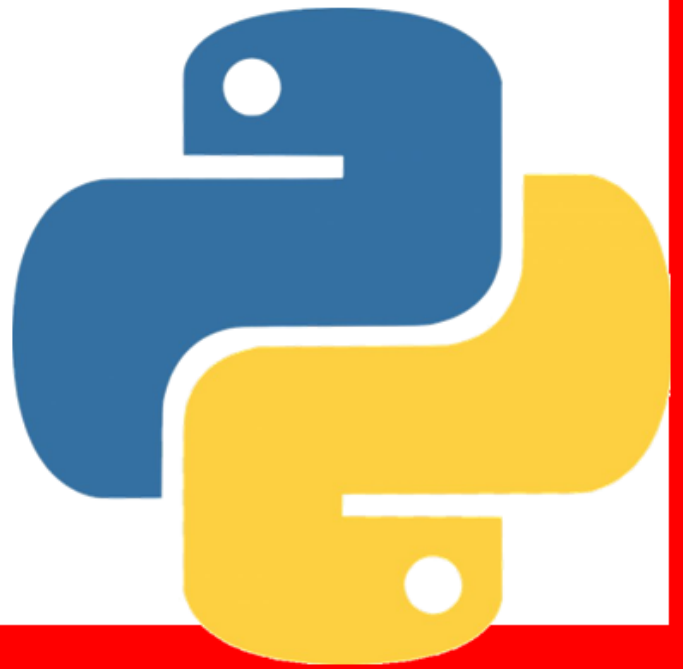
Funciones

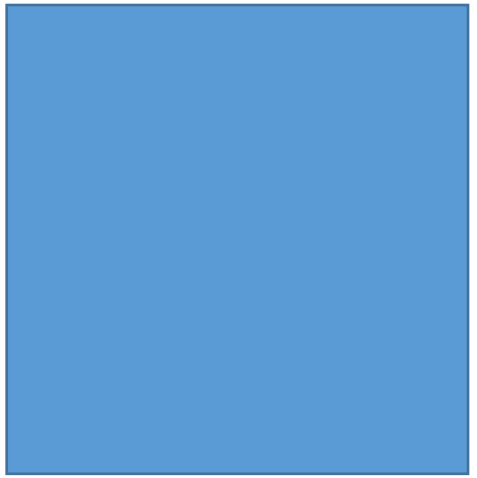
10

Curso Básico

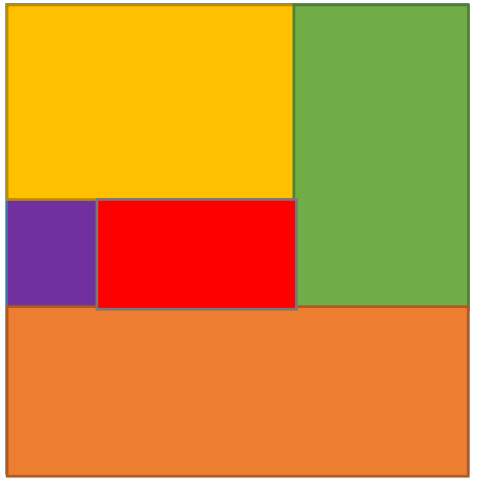
python



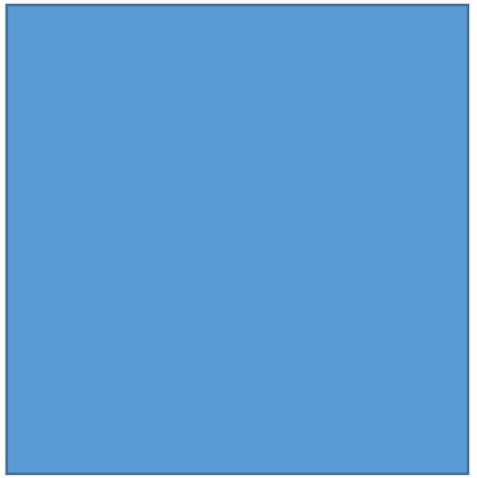
Programación modular. Funciones, procedimientos y parámetros



