

Python
Programación II (R-123)
Licenciatura en Ciencias de la Computación

Iker M. Canut

2020

1. Introducción

Tenemos los operadores aritméticos comunes, ****** (exponente), **//** (division entera) y **%** (módulo). Los operadores booleanos son **and**, **or** y **not**. Los strings se pueden acceder con **[:]**, existe la multiplicación de strings, el append se hace **+** y en un print se pueden usar las **,** para imprimir distintos tipos.

Dentro de las funciones, se puede cambiar el orden de los argumentos: *func(par2=2, par1=1)*.

input toma como parámetro una cadena a usar como prompt y devuelve una cadena con lo ingresado. Luego, se puede usar la función **int** para castear. La vuelta es con **str**.

2. Listas e iteraciones

Una lista es una colección ordenada de elementos. Se representan con corchetes. Se pueden acceder con **[:]**. Se pueden usar números negativos para contar desde el final. Los **:** hacen *slicing*. La sintaxis es *(inicio:fin:salto)*. La resta es la cantidad, nada es default. La concatenación es con **+**.

La palabra clave **for** recorre la lista. Sintaxis: *for elemento in lista:*. Se puede agregar una condición if como el ejemplo: *for elemento in lista if [condicion]:*. La palabra clave **range** genera un rango de numeros enteros. Sintaxis: *range(inicio, fin, salto)*.

while itera mientras la condicion sea verdadera. Sintaxis: *while [condicion]:*

3. Tuplas

Una tupla es una colección ordenada de elementos. Se representan con parentesis y se accede con **[]**. La diferencia es que son inmutables, es decir, no se pueden modificar ni los elementos ni el tamaño. Si tenemos una lista de tuplas con 3 elementos, se puede hacer un for como: *for (a,b,c) in listaDeTuplas:*

4. Diccionarios

Es una estructura de datos no indexada por posición sino por clave. Las claves tienen que ser únicas e inmutables. Para agregar datos simplemente se usan las claves: *a['key']='value'*. Luego, *for x in a:* nos permite manejar las claves. Si queremos la secuencia de pares, podemos usar *for x,y in a.items():*. También tenemos el método **dict()**, que crea un diccionario tomando una lista de pares (clave, valor), **.keys()** que devuelve las claves y **.values()** que devuelve los valores.

5. Conjuntos

Para crear un conjunto se usa el metodo **set()**. En los conjuntos no importan los elementos repetidos ni el orden. Pero no se pueden ingresar elementos mutables a un set. Se puede iterar con un for. Para agregar se usa **.add()**, **.update()** agrega multiples elementos, **.discard()** borra si existe y **.remove()** borra (con excepción si no existe). Operaciones: **|** es la unión, **&** es la intersección y **-** es la diferencia.

6. Archivos

f = open([name], [mode]). Donde los modos pueden ser **"r"** (lectura), **"w"** (sobrescritura, crea si falta), **"r+"** o **"w+"** (abre para lectura y escritura) y **"a"** (abre para escribir al final). Muy importante, hay que ejecutar **.close()** para cerrar el archivo. Se agregan cosas con **.write()**, que devuelve la cantidad de caracteres. Para leerlo podemos usar **.readlines()** que devuelve una lista formada por las lineas, **.readline()** que devuelve la siguiente linea o **.read(n)** que lee n caracteres. Se van acumulando, si llegamos al final nos devuelve el string vacio, y para movernos usamos **.seek(p)** que nos mueve a p lugares del inicio.