COLOMBIA POTENCIA DE LA VIDA

TIC

Módulo 1 - Unidad 2 - Lección 1

Actividad 4

Contenedores en Python

TALENTO TECH

AZ | PROYECTOS EDUCATIVOS

UTP Universidad Tecnológica de Pereira

COLOMBIA POTENCIA DE LA VIDA

TIC

Contenedores en Python

- Presentación de estructuras de datos como listas, tuplas y diccionarios.
- Operaciones y métodos básicos para manipular contenedores.

Listas

Tuplas

Diccionario

Ejercicios

Listas

En Python, una lista es un contenedor flexible y mutable que permite almacenar una colección ordenada de elementos.

Declaración de una Lista: pueden declarar una lista utilizando corchetes [ ].

```
mi_lista = [1, 2, 3, 'a', 'b', 'c']
```

Índices y Acceso a Elementos: los elementos de una lista se numeran desde 0 en adelante. Puedes acceder a un elemento mediante su índice.

```
primer_elemento = mi_lista[0]  # Resultado: 1
tercer_elemento = mi_lista[2]  # Resultado: 3
```

Listas

Longitud de una Lista: pueden obtener la longitud de una lista con la función len( ).

```
longitud = len(mi_lista)  # Resultado: 6
```

Modificación de Elementos: las listas son mutables, lo que significa que pueden modificar sus elementos.

```
mi_lista[3] = 'x'
# Resultado: [1, 2, 3, 'x', 'b', 'c']
```

Listas

Operaciones Comunes:

Pueden realizar diversas operaciones con listas como agregar elementos, eliminar elementos, extender una lista con otra, etc.

```
# Agregar elementos al final de la lista
mi_lista.append(4)
# Resultado: [1, 2, 3, 'x', 'b', 'c', 4]
```

```
# Eliminar elemento por valor
mi_lista.remove('b')
# Resultado: [1, 2, 3, 'x', 'c', 4]
```

```
# Extender la lista con otra lista
otra_lista = [5, 6]
mi_lista.extend(otra_lista)
# Resultado: [1, 2, 3, 'x', 'c', 4, 5, 6]
```

Listas

Slicing (Rebanado): pueden obtener porciones de una lista utilizando la técnica de "slicing".

```
sublista = mi_lista[1:4]  # Resultado: [2, 3, 'x']
```

Inclusión de Elementos: pueden verificar si un elemento está presente en una lista con el operador in.

```
existe = 'c' in mi_lista  # Resultado: True
```

Las listas son una herramienta versátil en Python y se utilizan comúnmente para almacenar y manipular colecciones de datos.

Tuplas

En Python, una tupla es otro tipo de contenedor, similar a una lista, pero con la diferencia principal de que es inmutable, lo que significa que no puedes modificar su contenido después de haber sido creada.

Declaración de una Tupla: pueden declarar una tupla utilizando paréntesis ().

```
mi_tupla = (1, 2, 3, 'a', 'b', 'c')
```

Índices y Acceso a Elementos: los elementos de una tupla también se numeran desde 0 en adelante. Pueden acceder a un elemento mediante su índice de la misma manera que con las listas.

```
primer_elemento = mi_tupla[0]  # Resultado: 1
tercer_elemento = mi_tupla[2]  # Resultado: 3
```

Tuplas

Longitud de una Tupla: Pueden obtener la longitud de una tupla con la función len() al igual que con las listas.

```
longitud = len(mi_tupla)  # Resultado: 6
```

Inmutabilidad: la principal diferencia con las listas es que las tuplas son inmutables, lo que significa que no pueden agregar, eliminar o modificar elementos una vez que la tupla ha sido creada.

```
# Esto generará un error
mi_tupla[0] = 'x'
```

Empaquetado y Desempaquetado: pueden "empaquetar" varios valores en una tupla y luego "desempaquetar" esos valores en variables individuales.

```
coordenadas = (10, 20)
x, y = coordenadas
# Ahora x es 10 y y es 20
```

Tuplas

Uso en Iteraciones: las tuplas se utilizan comúnmente en situaciones donde se necesita una estructura de datos inmutable, como claves en un diccionario o elementos en un conjunto.

Tuplas Anidadas: pueden tener tuplas dentro de tuplas, creando así estructuras de datos más complejas.

```
tupla_anidada = ((1, 2), ('a', 'b'))
```

Las tuplas son útiles cuando necesitan asegurarse de que los datos no cambien durante la ejecución de su programa, y se utilizan comúnmente en situaciones donde la inmutabilidad es esencial.

