

Ciclo 1 Fundamentos de programación con Python Sesión 16: Uso de Funciones en Python

Programa Ciencias de la Computación e Inteligencia Artificial Escuela de Ciencias Exactas e Ingeniería Universidad Sergio Arboleda Bogotá







Agenda

- 1. Elementos de un programa en Python (Repaso)
- 2. Funciones integradas
- 3. Funciones con un parámetro de entrada
- 4. Funciones para cadenas en Python
- 5. Funciones varias en Python
- 6. Ejemplo
- 7. Ejercicios







Un programa en Python como ya hemos visto es un archivo de texto, el cual contiene las expresiones y las sentencias o instrucciones propias para el leguaje Python.

Las expresiones y las instrucciones o sentencias las vamos a crear utilizando y combinando los elementos básicos del lenguaje.







Los elementos que forman este lenguaje son de diferentes tipos:

- Palabras reservadas (keywords)
- Funciones integradas (built-in functions)
- Literales
- Operadores
- Delimitadores
- Identificadores







Recordemos que para que un programa se pueda ejecutar, el programa debe ser sintácticamente correcto, es decir, utilizar los elementos del lenguaje Python respetando su reglas de "ensamblaje". Dentro de esta reglas esenciales tenemos:

- Líneas
- Espacios







Palabras reservadas (keywords)

> Las palabras reservadas de Python son las que forman el núcleo del lenguaje Python. Son las siguientes:

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield







Literales o datos simples

Los datos simples que Python es capaz de manejar:

- números: valores lógicos, enteros, decimales y complejos, en notación decimal, octal o hexadecimal
- Cadenas de texto







Operadores

Los operadores son los caracteres que definen operaciones matemáticas (lógicas y aritméticas). Son los siguientes:

```
+ - * ** / // % @

<< >> & | ^ ~

< > >= !=
```







Delimitadores

Los delimitadores son los caracteres que permiten delimitar, separar o representar expresiones. Son los siguientes:

```
' " # \
( ) [ ] { }
, : . ; @ = ->
+= -= *= /= //= %= @=
&= |= ^= >>= <<= **=
```







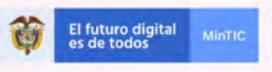
Identificadores

Los identificadores son las palabras que se utilizan para nombrar elementos creados por el usuario u otros usuarios. Esos elementos pueden ser variables u objetos funciones, clases que combinan ambos o módulos que agrupan los elementos anteriores, etc.

Están formados por letras (mayúsculas y minúsculas), números y el carácter guion bajo (_).







2. Funciones integradas

Funciones integradas (built-in functions)

Una función es un bloque de instrucciones agrupadas, que permiten reutilizar partes de un programa y Python incluye varias funciones de forma predeterminada.

Una función (en este contexto) es una parte separada del código de computadora el cual es capaz de:

- Efecto
- Resultado
- Argumento







2. Funciones integradas

```
abs()
              dict()
                           help()
                                        min()
                                                    setattr()
all()
              dir()
                                                    slice()
                           hex()
                                        next()
any()
              divmod()
                           id()
                                        object()
                                                    sorted()
              enumerate()
                                                    staticmethod()
ascii()
                           input()
                                        oct()
              eval()
bin()
                           int()
                                        open()
                                                    str()
bool()
              exec()
                           isinstance() ord()
                                                    sum()
              filter()
                           issubclass() pow()
                                                    super()
bytearray()
              float()
bytes()
                           iter()
                                        print()
                                                    tuple()
                           len()
callable()
              format()
                                        property() type()
              frozenset() list()
chr()
                                        range()
                                                    vars()
classmethod() getattr()
                           locals()
                                        repr()
                                                    zip()
                                        reversed() __import__()
compile()
              globals()
                           map()
complex()
              hasattr()
                                         round()
                           max()
delattr()
              hash()
                           memoryview() set()
```





.



2. Funciones integradas

Dentro del listado anterior de funciones las más conocidas son:

- Función print(): imprime en pantalla el argumento.
- Función input(): Permite la entrada de datos al usuario

```
print("como te llamas compañero")
nombre=input()
print("hola:", nombre, "bienvenid@ a este curso de Python...")

Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AM D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Liliana/Desktop/ejemplo función print e input.py ======
como te llamas compañero
Liliana Contreras
hola: Liliana Contreras bienvenid@ a este curso de Python...
>>> |
```







3. Función con un parámetro de entrada

¿Qué sucede cuando Python encuentra una invocación como la que está a continuación?



nombreFunción(argumento)





3. Función con un parámetro de entrada

En programación, así como en matemáticas, para las funciones definidas como:

 $f: A \rightarrow B$

- Al conjunto A se le denomina dominio y al conjunto B como rango.
- A partir de estos objetos se construye el encabezado de las funciones de programación.
- Sobre esta función se tiene que f corresponde al nombre de la función, el conjunto A corresponde al tipo de los argumentos de dicha función y el conjunto B que es el rango corresponderá al valor de retorno de dicha función.