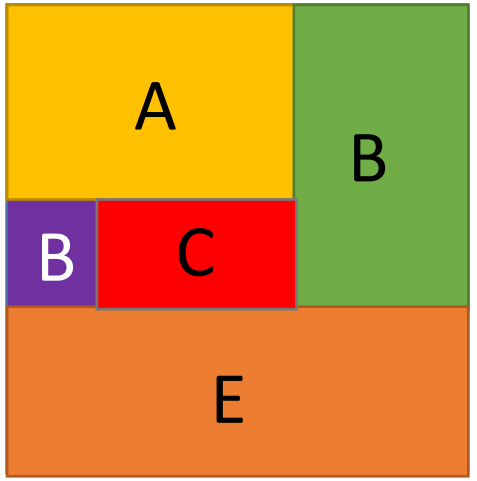
- Programación modular: Dividir un programa en módulos
- Un problema complejo se subdivide varias veces hasta resolver estructuras algorítmicas más simples
- Programas más claros, legibles y menos complejos
- Reutilizar código, subprogramas, módulos
- Facilita modificar y corregir los códigos por separado
- Crear una librería de módulos utilizables por otros programas.



Programación modular. Funciones, procedimientos y parámetros



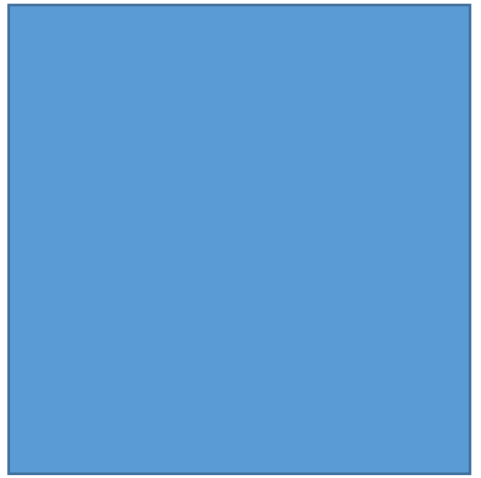
- Se parte de un programa principal que llama o utiliza otros subprogramas, que a su vez pueden utilizar otros subprogramas
- Los subprogramas (llamados también subrutinas) se refieren al conjunto de instrucciones que están separadas del programa principal y realizan cálculos o tareas:
  - Funciones
  - Procedimientos (no devuelven resultados)
- Los **valores que recibe una función** (procedimiento) son los **parámetros**.



Funciones

Turbo Código
https://www.youtube.com/user/juarezefren

Programación modular. Funciones, procedimientos y parámetros



reg

Valor que regresa

```
<class 'str'>
```

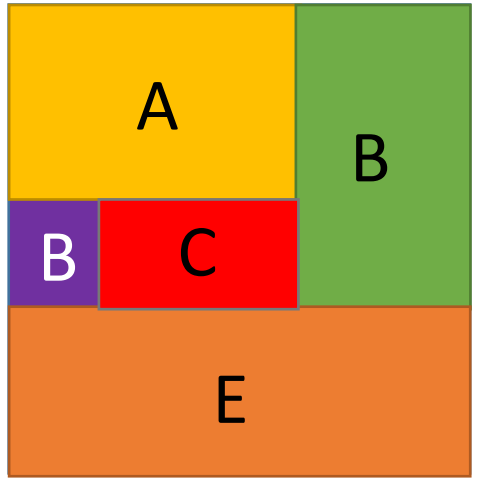
Función type()

```
reg = type( val )
```