Diccionario

En Python, un diccionario es un tipo de contenedor que almacena pares clave-valor, cada valor en un diccionario está asociado a una clave única que actúa como identificador. Aquí les explico los aspectos claves de los diccionarios en Python:

Declaración de una Diccionario: pueden declarar un diccionario utilizando llaves {}.

```
mi_diccionario = {'clave1': 'valor1', 'clave2': 'valor2', 'clave3': 'valor3'}
```

Acceso a Elementos por Clave: acceda a los elementos de un diccionario utilizando sus claves.

```
valor_asociado_a_clave1 = mi_diccionario['clave1']
```

```
# Resultado: 'valor1'
```

Diccionario

Modificación y Adición de Elementos: pueden modificar el valor asociado a una clave existente o agregar nuevas parejas clave-valor.

```
mi_diccionario['clave1'] = 'nuevo_valor'
mi_diccionario['nueva_clave'] = 'valor_nuevo'
```

Eliminación de Elementos: pueden eliminar un par clave-valor utilizando la palabra clave del.

```
del mi_diccionario['clave1']
```

Diccionario

Verificación de Existencia de Claves: pueden verificar si una clave existe en un diccionario utilizando el operador in.

```
existe_clave = 'clave1' in mi_diccionario
```

```
# Resultado: False (ya que la eliminamos)
```

Longitud del Diccionario: pueden obtener la cantidad de elementos en un diccionario con la función len().

```
cantidad_elementos = len(mi_diccionario)
```

```
# Resultado: 2 (luego de eliminar una clave)
```

Diccionario

Longitud del Diccionario: pueden obtener la cantidad de elementos en un diccionario con la función len().

```
cantidad_elementos = len(mi_diccionario)
```

```
# Resultado: 2 (luego de eliminar una clave)
```

Iteración a través de Claves, Valores o Elementos: pueden iterar a través de las claves, valores o pares clave-valor en un diccionario.

```
for clave in mi_diccionario:
    valor = mi_diccionario[clave]
```

Diccionario

Diccionarios Anidados: pueden tener diccionarios dentro de diccionarios, creando así estructuras de datos más complejas.

```
diccionario_anidado = {'clave1': {'subclave1': 'valor_subclave1'},
                       , 'clave2': {'subclave2': 'valor_subclave2'}}
```

Los diccionarios son extremadamente útiles cuando necesitan almacenar datos estructurados y asociar información de manera eficiente utilizando claves únicas.

Ejercicios

Ejercicios relacionados con contenedores en Python, abarcando listas, tuplas y diccionarios. Estos ejercicios son útiles para practicar el manejo de estos tipos de datos y mejorar la comprensión de su funcionamiento:

Ejercicios sobre listas

Ejercicios sobre Tuplas

Ejercicios sobre diccionarios

Estos ejercicios son flexibles y pueden ser adaptados según el nivel de los estudiantes y los conceptos específicos que deseen enfatizar.

Creación de Tuplas:

- Cree una tupla llamada mi_tupla con tres elementos de tu elección.
- Intente modificar un elemento de mi_tupla. ¿Qué sucede?

Desempaquetado de Tuplas:

- Cree una tupla de dos elementos llamada coordenadas con valores de latitud y longitud.
- Utilice el desempaquetado de tuplas para asignar estos valores a dos variables distintas.

Operaciones Básicas con Diccionarios:

- Cree un diccionario llamado mi_diccionario con al menos tres pares clave-valor.
- Agregue una nueva clave-valor a mi_diccionario.
- Imprima sólo las claves del diccionario.

Diccionarios Anidados:

- Cree un diccionario anidado llamado contactos con información de al menos dos personas (nombre, edad, etc.).
- Acceda e imprima información específica de una persona utilizando el diccionario anidado.

Iteración y Actualización:

- Utilice un bucle for para imprimir todas las claves y valores de mi_diccionario.
- Actualice el valor de una clave en mi_diccionario.

Operaciones Básicas:

- Cree una lista vacía llamada mi_lista.
- Agregue los números del 1 al 5 a mi_lista.
- Imprima mi_lista.
- Elimine el número 3 de la lista.

Listas y Ciclos:

- Utilice un bucle for para imprimir cada elemento de mi_lista creado en el ejercicio 1.

Rebanado de Listas:

- Cree una lista llamada números del 1 al 10.
- Imprima los primeros tres elementos de números.
- Imprima los elementos desde el tercero hasta el sexto.
- Imprima los últimos dos elementos.

COLOMBIA POTENCIA DE LA VIDA

TIC

TALENTO TECH

AZ | PROYECTOS EDUCATIVOS

UTP Universidad Tecnológica de Pereira