

# REPASO

Programación con Python 2.7+

# Conocimientos básicos

- Tipos de datos
  - Números, cadenas booleanos
    - 5, 3.714, “hola mundo”, True (1), False (0)
  - Función ***type***
    - `type(“hola”), type(5), type(5.0)`
- Colecciones
  - Listas
    - `l = [1,2,3,4]`
  - Tuplas
    - `t = (1,2,3,4)`
  - Diccionarios
    - `D = {“color”: “Rojo”, “Altura”: 170}`

# Control de Flujo

- Sentencias condicionales

- If

```
fav = "mundogeek.net"
# si (if) fav es igual a "mundogeek.net"
if fav == "mundogeek.net":
    print "Tienes buen gusto!"
    print "Gracias"
```

- If ... else

```
if fav == "mundogeek.net":
    print "Tienes buen gusto!"
    print "Gracias"
```

```
if fav != "mundogeek.net":
    print "Vaya, que lástima"
```

- If ... elif ... elif ... else

```
if numero < 0:
    print "Negativo"
elif numero > 0:
    print "Positivo"
else:
    print "Cero"
```

- A if C else B

```
var = "par" if (num % 2 == 0) else "impar"
```

# Control de flujo

- Bucles

- while

```
edad = 0
while edad < 18:
    edad = edad + 1
    print "Felicidades, tienes " + str(edad)
```

```
while True:
    entrada = raw_input("> ")
    if entrada == "adios":
        break
    else:
        print entrada
```

- for .. in

```
int mi_array[] = {1, 2, 3, 4, 5};
int i;
for(i = 0; i < 5; i++) {
    printf("%d\n", mi_array[i]);
}
```

```
secuencia = ["uno", "dos", "tres"]
for elemento in secuencia:
    print elemento
```

# Funciones

```
def mi_funcion(param1, param2):  
    """Esta funcion imprime los dos valores pasados  
    como parametros"""  
    print param1  
    print param2
```

```
mi_funcion("hola", 2)
```

```
mi_funcion(param2 = 2, param1 = "hola")
```

```
def imprimir(texto, veces = 1):  
    print veces * texto
```

```
>>> imprimir("hola")  
hola
```

```
>>> imprimir("hola", 2)  
holahola
```

```
def varios(param1, param2, *otros):  
    for val in otros:  
        print val
```

```
varios(1, 2)  
varios(1, 2, 3)  
varios(1, 2, 3, 4)
```

# Ejercicio 1

- Escribir un programa que pida dos números enteros e imprima True si la división es exacta, Falso si no.

```
1  # -*- coding: utf-8 -*-
2
3  |
4  def divisible(x, y):
5      return True if x % y == 0 else False
6
7
8  if __name__ == "__main__":
9      x = input("Ingrese un numero entero ")
10     y = input("Ingrese otro numero entero ")
11
12     print(divisible(x, y))
13
```

# Ejercicio 2

- Escriba una función que pida dos números e imprima cual es el mayor, cual es el menor o si son iguales

```
1  # -*- coding: utf-8 -*-
2
3
4  def compara(x, y):
5
6      if x == y:
7          print("son iguales")
8      elif x < y:
9          print("{} es mayor que {}".format(y, x))
10     else:
11         print("{} es mayor que {}".format(x, y))
12
13
14  if __name__ == "__main__":
15     x = input("ingrese primer número: ")
16     y = input("ingrese segundo número: ")
17     compara(x, y)
18
```

# Ejercicio 3

- Escriba una función que reciba una variable entera *size* y retorne en una lista todos los números que están entre 1 y *size* que sean múltiplos de 3 y 7.

```
1  # -*- coding: utf-8 -*-
2
3
4  def numeros_multiplos(size):
5      multiplos = list()
6      for i in range(size):
7          if (i % 3 == 0 and i % 7 == 0):
8              multiplos.append(i)
9      return multiplos
10
11
12 if __name__ == "__main__":
13
14     size = input("Ingrese un número: ")
15     print(numeros_multiplos(size))
16
```



# Ejercicio 4

- Escriba una función que reciba una lista de números y retorne la lista ordenada en orden ascendente. No utilice las funciones sort o sorted de python.

```
1  # -*- coding: utf-8 -*-
2
3
4  def burbuja(lista):
5      size = len(lista)
6      i, j = 0, 0
7      while i < size:
8          while j < size:
9              if lista[i] > lista[j]:
10                 temp = lista[i]
11                 lista[j] = lista[i]
12                 lista[i] = temp
13                 j += 1
14             i += 1
15         return lista
16
17
18 if __name__ == "__main__":
19
20     lista = [
21         45, 12, 4, 67, 890, 2, 100, 23,
22         45, 35, 981, 78, 16, 10030]
23     print(burbuja(lista))
24
```