Universidad Nacional de Colombia
Facultad de Ciencias Exactas y Naturales
Sede Manizales

Relatoría Segundo Corte

Informática III

Sofia Londoño Toro

16 de octubre de 2022

Resumen

Mediante la presente relatoría se pretende mostrar los avances y aprendizajes del segundo corte de la asignatura Informática III, donde se enfatizó en el manejo de listas, diccionarios y definición de funciones, esto con una orientación a la manipulación de bases de datos pequeñas, con el fin de entender las diferentes funciones integradas de cada tipo de dato y así manipular los datos de manera satisfactoria, según la necesidad.

Clase 5 (09/09/2022) : CICLO FOR

En python, el ciclo for se utiliza para recorrer CUALQUIER iterable, como las listas, las tuplas, strings o uno pre establecido, según la necesidad.
Notación

```
for <variable> in <iterable>:
<sentencias1>
<sentencias2>
```

<variable>: Es cualquier nombre que el programador decide usar para recorrer el ciclo for <iterable>: El cualquier elemento de Python que se puede recorrer o iterar:

- lista
- Tupla
- String
- Arreglo

Ejemplo:
Recorrer los siguientes iterables utilizando el ciclo for:

- alturas = [10,20, 50, 80, 1, 50]
- pesos = (70, 60, 55, 62, 45, 90)
- mensaje = "Hola Mundo Cruel"
- secuencia = range(1, 11, 1)

```
print('iterable 1")
for altura in alturas:
print(altura)
print('iterable 2")
for peso in pesos:
print(peso)
print(ïterable 3")
for letra in mensaje:
print(letra)
print('iterable 4")
for i in secuencia:
print(i)
alturas = [10,20, 50, 80, 1, 50]
pesos = (70, 60, 55, 62, 45, 90)
```

```
mensaje = "Hola Mundo Cruel"
secuencia = range(1, 11, 1)

print(" |n |nIterable 1")
for altura in alturas:
print(altura, end=")
print(" |n |niterable 2")
for peso in pesos:
print(peso, end=")
print(" |n |nIterable 3")
for letra in mensaje:
print(letra, end=")
print(" |n |nIterable 4")
for i in secuencia:
print(i, end=")

Ütilizar ciclo for para recorrer los anteriores iterables
sin necesidad de definirlos en las variables alturas,pesos,mensaje,
rango,"


print(" |n |n---Recorrido de iterables sin necesidad de definirlos--")

print(" |n |nIterable 1")
for altura in [10,20, 50, 80, 1, 50]:
print(altura, end=")

print(" |n |niterable 2")
for peso in (70, 60, 55, 62, 45, 90):
print(peso, end=")

print(" |n |nIterable 3")
for letra in "Hola mundo cruel":
print(letra, end=")

print("|n |nIterable 4") Esta es la manera más usada y tiene
mucha practicidad
for i in range(1,10):
print(i, end=")
```

Clase 7 (14/09/22 - 30/09/22 - 05/10/22) : ITERABLES Y SUS MÉTODOS

Es importante saber que todo elemento en python es un objeto. Y que cada objeto posee sus propios métodos métodos (funcionalidades).

Entre los objetos más comunes que se tienen están: strings, listas y diccionarios.
Cada uno de ellos posee métodos propios para ejecutar alguna funcionalidad. Por ejemplo:

cadena.count('A') => Cuenta el número de veces que se repite el caracter 'A' en la cadena
lista.append(1) => Agrega el entero 1 a la lista
diccionario.get('cristian") => Obtiene el elemento del diccionario cuya clave es çristian"

Veamos algunos de los métodos más importantes:

——strings———

- formateo: capitalize(), upper(), lower(), title(), center(spaces), strip()
- operaciones: count(subcadena), find(subcadena) replace(old, new)
- verificacion: isalnum(), isalpha(), isdigit(), isdecimal()
- Indexado: [indice]
- Slicing: [indice1 : indice2 : salto]

——Listas———

- operaciones: append(value), insert(index, value),
- pop(index), remove(value),
- count(value)
- ordenado: sort(), reverse()
- almacenamiento: clear(), copy()
- Indexado: [indice]
- Slicing: [indice1:indice2:salto]

——Tuplas———

- Operaciones: index(value), count(value)
- Indexado: [indice]
- Slicing: [indice1:indice2:salto]

——diccionarios———

- Extracción: items(), keys(), values(), get(key)
- Eliminar: pop(key)
- Almacenamiento: clear(), copy()
- Indexado: [key]

Ejemplos Cadenas:

```
Todo elemento en python es un objeto de alguna Clase (modelo)
isinstance([1,2,3], list)
isinstance('sdafa', str)
isinstance(1:2, dict)

cadena1 = 'Anita lava la tina'

cadena1.upper()  Retorna una nueva cadena en mayus
cadena1.lower()  Retorna una nueva cadena en minus
cadena1.count('a')  Retorna un entero
cadena1.isalpha()  Retorna un booleano
```

```
cadena1.isalnum()  Retorna un booleano
cadena1.isnumeric()  Retorna un booleano
cadena1.replace('Anita', 'Cristian')  Retorna una nueva cadena con reemplazos
cadena1.center(50)  Retorna una nueva cadena centrada 50 espacios

Estan retornando una nueva cadena sin alterar la original
Cadenas son inmutables (no se pueden alterar)

---------- Indexación y slicing -------------

Indexacion: Es un método para acceder a los elementos de las cadenas
slicing: Es un método para acceder a rebanadas de las cadenas

cadena2 = 'Mensaje para Informatica 3'
=> Indexacion
print("primer =>", cadena2[0])  primer elemento de la cadena
print("Segundo =>", cadena2[1])  Segundo elemento de la cadena
print("PenUltimo =>", cadena2[-2])  PenUltimo elemento de la cadena
print('Ultimo =>", cadena2[-1])  Ultimo elemento de la cadena

=>Slicing
Extraer toda la cadena al revés
print('Opcion 1: ", cadena2[-1:-27:-1])
print('Opcion 2: ", cadena2[-1::-1])
print('Opcion 3: ", cadena2[::-1])

Extraer cada tercer elemento empezando en el segundo indice
print('Opcion 1: ", cadena2[2:27:3])
print('Opcion 2: ", cadena2[2::3])
print('Opcion 3: ", cadena2[-24::3])

Extraer los dos elementos en la mitad de la cadena
print('Opcion 1: ", cadena2[12:14:1])
print('Opcion 2: ", cadena2[-14:-12:1])
```

Ejemplos Listas:

Crear las siguientes listas:
lista1 = [1,2,3,4,5,6,7,8]
lista2 = lis(range(20)) + ['a', 'b', 'c', 100]
lista3 = ['cruel', 'mundo', 'hola', 1, 100, 200, 500]

Realice lo siguiente:
- Agregar al final de lista1, los elementos =>1,2,3,4
- Agregar al comienzo de lista1, los elementos =>0,0,2,4
- Eliminar los 3 ultimos elementos de lista1
- Eliminar los 3 últimos elementos de lista1
- Eliminaro los 2 primeros elementos de liista 1

- Sume todos los elemtnos de lista2 que pueden operarse algebráicamente

- Organice lista2 de forma ascendente
- Elimine todos los elementos de la lista2 original que son enteros
- Sume todos los elemtnos de lista3 que pueden operarse algebraicamente sin alterar ni cambiar la lista original.
- Haga una copia de lista3 de la siguiente manera:
lista3Copia = lista3
Y agregue 3 nuevos elementos sobre esta nueva lista. ¿Qué sucede con la lista3 original?
- Encuentre una manera de alterar la copia sin afectar la lista original.

INDEXADO:
- Extraer elemento inicial de lista1 (de dos maneras)
- Extrear elemento final de lista1 (de dos maneras)
- Extraer el elemento del medio de lista2 (de dos maneras)

SLICING:
- Extraer los primeros 3 elementos de lista1, ista2 y lista3 y crear una nuevaLista
- Extraer cada 2 elementos de lista 2
- Extraer todos los elementos de lista3 al revés
- Extraer los elementos de lista3 ubicados en indices impares

```
Creación de las listas:

lista1 = [1,2,3,4,5,6,7,8]
lista2 = list(range(20)) + ['a', 'b', 'c', 100]
lista3 = ['cruel', 'mundo', 'hola', 1, 100, 200, 500]

for i in [1,2,3,4]:
lista1.append(i)
print(lista1)

for i in [4, 2, 0, 0]:
for j in range(0, 3, 1):
lista1.insert(j, i)
break
print(lista1)

for i in [15, 12, -1]:
lista1.pop(i)  Por definición elimina el ultimo valor,
se puede mirar con help(lista1.pop())
print(lista1)

lista1.pop(0)
lista1.pop(0) print(lista1)

Suma algebraica de la lista:
lista2.remove('a')
lista2.remove('b')
lista2.remove('c')
resultado = sum(lista2)
print(lista2, resultado)
```

```
Lista 2 de forma ascendente
lista2.sort(reverse = True) Por defecto organiza de manera ascendente, para hacerlo
descendente hay que usar el parámetro reverse
print(lista2)

Elimine todos los elementos de la lista2 original que son enteros:
listaApoyo = []
for elemento in lista2:
if type(elemento) == str:
listaApoyo.append(elemento)
else:
pass En el otro caso no hacer nada
lista2 = listaApoyo
print(lista2)

Sume todos los elemtnos de lista3 que pueden operarse algebraicamente sin alterar
ni cambiar la lista original.

Creamos una copia de la lista:
lista3Copia = lista3.copy() Es diferente a lista3Copia = lista porque esto es una
reasignación de nombre
La que hicimos si es una copia

lista3Copia.append('Sofia')
lista3Copia.append('Londoño')
lista3Copia.append('Toro')

print(lista3) => Sigue intacta
print(lista3Copia)

INDEXADO:
lista1 = [1,2,3,4,5,6,7,8]
lista2 = list(range(20)) + ['a', 'b', 'c', 100]
lista3 = ['cruel', 'mundo', 'hola', 1, 100, 200, 500]

print(lista1[0], lista1[-8])  Extraer elemento inicial
print(lista1[-1], lista1[7])  Exraer elemento final
print(lista2[11], lista2[-13])  Extraer elemento del medio

print(lista1[0:3] + lista2[0:3] + lista3[0:3])  Extraer los 3 primeros indicies
de cada lista y sumarlos
print(lista2[::2])  Extraer los elementos de dos en 2 de principio a fin
print(lista3[::-1])  Extraer todos loes elementos al revés
print(lista3[1::2])  Extraer solo los indices impares
```

Ejemplos Diccionarios:

```
diccionariosEstudiantes = 'Cristian Pachon': 2.0,
'Maria Bermudez' : 3.0,
```

```
'Camilo Ibarra' : 4.0,
'Fernanda Gutierrez' : 5.0

diccionarioPaises = 'Colombia' : ['Cali', 'Manizales', 'Cartagena'],
'Argentina' : ['Buenos Aires', 'La Plata', 'Cordoba'],
'Brasil' : ['Sao Pablo', 'Brasilia', 'Minas Gerais']

dicccionarioNumeros = 1 :'uno',
2 :'dos',
3 :'tres',
4 :'cuatro',
5 :'cinco'
```

¿Cómo extraer un valor dada una clave? Extraer del diccionario estudiantes el valor de la clave Camilo Ibarra

```
valor = diccionariosEstudiantes['Camilo Ibarra']
print(valor)
Imprimir todas los valores de diccionario estudiantes
for clave in diccionariosEstudiantes:
print(diccionariosEstudiantes[clave])
```

¿Cómo eliminar un par clave-valor?
Eliminar los elementos coya clave empiece por la letra C en el diccionarios Estudiantes

```
for clave in ['Cristian Pachon', 'Maria Bermudez',
'Camilo Ibarra', 'Fernanda Gutierrez']:
if clave[0] == 'C':
diccionariosEstudiantes.pop(clave)
print('-', diccionariosEstudiantes)
```

¿Cómo cambiar un valor?
''Cambiar la calificacion de diccionarioEstudiantes así: -Mujeres: 5.0 -Hombres: 0.0 '''

```
for clave in diccionariosEstudiantes:
if clave.split()[0][-1] == 'a':
diccionariosEstudiantes[clave] = 5.0
else:
diccionariosEstudiantes[clave] = 0.0
print(diccionariosEstudiantes)
```

¿cómo extraer todas las claves de un diccionario?

```
claves = diccionariosEstudiantes.keys()
print(claves)
```

¿ Cómo extraer todos los valores de un diccionario?

```
valores = diccionariosEstudiantes.values()
print(valores)

ÏMPRIMIR TODAS LAS PAREJAS CLAVE-VALOR DE diccionarioEstudiantes
DE 2 MANERAS (utilice ciclo for)
ejemplo =>Maria Bermudez-5.0, Fernanda Gutierrez-5.0 "

for item in diccionariosEstudiantes:
print(clave, valor)

¿ Como extraer los pares clave-valor?
parejas = diccionariosEstudiantes.items()
print(parejas)
```

EJERCICIOS CLASE

1) Crear la siguiente base de datos utilizando diccionarios Costo de entrada al cine
Niños Adultos
Entre semana 5000 7000
Fin de semana 8000 10000
* ¿Cómo acceder al precio de la entrada utilizando como claves si es niño-adulto y si es Entre Semana-Fin de semana
CostoCine = { 'Entre Semana' : { 'Niños': 5000, 'Adultos': 7000 },
'Fin de semana' :{'Niños': 8000, 'Adultos': 10000 } }
print(CostoCine['Entre Semana']['Niños'])

2) Crear la siguiente base de datos utilizando diccionarios:

-

-

-

-

-

-

-

-

-

*Cómo acceder a la base de datos utilizando como clave el codigo y la materia
*Determine el promedio de cada estudiante
*Determine el promedio de cada una de las materias
*Determine los 3 estudiantes con peor promedio

Cod Estudiante	Ingles	Deportes	Idiomas	Cuantica	Español
01	2.0	2.2	4.2	4.0	0.5
02	2.2	1.0	4.0	3.1	4.0
03	2.9	4.2	3.1	0.0	3.1
04	2.0	4.0	4.0	0.2	0.0
05	2.2	0.2	0.2	1.0	0.2
06	2.0	5.0	1.0	1.3	1.0
07	5.0	1.2	1.2	1.9	1.3
08	0.2	2.9	1.0	4.2	1.9
09	5.0	2.3	2.9	2.9	0.2
10	4.2	5.0	4.2	4.2	3.9
11	4.5	4.2	4.0	0.5	4.2
12	4.2	4.5	4.2	0.0	0.5
13	0.5	0.5	2.3	4.2	0.0
14	4.1	3.1	2.5	4.3	3.2
15	4.2	4.2	4.2	2.5	4.3
16	4.1	0.0	4.5	4.2	2.5
17	1.2	3.1	0.5	4.5	3.2
18	0.5	0.2	4.1	4.1	4.5
19	2.2	0.5	0.2	0.2	4.1

