de su código se usan las funciones contenidas en los restantes archivos).

Hemos creado dentro de la carpeta Programacion estructurada, cuatro archivos. Uno de ellos es el archivo vacío **\_\_init\_\_.py**, el cual identifica a la carptea como contenedor de archivos módulos que se pueden importar (import) y usar en otros archivos.



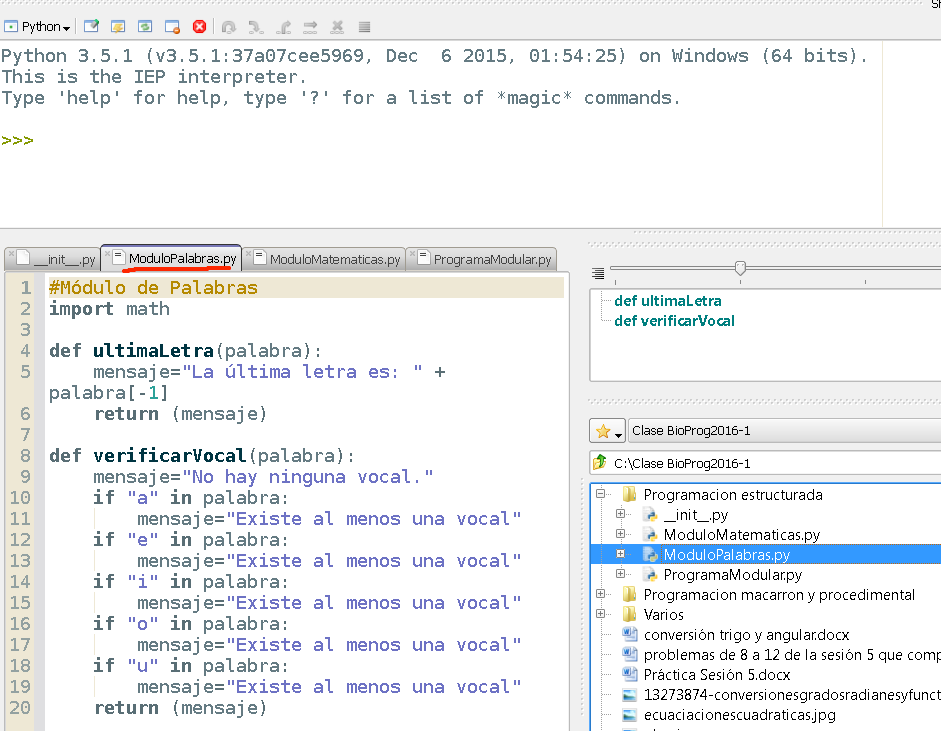
Tambien hemos creado los siguientes archivos:

Archivo **ModuloPalabras. py** , contiene las funciones ultimaLetra y VerificarVocal , ambas son parecidas, por eso se agrupan en un solo archivo.

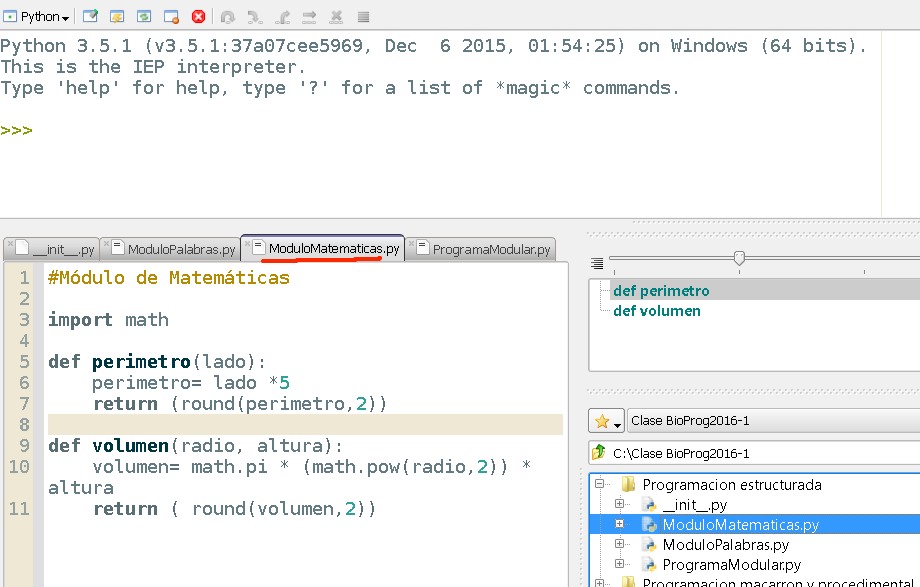
Archivo **ModuloMatematicas.py**, contiene las funciones perimetro y volumen, ambas también son parecidas.

Archivo **ProgramaModular.py**, este será el programa que ejecutaremos y contiene el programa principal que consume las funciones de los otros archivos.

Veamos el contenido del archivo **ModuloPalabras.py**:



Y ahora veremos el contenido de **ModuloMatematicas.py**



Y por último el archivo **ProgramaModular.py**

