Módulo 3: Estructuras de Control

Programación I

Repaso: Estructura de Datos: Listas

Es un tipo de colección ordenada. Sería equivalente a lo que en otros lenguajes se conoce por arrays, o vectores.

Pueden contener cualquier tipo de dato: números, cadenas,booleanos, ... y también listas.

Crear una lista es tan sencillo como indicar entre corchetes, y separados por comas, los valores que queremos incluir en la lista:

```
l = [22, True, "una lista", [1, 2]]
```

Repaso: Estructura de Datos: Listas, Slicing

Si en lugar de un número escribimos dos números **inicio** y **fin** separados por dos puntos **(inicio:fin)** Python interpretará que queremos una lista que vaya desde la posición **inicio** a la posición **fin**, **sin incluir este último**. Si escribimos tres números **(inicio:fin:salto)** en lugar de dos, el tercero se utiliza para determinar cada cuantas posiciones añadir un elemento a la lista.

```
l = [99, True, "una lista", [1, 2]]
mi_var = l[0:2]   # mi_var vale [99, True]
mi_var = l[0:4:2] # mi_var vale [99, "una lista"]
```

Repaso: Estructura de Datos: Diccionarios

Los diccionarios, también llamados matrices asociativas, deben su nombre a que son colecciones que relacionan una clave y un valor. Ejemplo:

```
>>> diccionario = {"nombre" : "Carlos", "edad" : 22, "dni": 34167583 }
>>> print(diccionario["nombre"])
Carlos
>>> print(diccionario["edad"])
22
>>> print(diccionario["edad"] * diccionario["dni"])
751686826
>>> diccionario["apellido"] = "Carlin"
>>> print(diccionario)
{'edad': 22, 'nombre': 'Carlos', 'dni': 34167583, 'apellido': 'Carlin'}
```

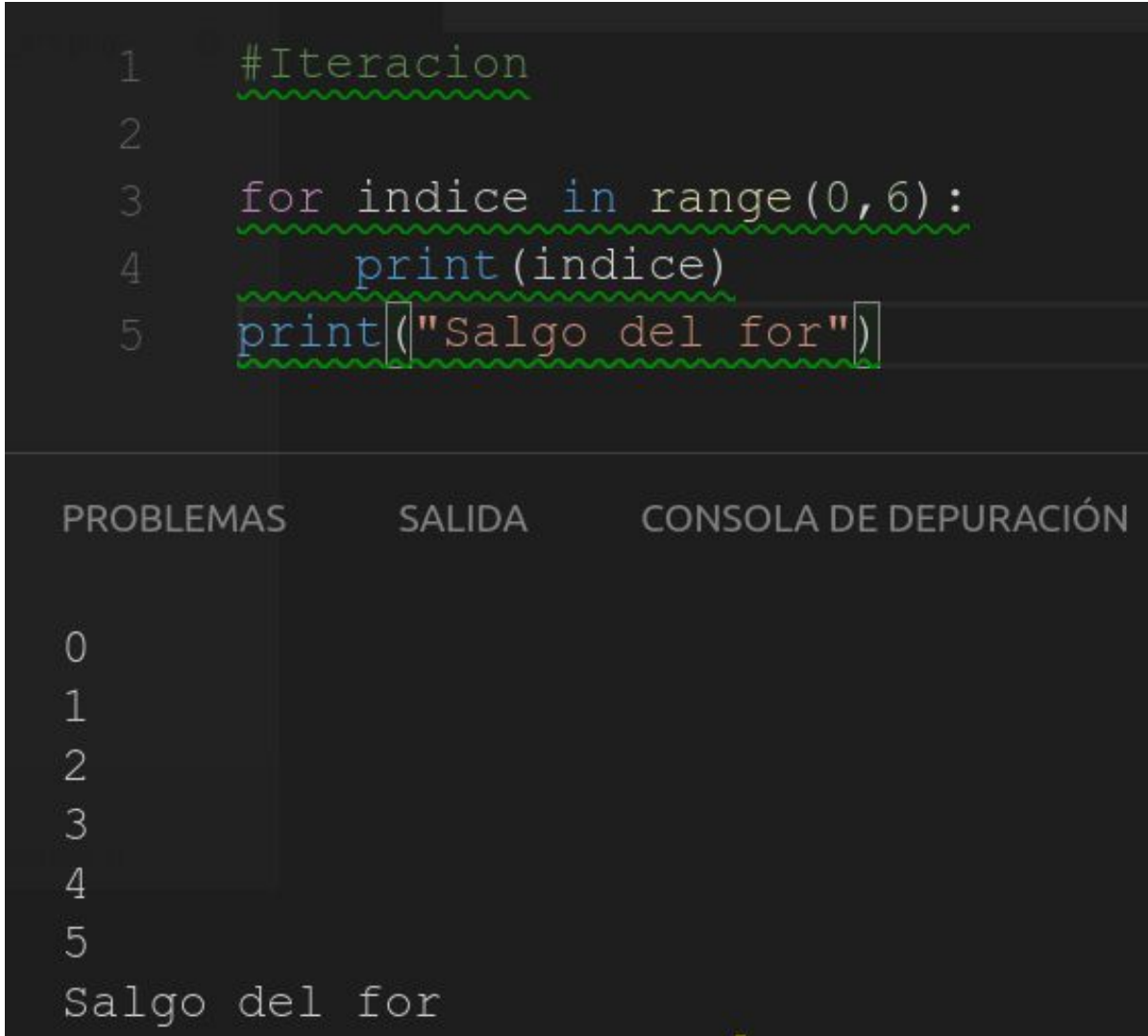
Estructuras de Control: Condicionales

```
1   #Condicionales if, elif, else
2
3   semana = ['lunes','martes','miercoles','jueves','viernes']
4
5   if (semana[3] == 'viernes'):
6     print("Es viernes")
7   elif (semana[3] == 'jueves'):
8     print("Es jueves")
9   else:
10    print("No se ni en que dia estoy")
11  print("Y salgo del if")
12
```