val

Parámetro

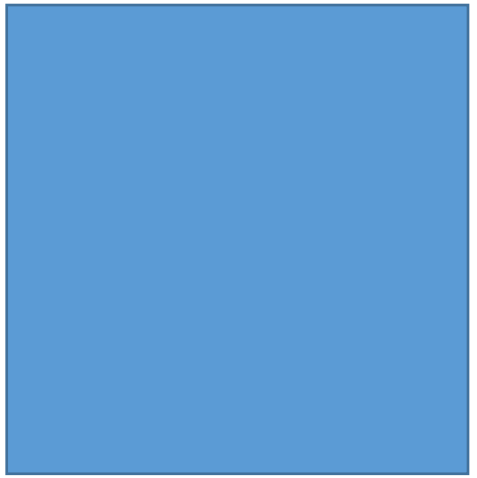
```
>>> val = 'sol'
>>> reg = type(val)
>>> reg
<class 'str'>
```



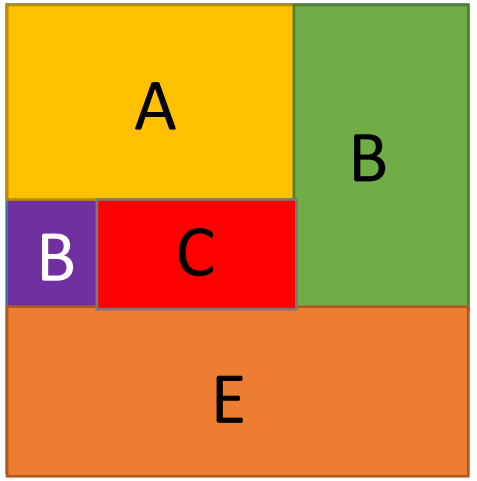
Funciones

Turbo Código
https://www.youtube.com/user/juarezefren

Programación modular. Funciones, procedimientos y parámetros



- En Python se pueden guardar un grupo de funciones en un módulo.
- Los **módulos** **se pueden** interpretar como una **biblioteca** o una caja de herramientas de funciones de una especialidad.
- Los **módulos** que se guardan en una **carpeta forman un *package*.**
- ***Los package en Python son una colección de módulos***



Turbo Código
https://www.youtube.com/user/juarezefren

Uso de funciones internas y de módulos

Función interna	Devuelve
type	tipo de dato
id	Identidad o ubicación de memoria
bin	string del binario equivalente al entero dado
int, float, str	entero, real, string del valor dado
input	string del texto leído del teclado
print	Valores a imprimir en pantalla
abs	valor absoluto de un número
round	redondea un real a los decimales especificados
Módulo math	
pi	valor de π (no es función)
sqrt	raíz cuadrada de un número
Módulo random	
randint	número entero aleatorio en el rango dado

Uso de funciones internas y de módulos

Función interna	Devuelve
type	tipo de dato
id	Identidad o ubicación de memoria
bin	string del binario equivalente al entero dado
int, float, str	entero, real, string del valor dado
input	string del texto leído del teclado
print	Valores a imprimir en pantalla
abs	valor absoluto de un número
round	redondea un real a los decimales especificados
Módulo math	
pi	valor de π (no es función)
sqrt	raíz cuadrada de un número
Módulo random	
randint	número entero aleatorio en el rango dado

Una **función es una instrucción o un bloque de** instrucciones que realizan un cálculo o una tarea

Lista de funciones internas:
https://docs.python.org/3/library/functions.html

```
from math import sqrt
x = abs(-9) + 3
y = sqrt(x)
print('Raíz cuadrada de',x, '=',y)
```


Uso de funciones internas y de módulos

Módulo math	devuelve
pi	valor π = 3.141592653589793
e	valor e = 2.718281828459045
ceil(x)	entero mayor que x, hacia ∞
floor(x)	entero menor que x, hacia -∞
trunc(x)	redondea hacia 0
factorial(x)	x!
exp(x)	e**x
log(x)	logaritmo natural (base e), ln(x)
log10(x)	logaritmo base 10
sqrt(x)	raíz cuadrada de x
sin(x), cos(x), tan(x)	seno, coseno, tangente de x
degrees(x)	ángulo x de radianes a grados
radians(x)	ángulo x de grados a radianes

