



Caratula para entrega de Prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: _____ Ing. Marco Antonio Martinez Quintana _____

Asignatura: __ Estructura de Datos y Algoritmos I (1227) __

Grupo: _____ 17 _____

No. de Práctica(s): _____ 9° _____

Integrante(s): _____ Avila Laguna Ricardo _____

No. de Equipo de
cómputo empleado: _____ 10 _____

No. Lista o Brigada: _____ 6 _____

Semestre: _____ 2° _____

Fecha de entrega: _____ Marzo del 2020 _____

Observaciones: _____

CALIFICACIÓN: _____

1° Objetivos

Aplicar las bases del lenguaje de programación Python en el ambiente de Jupyter notebook.

2° Introducción

En esta práctica se iniciara un repaso del lenguaje de programación Python, por lo que veremos algunos temas como variables, tipo de datos, cadenas, operadores(aritméticos, booleanos y comparativos), listas, tuplas, diccionario, funciones y variables globales.

3° Desarrollo y Resultados

Actividades:

- **Insertar y ejecutar código en las celdas de la notebook**
- **Insertar texto en las celdas de la notebook**
- **Declarar variables**
- **Declarar cadenas**
- **Aplicar operadores**
- **Crear y manipular listas, tuplas y diccionarios**
- **Crear y ejecutar funciones**

Código:

```
#Como en C las variable inician en 0
#iniciando variables
x=10
print(x)
print("\n")
#Se pueden utilizar comillas simples o dobles en creacion de una cadena
cadena = "Hola Mundo"
print(cadena)
cadena_1 = 'Hola Mundo'
print(cadena_1)
print("HOLA")
print("\n")
#Asignar un valor a 3 ó más variables
x=y=z=10
print(x,y,z)
print("\n")
#Variable type permite conocer el tipo de variable
b = 10
print(type(b))
print(type(cadena))
print("\n")
#Se pueden cambiar los valores de las variables y su tipo cambia automaticamente
x = 'Hola Mundo!'
print(x)
print(type(x))
cadena = 10
print(cadena)
print(type(cadena))
print("\n")
#Cuando una variable tienen una cantidad constante se escribe en mayusculas
SEGUNDOS_POR_DIA = 60*60*24
print(SEGUNDOS_POR_DIA)
PI = 3.14159
print(PI)
print("\n")
```

Para el primer programa realizado aprendimos cómo asignarle un valor a una variable que nosotros declaremos con la importancia de no utilizar las “palabras reservadas”, al igual nos dimos cuenta de las diferencias que tiene con el lenguaje C ya que aquí no es necesario especificar el tipo de las variables..

```

#Iniciando cadenas
cadena_1 = "Hola"
cadena_2 = 'Mundo'
print(cadena_1)
print(cadena_2)
concatenar_cadenas = cadena_1 + cadena_2#HolaMundo
concatenar_cadenas = cadena_1+' '+cadena_2#Hola Mundo
print(concatenar_cadenas)
print("\n")
#Concatenar cadenas con numeros, se tienen que utilizar str(<numero>) o str(<variable>)
numero_cadena = concatenar_cadenas+' '+str(3)
print(numero_cadena)
numero_cadena_2 = concatenar_cadenas+' '+str(y)
print(numero_cadena_2)
print("\n")
#Otra forma de concatenar cadenas y numeros es el uso de .format(<cadena>,<variable>)
numero_cadena_3 = "{} {} {}".format(cadena_1,cadena_2,7) #{} se va a sustituir por el valor de la variable
print(numero_cadena_3) #Las llaves van separadas por pares
print("{} {} {}".format(cadena_1,cadena_2,7))#Otra forma de imprimir
print("\n")
#Se puede cambiar el orden en como se imprimen las variables
numero_cadena_4 = "Cambiando el orden: {2} {1} {0}".format(cadena_1,cadena_2,20)
print(numero_cadena_4)
print("\n")

#LISTAS []
#Listas simples
lista_Dias = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22]
print(lista_Dias) #Imprime toda la lista
print(lista_Dias[0]) #Imprime el 1° elemento
print(lista_Dias[7]) # Imprime el 6° elemento
print("\n")
#Declaracion de listas anidadas
lista_num = [['cero',0],['uno',1,'UNO'],['dos',2],['tres',3],['cuatro',4],['X',5]]
print(lista_num) #Imprime toda la lista
print(lista_num[0]) #Imprime todo el elemento 0
print(lista_num[1]) #Imprime todo el elemento 1
print(lista_num[1][0]) #Imprime el elemento 0 de la lista 1
print(lista_num[1][1]) #Imprime el elemento 1 de la lista 1
print(lista_num[1][2]) #Imprime el elemento 2 de la lista 1
#Cambiar valor de los elementos de la lista (MUTABILIDAD)
lista_num[5][0] = 'cinco'
print(lista_num[5])