Estructuras de Control: Iteración

La iteración sirve para repetir una secuencia de código (un bloque) la cantidad de veces que lo necesitemos.

A diferencia de otros lenguajes, en Python el FOR itera sobre una secuencia de elementos.



for_example.py

```python
#Iteracion

semana = ['lunes','martes','miecoles','jueves','viernes']

for dia in semana:
    print(dia)
print("Salgo del for")
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL 1: Python

```
lunes
martes
miecoles
jueves
viernes
Salgo del for
```

for_example.py

```python
#Iteracion

semana = ['lunes','martes','miercoles','jueves','viernes']
responsable = ['Joaquin','Julia','Mauro','Adan','Pablo']

for dia in semana:
    indice_dia = semana.index(dia)
    responsable_dia = responsable[indice_dia]
    print("El responsable de los %s es %s." % (dia, responsable_dia))
print("Un gran poder conlleva una gran responsabilidad")
```

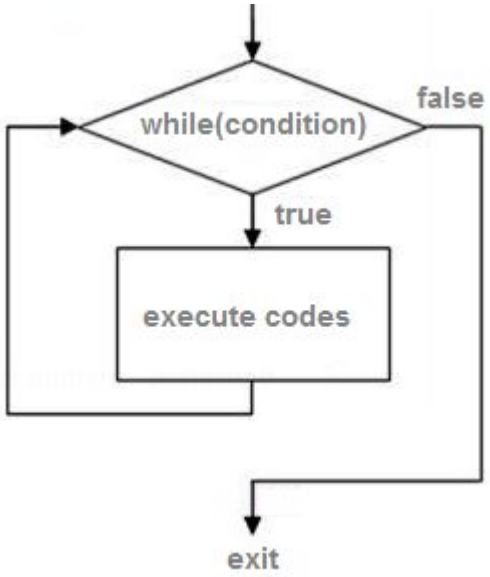
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

```
El responsable de los lunes es Joaquin.
El responsable de los martes es Julia.
El responsable de los miercoles es Mauro.
El responsable de los jueves es Adan.
El responsable de los viernes es Pablo.
Un gran poder conlleva una gran responsabilidad
```

Estructuras de Control: Bucle

Mientras que los condicionales nos permiten ejecutar distintos fragmentos de código dependiendo de ciertas condiciones, los bucles nos permiten ejecutar un mismo fragmento de código un cierto número de veces, mientras se cumpla una determinada condición.

Muy útil cuando no conozco a priori la cantidad de veces que debo repetir el bloque de código pero sí sé que en algún momento la condición se cumplirá.



for_example.py while_example.py