4. Funciones para cadenas en Python

Función	Utilidad	Ejemplo	Resultado
len()	Determina la longitud en caracteres de una cadena.	len("Hola Python")	11
join()	Convierte en cadena utilizando una separación	Lista = ['Python', 'es'] '-'.join(Lista)	'Python-es'
split()	Convierte una cadena con un separador en una lista	a = ("hola esto sera una lista") Lista2 = a.split() print (Lista2)	['hola', 'esto', 'sera', 'una', 'lista']
replace()	Reemplaza una cadena por otra	texto = "Manuel es mi amigo" print (texto.replace ('es', 'era'))	Manuel era mi amigo
upper()	Convierte una cadena en Mayúsculas	texto = "Manuel es mi amigo" texto.upper()	'MANUEL ES MI AMIGO'
lower()	Convierte una cadena en Minúsculas	texto = "MaNueL eS ml AmlgO" texto.lower()	'manuel es mi amigo'





5. Funciones varias en Python

Función	Utilidad	Ejemplo	Resultado
range()	Crea un rango de números	x = range (5) print (list(x))	[0, 1, 2, 3, 4]
str()	Convierte un valor numérico a texto	str(22)	′22′
int()	Convierte a valor entero	int('22')	22
float()	Convierte un valor a decimal	float('2.22')	2.22
max()	Determina el máximo entre un grupo de números	x = [0, 1, 2] print (max(x))	2
min()	Determina el mínimo entre un grupo de números	x = [0, 1, 2] print (min(x))	0
sum()	Suma el total de una lista de números	x = [0, 1, 2] print (sum(x))	3







5. Funciones varias en Python

ı	Función	Utilidad	Ejemplo	Resultado
	list()	Crea una lista a partir de un elemento	x = range (5) print (list(x))	[0, 1, 2, 3, 4]
	tuple()	Crea o convierte en una tupla	print(tuple(x))	(0, 1, 2, 3, 4)
	open()	Abre, crea, edita un elemento (archivo)	with open("Ejercicios/Ejercicio.py", "w") as variables: variables.writelines("Eje")	Crea el archivo "Ejercicio.py" con el contenido "Eje"
	ord()	Devuelve el valor ASCII de una cadena o carácter.	print(ord('A'))	65
	round()	Redondea después de la coma de un decimal	print (round(12.723))	13
	type()	Devuelve el tipo de un elemento	type(x)	<class 'range'=""></class>







 Realizar un programa que capture dos números enteros ingresados por el usuario y realice las operaciones aritméticas básicas (suma, resta, multiplicación y división)









```
peracionesBasicas.py - C:/Users/Liliana/Desktop/operacionesBasicas.py (3.9.2)
<u>File Edit Format Run Options Window Help</u>
print("digite el primer numero")
numero1=float(input())
print("digite el segundo numero")
numero2=float(input())
resultado=numero1+numero2
print("La suma de", numero1,"+", numero2, "=", resultado)
resultado=numero1-numero2
print("La resta de", numero1, "-", numero2, "=", resultado)
resultado=numero1*numero2
print("La producto de", numero1, "*", numero2, "=", resultado)
resultado=numero1/numero2
print("La división de", numero1,"/", numero2, "=", resultado)
```



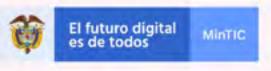




```
IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AM
D64) 1 on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======= RESTART: C:/Users/Liliana/Desktop/operacionesBasicas.py ========
digite el primer numero
4.5
digite el segundo numero
3.2
La suma de 4.5 + 3.2 = 7.7
La producto de 4.5 * 3.2 = 14.4
La división de 4.5 / 3.2 = 1.40625
>>>
```







Realizar un programa que calcule la siguiente expresión:

```
((5 * ((25 \mod 13) + 100) / (2 * 13)) // 2)
```

```
print((5 * ((25 % 13) + 100) / (2 * 13)) // 2)
========= RESTART: C:/Users/Liliana/Desktop/operacionesBasicas.
10.0
>>> |
```







Crea un programa que pida al usuario un número real y muestra su raíz cuadrada.

```
import math
print("Digite el número al que desea calcularte la raiz cuadrada")
numero=float(input())
raizC=math.sqrt(numero)
print("La raiz cuadrada de ",numero, "es",raizC)
========== RESTART: C:\Users\Liliana\Desktop\operacionesBasicas.py ============
Digite el número al que desea calcularte la raiz cuadrada
7.66
La raiz cuadrada de 7.66 es 2.7676705006196096
>>>> |
```





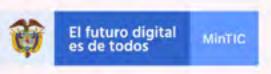




 Crear un programa que permita solucionar un problema a través del teorema de Pitágoras:









7. Ejercicios propuestos

- Crea un programa que pida al usuario un número real y muestra su raíz cuarta (la raíz cuadrada de la raíz cuadrada).
- Crear un programa que calcule y escriba el área para un circulo.
- Elaborar un programa que calcule y escriba el precio de venta para un artículo, se tiene, el nombre del producto y el costo de producción; el precio de venta se calcula añadiendo el 120 % como utilidad y 15 % de impuestos.

Preguntas







