



Fundamentos de programación

Python

Para el análisis de datos

Sesión 5

¿Qué es una Estructura de datos?

Son formas de organizar y almacenar información en un programa, facilitando el acceso y manipulación de los datos, optimizando el rendimiento y haciendo el código más claro y manejable.



Listas en Python

Una lista almacena una secuencia ordenada de elementos, permite agregar, eliminar y acceder a elementos por su índice. Es una estructura mutable.

```
mi_lista = [1, 2, 3, 4] # Creación de una lista

mi_lista.append(5) # Uso de método append() para agregar un elemento al final de la lista

elemento = mi_lista[0] # Acceder al elemento por el índice, para este ejemplo al Primer elemento

sublista = mi_lista[1:3] #La anotación [inicio:fin] permite una sublista de un rango de elementos

cadena = "Hola Mundo" # cadena se almacena en lista_cadena y se divide usando el método split()
lista_cadena = cadena.split() # ['Hola', 'Mundo']

matriz = [[1, 2], [3, 4]] #Lista puede contener otras listas, representando matrices

matriz = [['1', '2'], ['3', '4']] #lista puede contener otras listas, representando matrices
```

Diccionarios en Python

Almacenan pares clave-valor, permitiendo un acceso rápido mediante claves únicas. Son útiles para representar datos tabulares o relaciones entre elementos.

```
mi_diccionario = {'nombre': 'