

# Python 5: Shell, Funciones y Ejecución Condicional en Python

Juan F. Pérez

Departamento MACC  
Matemáticas Aplicadas y Ciencias de la Computación  
Universidad del Rosario

*[juanferna.perez@urosario.edu.co](mailto:juanferna.perez@urosario.edu.co)*

2018

# Contenidos

- 1 Python Shell
- 2 Ejecución Condicional en Python
- 3 Bloques if/else
- 4 Bloques if/elif/else
- 5 Funciones: argumentos y resultados
- 6 Funciones y Condicionales

# Python Shell

# Python Shell

- Interfaz interactiva
- Intérprete de líneas de comandos
- Usuario interactúa a través de líneas de comandos
- Ejecuta un comando ingresado...

# Python Shell

- Interfaz interactiva
- Intérprete de líneas de comandos
- Usuario interactúa a través de líneas de comandos
- Ejecuta un comando ingresado...
- ... y muestra inmediatamente el resultado

# Python Shell

En Windows IDLE:

- Iniciamos IDLE

# Python Shell

En Windows IDLE:

- Iniciamos IDLE
- Directamente en IDLE escribimos los comandos

# Python Shell

En Windows IDLE:

- Iniciamos IDLE
- Directamente en IDLE escribimos los comandos

## Ejemplo

```
print(" Hola  Mundo!" )
```



# Python Shell

## Otro Ejemplo

```
a = 5.3
```

```
b = 3.2
```

```
a+b
```

```
a*b
```

```
a/b
```

# Python Shell

En Windows línea de comandos:

- Inicio+CMD

# Python Shell

En Windows línea de comandos:

- Inicio+CMD
- `python`

# Python Shell

En Windows línea de comandos:

- Inicio+CMD
- `python`
- comandos

# Python Shell

En Windows línea de comandos:

- Inicio+CMD
- `python`
- comandos
- `quit()` para terminar

# Python Shell

En Ubuntu línea de comandos:

- CTRL+T

# Python Shell

En Ubuntu línea de comandos:

- CTRL+T
- `python`

# Python Shell

En Ubuntu línea de comandos:

- CTRL+T
- `python`
- comandos



# Python Shell

En Ubuntu línea de comandos:

- CTRL+T
- `python`
- comandos
- `quit()` para terminar

# Ejecución Condicional en Python

# Ejecución condicional

## Ejecución condicional:

Algunas instrucciones se ejecutan solamente si se cumple una condición

# Ejecución condicional

## Ejecución condicional:

Algunas instrucciones se ejecutan solamente si se cumple una condición

Si no se cumple la condición, otras instrucciones pueden ejecutarse

# Ejecución condicional

## Ejecución condicional:

Algunas instrucciones se ejecutan solamente si se cumple una condición

Si no se cumple la condición, otras instrucciones pueden ejecutarse

Dos tipos de instrucciones para ejecución condicional:

# Ejecución condicional

## Ejecución condicional:

Algunas instrucciones se ejecutan solamente si se cumple una condición

Si no se cumple la condición, otras instrucciones pueden ejecutarse

Dos tipos de instrucciones para ejecución condicional:

IF/THEN

# Ejecución condicional

## Ejecución condicional:

Algunas instrucciones se ejecutan solamente si se cumple una condición

Si no se cumple la condición, otras instrucciones pueden ejecutarse

Dos tipos de instrucciones para ejecución condicional:

IF/THEN

IF/THEN/ELSE

# Lógica en Python

## Operaciones lógicas:

- `and`: y
- `or`: o
- `not`: negación



# Lógica en Python

## Operaciones lógicas:

- and: y
- or: o
- not: negación

## Valores lógicos :

- True: verdadero
- False: falso

# Operaciones lógicas en Python (Shell)

```
True and True
```

```
True and False
```

```
not (True and False)
```

```
1==1
```

```
1==1 and 1==2
```

```
not (1 != 10 or 3 == 4 )
```

```
"test" == "test"
```

```
"test" == "testing"
```

```
"hola" == "adios" and not (3 == 4 and 3==3)
```