```python
# Bucle

num = 1246
while (num % 3 != 0):
    print("El numero %s no es divisible por 3 todavia, lo divido en 2" % (num))
    num = num / 2
print("El numero %s es divisible por 3" % (num))
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL 1: Python

```
El numero 1246 no es divisible por 3 todavia, lo divido en 2
El numero 623 no es divisible por 3 todavia, lo divido en 2
El numero 311 no es divisible por 3 todavia, lo divido en 2
El numero 155 no es divisible por 3 todavia, lo divido en 2
El numero 77 no es divisible por 3 todavia, lo divido en 2
El numero 38 no es divisible por 3 todavia, lo divido en 2
El numero 19 no es divisible por 3 todavia, lo divido en 2
El numero 9 es divisible por 3
```

Estructuras de Control: Bucle

Es muy útil y muy utilizado para generar loops infinitos que deban ser terminados por cierta condición que las variables internas al loop obtengan. Por ejemplo: una variable leída desde el teclado.

```
1   # Bucle
2
3   salir = False
4   while not salir:
5       entrada = raw_input("Hola! ")
6       if entrada == "chau":
7           print("Hasta la proxima!")
8           salir = True
9       else:
10          print("Escribiste: %s" % (entrada))
11  print("Salgo del while")
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

```
Hola! hola
Escribiste: hola
Hola! 1234
Escribiste: 1234
Hola! chau
Hasta la proxima!
Salgo del while
```

```python
#while con diccionarios

salir = False
lista_personas = []

while not salir:
    persona = {}
    persona["nombre"] = raw_input("Ingrese nombre: ")
    persona["apellido"] = raw_input("Ingrese apellido: ")
    persona["edad"] = input("Ingrese nombre: ")
    lista_personas.append(persona)
    print("Persona Agregada")
    salir = input("Terminar:1, Continuar:0 ")
print("Lista de Personas Agregadas:")
print(lista_personas)
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL 1: Python

```
Lista de Personas Agregadas:
[{'edad': 1, 'nombre': 'n1', 'apellido': 'a1'}, {'edad': 2, 'nombre': 'n2', 'apell
ido': 'a2'}, {'edad': 3, 'nombre': 'n3', 'apellido': 'a3'}]
```

Librerías o Bibliotecas???

Biblioteca (informática)

En informática, una **biblioteca** (del inglés *library*) es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca.

A diferencia de un programa ejecutable, el comportamiento que implementa una biblioteca no espera ser utilizada de forma autónoma (un programa sí: tiene un punto de entrada principal), sino que su fin es ser utilizada por otros programas, independientes y de forma simultánea. Por otra parte, el comportamiento de una biblioteca no tiene por qué diferenciarse en demasía del que pudiera especificarse en un programa. Es más, unas bibliotecas pueden requerir de otras para funcionar, pues el comportamiento que definen refina, o altera, el comportamiento de la biblioteca original; o bien la hace disponible para otra tecnología o lenguaje de programación.

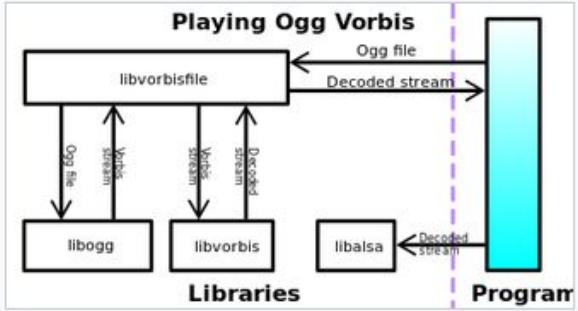


Ilustración de una aplicación que utiliza la biblioteca `libvorbisfile.so` para reproducir un archivo Ogg Vorbis.

Librerías o Bibliotecas???

Librerías o módulos estándar de python

15 librerías para Python que no te debes perder si te interesa la programación

20 librerías de Python que son sencillamente irresistibles

Aprende Git Codeacademy

Github Python

Aprende Git Codeschool