Módulo random	devuelve aleatorio
randint(a,b)	entero en el rango [a, b]
randrange(a,b,paso)	de range(a, b, paso)
shuffle(s)	baraja la secuencia s
choice(s)	escogido de la secuencia s
random()	real en el rango [0.0 1.0)
seed()	inicializa generador aleatorios

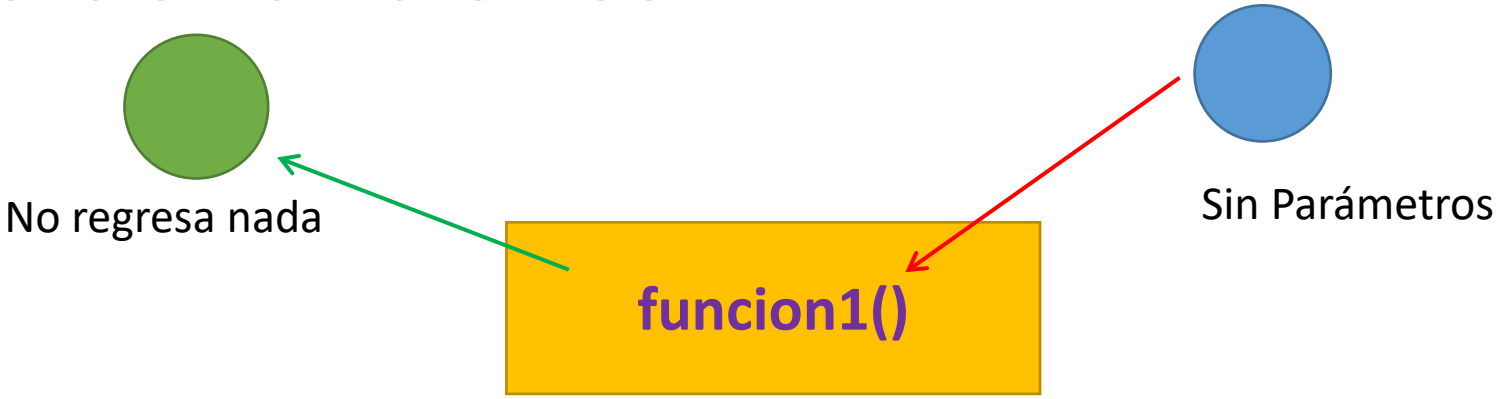
La lista completa de módulos se puede consultar en
https://docs.python.org/3/library/index.html

Diseño de funciones

- Una función, es la forma de **agrupar** expresiones y **sentencias** (algoritmos) que **realicen determinadas acciones**.
- Una función **no es ejecutada, al menos que sea llamada**.
- Se requiere que la función esté **definida antes** de que sea **usada**.
- En Python **la definición de una función es de la forma:**

```
def  nombre_funcion(parametros):
    cuerpo_de_la_funcion
```

Diseño de funciones

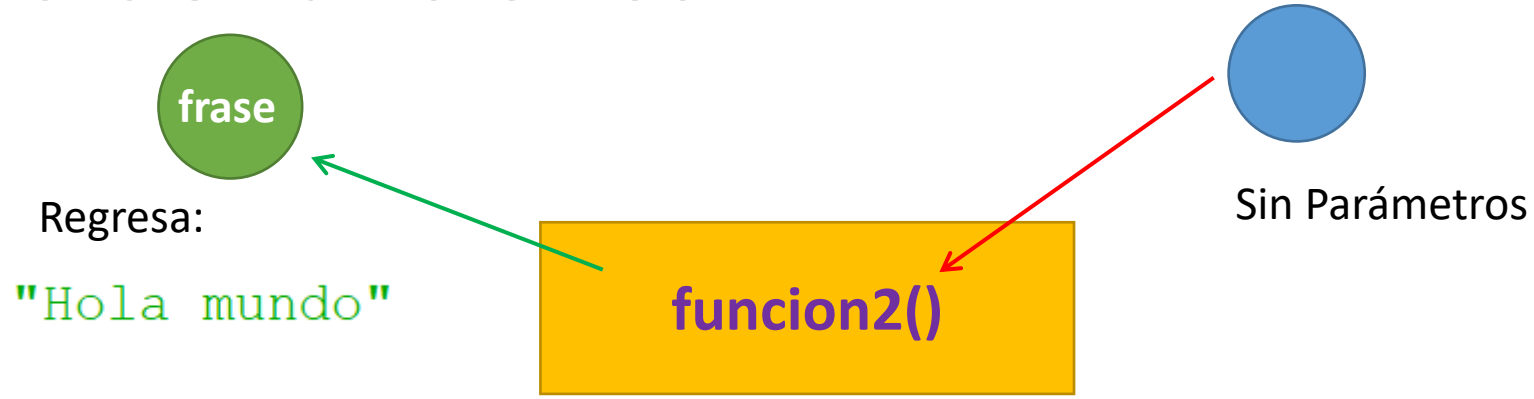



```
def funcion1():
    print("Hola mundo")
```

```
#Esto esta fuera de la función

funcion1() #Aqui llamamos a la función
```