# Condicionales en Python: `if`

```
# -*- coding: utf-8 -*-  
mi_edad = 80  
edad_max_adoles = 18  
edad_max_adulto = 75  
  
if mi_edad < edad_max_adoles:  
    print(u"Es un niño/adolescente")  
if mi_edad > edad_max_adoles and \\\nmi_edad < edad_max_adulto:  
    print(u"Es un adulto")  
if mi_edad > edad_max_adulto:  
    print(u"Es un adulto mayor")
```

# Condicionales en Python: if - Test

Probemos el código anterior con

- `mi_edad=5`
- `mi_edad=25`
- `mi_edad=75`
- `mi_edad=80`

# Condicionales en Python: if - Test

Probemos el código anterior con

- `mi_edad=5`
- `mi_edad=25`
- `mi_edad=75`
- `mi_edad=80`
- ¿Problemas?

# Nueva versión

```
# -*- coding: utf-8 -*-  
mi_edad = 80  
edad_max_adoles = 18  
edad_max_adulto = 75  
  
if mi_edad <= edad_max_adoles:  
    print(u"Es un niño/adolescente")  
if mi_edad > edad_max_adoles and \\\nmi_edad <= edad_max_adulto:  
    print(u"Es un adulto")  
if mi_edad > edad_max_adulto:  
    print(u"Es un adulto mayor")
```

## Bloques if/else

# Bloques if/else

```
# -*- coding: utf-8 -*-  
mi_edad = 5  
edad_max_adoles = 18  
edad_max_adulto = 75  
if mi_edad <= edad_max_adoles:  
    print(u"Es un niño/adolescente")  
else:  
    if mi_edad <= edad_max_adulto:  
        print(u"Es un adulto")  
    else:  
        print(u"Es un adulto mayor")
```

Ojo con la indentación



## Bloques if/elif/else

# Bloques if/elif/else

```
# -*- coding: utf-8 -*-  
mi_edad = 5  
edad_max_adoles = 18  
edad_max_adulto = 75  
if mi_edad <= edad_max_adoles:  
    print(u"Es un niño/adolescente")  
elif mi_edad <= edad_max_adulto:  
    print(u"Es un adulto")  
else:  
    print(u"Es un adulto mayor")
```

# Funciones en Python

- Todas las funciones se definen al principio (como para Karel)

# Funciones en Python

- Todas las funciones se definen al principio (como para Karel)
- La definición de una función sigue el formato

```
def nombre():  
    instruccion 1  
    instruccion 2  
    instruccion 3  
    ...
```

# Funciones en Python

- Todas las funciones se definen al principio (como para Karel)
- La definición de una función sigue el formato

```
def nombre():  
    instruccion 1  
    instruccion 2  
    instruccion 3  
    ...
```

- Note la indentación (4 espacios estándar - *evite tabs*)

# Funciones en Python

- Todas las funciones se definen al principio (como para Karel)
- La definición de una función sigue el formato

```
def nombre():  
    instruccion 1  
    instruccion 2  
    instruccion 3  
    ...
```

- Note la indentación (4 espacios estándar - *evite tabs*)
- Después de definidas, llamamos las funciones como  
nombre()

## Funciones en Python - Otro Ejemplo

```
# -*- coding: utf-8 -*-  
def imprimeDia():  
    print(15)  
def imprimeMes():  
    print(" Enero")  
  
print(u"¿Qué día cumples años?")  
imprimeDia()  
print(u"¿En qué mes?")  
imprimeMes()
```

# Algunos comentarios sobre las funciones

- Cada nueva función es como una entrada en un diccionario para Python.
- Todas las funciones nuevas las definimos al principio, antes del bloque de ejecución principal.