Cuadro 1: Base de Datos, Diccionario

```
NotasMaterias = { '01': { 'Inglés': 2.0, 'Deportes': 2.2, 'Idiomas':
4.2, 'Cuantica': 4.0,'Español': 0.5 },
'02': { 'Inglés': 2.2, 'Deportes': 1.0, 'Idiomas':
4.0, 'Cuantica': 3.1,'Español': 4.0 },
'03': { 'Inglés': 2.9, 'Deportes': 4.2, 'Idiomas': 3.1,
'Cuantica': 0.0,'Español': 3.1 },
'04': { 'Inglés': 2.0, 'Deportes': 4.0, 'Idiomas': 4.0,
'Cuantica': 0.2,'Español': 0.0 },
'05': { 'Inglés': 2.2, 'Deportes': 0.2, 'Idiomas': 0.2,
'Cuantica': 1.0,'Español': 0.2 },
'06': { 'Inglés': 2.0, 'Deportes': 5.0, 'Idiomas': 1.0,
'Cuantica': 1.3,'Español': 1.0 },
'07': { 'Inglés': 5.0, 'Deportes': 1.2, 'Idiomas': 1.2,
'Cuantica': 1.9,'Español': 1.3 },
'08': { 'Inglés': 0.2, 'Deportes': 2.9, 'Idiomas': 1.0,
'Cuantica': 4.2,'Español': 1.9 },
'09': { 'Inglés': 5.0, 'Deportes': 2.3, 'Idiomas': 2.9,
'Cuantica': 2.9,'Español': 0.2 },
'10': { 'Inglés': 4.2, 'Deportes': 5.0, 'Idiomas': 4.2,
'Cuantica': 4.2,'Español': 3.9 },
'11': { 'Inglés': 4.5, 'Deportes': 4.2, 'Idiomas': 4.0,
'Cuantica': 0.5,'Español': 4.2 },
'12': { 'Inglés': 4.2, 'Deportes': 4.5, 'Idiomas': 4.2,
'Cuantica': 0.0,'Español': 0.5 },
'13': { 'Inglés': 0.5, 'Deportes': 0.5, 'Idiomas': 2.3,
```

```
'Cuantica': 4.2,'Español': 0.0 },
'14': { 'Inglés': 4.1, 'Deportes': 3.1, 'Idiomas': 2.5,
'Cuantica': 4.3,'Español': 3.2 },
'15': { 'Inglés': 4.2, 'Deportes': 4.2, 'Idiomas': 4.2,
'Cuantica': 2.5,'Español': 4.3 },
'16': { 'Inglés': 4.1, 'Deportes': 0.0, 'Idiomas': 4.5,
'Cuantica': 4.2,'Español': 2.5 },
'17': { 'Inglés': 1.2, 'Deportes': 3.1, 'Idiomas': 0.5,
'Cuantica': 4.5,'Español': 3.2 },
'18': { 'Inglés': 0.5, 'Deportes': 0.2, 'Idiomas': 4.1,
'Cuantica': 4.1,'Español': 4.5 },
'19': { 'Inglés': 2.2, 'Deportes': 0.5, 'Idiomas': 0.2,
'Cuantica': 0.2,'Español': 4.1 } }

Para acceder:
print(NotasMaterias['06']['Cuantica'])

Para sacar los promedios por estudiantes:
for cod in NotasMaterias.keys():
print(NotasMaterias[cod]) Recorro los diccionarios de las notas
print('PROMEDIO DEL ESTUDIANTE ' , cod, (NotasMaterias[cod]['Inglés'] +
NotasMaterias[cod]['Deportes'] + NotasMaterias[cod]['Idiomas'] +
NotasMaterias[cod]['Cuantica'] + NotasMaterias[cod]['Español'])/5)

Para sacar promedios por asignatura:

acumIngles = 0
acumDeportes = 0
acumIdiomas = 0
acumCuantica = 0
acumEspañol = 0

for cod in NotasMaterias.keys():
acumIngles += NotasMaterias[cod]['Inglés']
acumDeportes += NotasMaterias[cod]['Deportes']
acumIdiomas += NotasMaterias[cod]['Idiomas']
acumCuantica += NotasMaterias[cod]['Cuantica']
acumEspañol += NotasMaterias[cod]['Español']

print('Acumulado Inglés', acumIngles / len(NotasMaterias))
print('Acumulado Deportes', acumDeportes / len(NotasMaterias))
print('Acumulado Cuantica', acumCuantica / len(NotasMaterias))
print('Acumulado Idiomas', acumIdiomas / len(NotasMaterias))
print('Acumulado Español', acumEspañol / len(NotasMaterias))

Para sacar los 3 peores promedios:
promedios =
for cod in NotasMaterias():
promedios[cod] = (NotasMaterias[cod]['Inglés'] +
NotasMaterias[cod]['Deportes'] +
```

```
NotasMaterias[cod]['Idiomas'] +NotasMaterias[cod]['Cuantica'] +
NotasMaterias[cod]['Español'])/5

minimio = ''
promedioMin = 1000
for cod in NotasMaterias():
if promedios[cod] <promedioMin:
promedioMin = promedios[cod]
minimo = cod

print('El estudiante con el menor promedio es ', minimo, 'con calificacion: ',
promedioMin)
```

Clase 8 (07/10/22): FUNCIONES

Son un paradigma de programación, para cuando tengo una linea de código que se repite constantemente, con el fin de hacer el código más compacto de manera modular.

NOTACIÓN:

- Función sin entrada y sin salida:

def nombreDeLaFuncion():
pass

- Función con entrada y salida:

def nombreFuncion(variable1): ->variable1 es la entrada
resultado = variable1 ** 2 ->se opera la entrada
return resultado ->se muestra la salida para almacenarla

- Función con multiples entradas y salidas:

def nombreFuncion2(variable1, variable2, variable3): ->variable1... es la entrada
resultado = variable1 ** 2 ->se operan las entradas
resultado2 = str(variable2) + str(variable3)
return [resultado, resultado2] ->se muestra la salida

- Función con parámetros definidos:

def nombreFuncion3(variable1 = 1, variable2 = 2): ->variable1... es la entrada
return 'primero': variable1, 'segundo': variable2 ->se muestra la salida

=>Las salidas pueden tener cualquier forma, clase

Las funciones se llaman de la siguiente manera:
res1 = nombreDeLaFuncion() ->devuelve un None, que es un tipo de dato class 'NoneType'
res2 = nombreFuncion(3)
res3lista = nombreFuncion2(3, 2, 1)

res4Dic = nombreFuncion3() ->usa los parametros definidos
res4Dic = nombreFuncion3(2, 3)

Ejemplos:

```
def mayoriaEdad(edad):
if edad >= 18:
Ans = True
elif edad <18:
Ans = False
return Ans => Permite almacenar la respuesta

mayoriaEdad(23) => Se aplica la función

def salarios(nombre, salario):
return 'Hola '+ nombre + 'su salario es ' + str(salario)
salarios('Juan', 12000)
```