Diseño de funciones




```
def funcion2():
    return "Hola mundo"

#Esto esta fuera de la función

frase = funcion2() #Aqui llamamos a la función
print(frase)
```

Diseño de funciones


```
def multi(num):
    if num > 0:
        res=list(range(num,num*10+1,num))
        return res
    else:
        return []
```

```
n = int(input("introduce un numero entero: "))
r = multi(n)
print(r)
```

Variables locales y variables globales

variables locales, se usan solo en la función donde fueron declaradas y al acabar la función se borran.

```
from random import random

def generaAleatorios(cant, ini, fin):
    lista=[]
    ind=0
    while ind<cant:
        val = ini + (fin - ini) * random()
        lista.append(val)
        ind+=1
    return lista

n = int(input("Cuantos numeros deseas: "))
ini = int(input("Inicio de rango: "))
fin = int(input("Final de rango: "))
print( generaAleatorios(n,ini,fin) )

print(lista)
```

X

variables globales. Se declaran precedidas por la palabra reservada global

```
def f(x):
    global a
    a = 3
    return 3*x

a = 7
print('Función:',f(4))
print('Valor de la variable global a:',a)
```

Turbo Código

https://www.youtube.com/user/juarezefren

Módulos: integrar funciones en una biblioteca

- Las funciones se pueden **guardar en un módulo** para ser reutilizadas cuando queramos.
- Un **módulo será un archivo de extensión .py que contiene la definición de un grupo de** funciones y otros valores.
- Los módulos **los podemos importar** para hacer uso de sus funciones y constantes.

```
from Geomet import pi, AreaCirc, AreaRect
print(AreaCirc(0.5))
print(AreaRect(3, 4))
```

Principal.py

```
pi = 3.14159

def AreaCirc(radio):
    return pi*radio**2

def PerimCirc(radio):
    return 2*pi*radio

def VolCilindro(radio, h):
    return h*pi*radio**2

def AreaRect(b, h):
    return b*h

def PerimRect(b, h):
    return 2*(b+h)

def VolOrtoed(b, h, a):
    return b*h*a
```

Geomet.py

__main__

```python
def suma(a, b):
    return a+b

res = suma(3, 5)
print(res)
```

```python
def suma(a, b):
    return a+b

if __name__ == "__main__":
    res = suma(3, 5)
    print(res)
```

```python
from random import random

def generaAleatorios(cantNumeros, ini, fin):
    lista=[]
    ind=0
    while ind<cantNumeros:
        val = ini + (fin - ini) * random()
        lista.append(val)
        ind+=1
    return lista

def main():
    n = int(input("Cuantos numeros deseas: "))
    ini = int(input("Inicio de rango: "))
    fin = int(input("Final de rango: "))
    print( generaAleatorios(n,ini,fin) )

if __name__ == "__main__":
    main()
```

Funciones

10

Curso Básico

python