# Funciones: argumentos y resultados

# Funciones: pasando argumentos

- Las funciones no pueden cambiar su definición
- Pero pueden recibir argumentos que cambian su resultado (o incluso las instrucciones que ejecuta)

```
# -*- coding: utf-8 -*-
def imprimaNumero(arg1):
    print(u"El número es %d" % arg1)
    print("— — — — —")
```

```
imprimaNumero(1)
imprimaNumero(10)
imprimaNumero(100)
imprimaNumero(1000)
```

# Funciones con argumentos: otro ejemplo

```
# -*- coding: utf-8 -*-
def imprimaSuma(arg1, arg2):
    print(u"La suma %f + %f es igual a %f" \
          %(arg1, arg2, arg1+arg2))
    print("— — — — —")
def imprimaProducto(arg1, arg2):
    print(u"El producto %f * %f es %f" \
          %(arg1, arg2, arg1*arg2))
    print("— — — — —")

imprimaSuma(2, 5)
imprimaProducto(2, 5)
imprimaSuma(10.5, 5.2)
imprimaProducto(10.5, 5.2)
```

# Funciones con argumentos: y otro ejemplo

```
# -*- coding: utf-8 -*-  
def imprimaDiferencia(nota1, nota2):  
    print(nota1)  
    print(nota2)  
    print(nota1-nota2)  
  
miNota = 95;  
tuNota = 98;  
imprimaDiferencia(miNota, tuNota)  
imprimaDiferencia(miNota, 90)  
imprimaDiferencia(tuNota+1, miNota-10)  
print("mi nota: %d" % miNota)  
print("tu nota: %d" % tuNota)
```

# Funciones: retornando resultados

- Las funciones pueden retornar un resultado
- Valor del resultados se puede usar en el programa principal (o la función que llama a la otra función)

## Funciones: retornando resultados

- Las funciones pueden retornar un resultado
- Valor del resultados se puede usar en el programa principal (o la función que llama a la otra función)

```
def suma(a,b):  
    c = a + b  
    return c
```

```
num1 = 5  
num2 = 63  
num3 = 18  
res1 = suma(num1,num2)  
print(" % f + % f = % f" % (num1, num2, res1))  
res2 = suma(res1,num3)  
print(" % f + % f = % f" % (res1, num3, res2))
```

# Funciones: retornando varios resultados

```
# -*- coding: utf-8 -*-  
def sumaDif(a,b):  
    suma = a + b  
    dif = a - b  
    return suma, dif  
  
num1 = 5  
num2 = 63  
suma12, dif12 = sumaDif(num1,num2)  
print(u"Números: % f, % f" % (num1, num2))  
print(u"Suma: % f" % suma12)  
print(u"Diferencia: % f" % dif12)
```

# Funciones y Condicionales



# Funciones y Condicionales (1 de 5)

```
def bienvenida():  
    print(u" Bienvenido al programa concurso")  
    print(u" El Precio es Correcto\n")
```

## Funciones y Condicionales (2 de 5)

```
def elegirPremio():  
    print(u"Al frente tiene 2 posibles premios")  
    print(u"Seleccione el número:")  
    print(u"\t1 para escoger el premio grande")  
    print(u"\t2 para escoger el premio pequeño")  
    premio = int(input("?"))  
    if premio == 1:  
        gana = premio1()  
    elif premio == 2:  
        gana = premio2()  
    else:  
        print(u"Premio no válido. Adios!")  
        gana = False  
    return gana
```

## Funciones y Condicionales (3 de 5)

```
def premio1():  
    print(u"Este premio es un carro 0km de gama media"  
    print(u"¿Cuál es el precio del carro en millones?"  
    precioCarro = float(input("?") )  
    if precioCarro >= 65 and precioCarro <= 66:  
        print(u" Felicitaciones!")  
        print(u"Se ha ganado el carro!")  
        return True  
    else:  
        print(u"Perdió! El precio era 65 millones")  
        return False
```

## Funciones y Condicionales (4 de 5)

```
def premio2():  
    print(u"Este premio es un reloj de alta gama")  
    print(u"¿Cuál es el precio del reloj en millones?")  
    precioReloj = float(input("?"))  
    if precioReloj >= 15 and precioReloj <= 16:  
        print(u"Felicitaciones!")  
        print(u"Se ha ganado el reloj!")  
        return True  
    else:  
        print(u"Perdió! El precio era 15 millones")  
        return False
```

## Funciones y Condicionales (5 de 5)

```
# Programa principal
bienvenida()
gana = elegirPremio()
if gana == True:
    print(u"Puede reclamar su premio en \
las instalaciones de El Precio es Correcto.")
else:
    print( u"Lo sentimos, lo esperamos en \
una próxima oportunidad.")
```