

PYTHON AVANZADO



Aprendices

Cesar Ivan Pinzon

Tutor

Erick Granados

**SERVICIO NACIONAL DE APRENDIZAJE SENA
MODALIDAD PRESENCIAL
CENTRO DE GESTIÓN DE MERCADOS, LOGÍSTICA
Y TECNOLOGÍA DE LA INFORMACIÓN
7 OCTUBRE**

TALLER ESTRUCTURA DE DATOS

INTRODUCCIÓN

En esta actividad conoceremos y utilizaremos métodos y técnicas especializadas en la estructuración y manejo de datos, empezando por un tema ya visto anteriormente como lo son las listas, este tipo de dato que es especialista en almacenar otros datos en sí misma

1.MÉTODOS EN LISTA

Como ya conocemos su definición, iremos directamente con los métodos de manejo de este tipo de dato.

Primeramente debemos conocer el método “append”, este utilizado para agregar un dato o varios (estos mismos se almacenarán al final de las listas), un uso más cercano de este método lo podemos observar a continuación :

>>nombre lista.append(“dato a agregar”)

Así mismo encontramos el método extend que cumple mas misma función pero pudiendo agregar más datos iterativos.

>>nombre lista.extend(“datos a agregar”)

Seguido a esto encontramos el método “insert”, ahora utilizado para agregar datos y ubicarlos en una posición específica, la sintaxis utilizada para este será la siguiente:

>>nombre lista.insert(“ubicación que reemplazará el nuevo dato”,”dato a agregar”)

Dentro de los métodos de consulta (buscar y brindar información de la lista), encontramos el método sort(organiza alfabética/Numéricamente la lista), count(muestra cuántas veces se repite un dato), index(da a conocer la posición que un dato ocupa en la lista), entre otros, la sintaxis de estas las veremos en los ejemplos adjuntos a la carpeta

2.LISTAS EN PILAS

“Último en entrar, primero en salir”, con esto podemos definir a una pila, y esta función la podemos realizar con una lista, utilizando el método Append agregaremos un elemento al final de la lista, y seguido de esto, con el método pop, podremos eliminar el último elemento agregado.

3.LISTAS EN COLAS

Las colas tienen como función lo contrario a las pilas, se basan en “El primero en entrar, primero en salir”, esto si bien es más complicado de aplicar con una lista, podremos hacer uso de una librería como lo es “collections” para hacernos más sencillo el trabajo, pudiendo utilizar así el método popleft que eliminará el primer elemento que encuentre, en este caso partiendo del lado izquierdo.

4.COMPRENSIÓN DE LISTAS

La comprensión de listas es una manera de generar y crear listas basadas en algoritmos o operaciones básicas. Una lista de comprensión consiste de corchetes rodeando una expresión seguida de la declaración for y luego cero o más declaraciones for o if. El resultado será una nueva lista que sale de evaluar la expresión en el contexto de los for o if que le siguen.

5.INSTRUCCIÓN “DEL”

El método del es similar al método pop, pero permite eliminar en base a la posición de un elemento en la lista, de igual forma permite eliminar un segmento de esta, todos los elementos de la lista, e incluso, una variable del programa.

6. TUPLAS Y SECUENCIAS

Las tuplas por el contrario de las listas, son inmutables, sin embargo poseen elementos que si los son, se representan dentro de paréntesis, y para definir los elementos dentro de una tupla, deben de encontrarse separados por una coma, de igual forma se pueden empaquetar elementos en una variable, esto directamente convirtiéndola en una tupla.

7.CONJUNTOS

La creación de conjuntos se basan en el método set(), este tipo de dato, almacena otros datos irrepetibles y sin ordenar, esto permite de igual forma, métodos de consulta como observar datos que se encuentren solo en un conjunto, o todos los de un solo conjunto, etc.

8.DICCIONARIOS

Los diccionarios son otro tipo de colección que encontramos en python, estos permiten almacenar datos anidados, esto normalmente utilizado para almacenar key and values o llaves y valores, por ejemplo: “Computador”: 2000. Estos se inicializan con llaves y la separación de una llave y un valor se realiza con comas, de igual forma se pueden usar métodos de consulta para comprobar las llaves o saber si una está o no en dicho diccionario.

9.TÉCNICAS DE ITERACIÓN

Las técnicas de iteración nos permiten realizar una consulta de una o más colecciones mientras una condición se cumpla, o por una cantidad definida de veces, esto gracias a los métodos for y while, estas técnicas y consultas varían en base a qué tipo de colección se usa (diccionarios, tuplas o listas)

10. CONDICIONES

Las condiciones que podemos utilizar en los métodos anteriormente mencionados, pueden ser si bien comparaciones o operaciones, esto utilizando los operadores booleanos como lo son or, and, xor or not, siendo los primeros dos también llamados operadores cortocircuito, de igual forma podemos guardar el resultado de una comparación en una variable.

11. COMPARACIÓN DE SECUENCIAS

La comparación de secuencias se puede hacer mediante el orden lexicográfico, esto según sean strings o enteros, el tipo de dato que dichas colecciones posean

CONCLUSIÓN

En este taller nos adentramos más en las funciones de almacenamiento y control de datos que podemos hacer con este lenguaje de programación, conocimos una librería nueva como lo fue collections, y de igual forma términos como la lexicografía fueron dados a conocer, y dentro del tema tratado, es verdad que como se menciona en la documentación, python sigue siendo un lenguaje aún en desarrollo y que probablemente implemente más colecciones, consultas u operadores.