```

En el segundo programa empezamos declarando cadenas de caracteres para entender su manejo dentro del lenguaje de programación Python, ya que algunas de las funciones son que podemos modificar el orden en que aparecen las cadenas, al igual concatenamos varias cadenas junto con números e iniciamos la parte de listas con el manejo de las mismas para el manejo de datos..

```
*Practica9.py - C:\Users\Marbella\Documents\ProyectosPython\Practica9.py (3.8.2)*
File Edit Format Run Options Window Help

#TUPLAS (No mutables) ()
#as tupla consumen meno memoria que las listas
tupla_Dias = (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22)
print(lista_Dias) #Imprime toda la tupla
print(lista_Dias[0]) #Imprime el 1º elemento
print(lista_Dias[11]) # Imprime el 10º elemento
print("\n")

#Declaracion de tuplas anidadas
tupla_num = (('cero',0),('uno',1,'UNO'),('dos',2),('tres',3),('cuatro',4),('X',5))
print(tupla_num) #Imprime toda la lista
print(tupla_num[0]) #Imprime todo el elemento 0
print(tupla_num[1]) #Imprime todo el elemento 1
print(tupla_num[1][0]) #Imprime el elemento 0 de la lista 1
print(tupla_num[1][1]) #Imprime el elemento 0 de la lista 1
print(tupla_num[1][2]) #Imprime el elemento 0 de la lista 1
#Comparando la mutabilidad entre listass y tuplas
print("Valor actual (Lista) {}".format(lista_Dias[0]))
lista_Dias[0] = 50
print("Valor Actualizado (Lista) {}".format(lista_Dias[0]))
print("Valor actual (Tupla) {}".format(tupla_Dias[0]))
#tupla_Dias[0] = 50 #NO se puede por lo que manda ERROR
#print("Valor Actualizado (Tupla) {}".format(tupla_Dias[0]))#NO se puede por lo q
print("\n")

#Tuplas con Nombre
#Modulo "NAMEDTUPLE"
from collections import namedtuple
#Creamos la tupla y que datos tiene <Nombre> = namedtuple('Nombre',['Campo','Campo
planeta = namedtuple('planeta',['nombre','numero'])
planeta_1 = planeta('Mercurio',0)
print(planeta_1)
planeta_2 = planeta('Venus',1)
print(planeta_2)
#Imprimir los valores de los campos
print(planeta_1.nombre,planeta_1.numero)
print(planeta_2[0],planeta_2[1])
#Imprimir los campos de la tupla que pide
print('Campos de la tupla: {}'.format(planeta_1._fields))
print("\n")
```

Para el tercer programa aprendimos a utilizar las tuplas las cuales son muy parecidas a las listas pero aquí nosotros podemos asignarles nombre y personalizarlas más, con el único defecto que no son mutables lo que significa que no podemos cambiar el valor o tipo de las variables declaradas dentro de las tuplas.


```
*Practica9.py - C:\Users\Marbella\Documents\ProjetsPython\Practica9.py (3.8.2)*
File Edit Format Run Options Window Help

#Diccionarios {}
#Diccionario simple
tabla_periodica = {'hidrogeno':1,'helio':2,'carbono':6}
print(tabla_periodica) #Todo el diccionario #Pueden aarecen en desorden
print(tabla_periodica['helio']) #El valor del nombre del elemento
#Agregar elementos al diccionario
tabla_periodica['litio'] = 3
tabla_periodica['nitrogeno'] = 8
print(tabla_periodica)
#Nuevo diccionario LLAVE
elementos = {}
elementos['H'] = {'name':'Hydrogen','number':1,'weight':'1.00794'} #1° Elemento
elementos['He'] = {'name':'Helium','number':2,'weight':'4.002602'} #2° Elemento
print(elementos) #Todos
print(elementos['H']) #Todo el elemento H
print(elementos['H']['name']) #Solo el nombre del elemento H
print(elementos['H']['number']) #Solo el numero del elemento H
elementos['H']['weight'] = 4.3 #Cambiamos el dato del elemento H
print(elementos['H']['weight'])
#Agregar elementos a una llave
elementos['H'].update({'gas noble':True}) #diccionario[llave]=update({'Nuevo_campo':valor})
print(elementos['H'])
#Mostrar todos los elementos del diccionario
print(elementos.items()) #diccionario.items()
#Muestra todas las llaves del diccionario
print(elementos.keys()) #diccionario.key()
print("\n")

#Funciones
def imprimir_nombre(nombre,apellidol,apellido2):
    print("Hola "+nombre+" "+apellidol+" "+apellido2)
    return 0

imprimir_nombre("Ricardo","Avila","Laguna")

def cuadrado(x):
    return x**2

x=5
print("El cuadrado de {} es {}".format(x,cuadrado(x)))

def varios(x):
    return x**2,x**3,x**4
```

Ln: 36 Col:

Para el cuarto y último programa aprendimos lo que son los diccionarios los cuales son muy idénticos a las listas y a las tuplas por lo que igual nos sirve para emplearla de manejo de datos, en este programa igual realizamos funciones para comprender su estructura y poder emplearlas cuando programemos..

4° Conclusiones

Avila Laguna Ricardo :

Los objetivos se cumplieron ya que aprendimos las bases de el lenguaje de programación Python, en lo que consta aprendimos las palabras reservadas, la asignación en las variables, el manejo de cadenas, concatenación de cadenas, las lista, tuplas y diccionarios, así mismo aprendimos de la funciones y de la importancia de las variables globales tanto com locales.

Bibliografía

<http://lcp02.fi-b.unam.mx/>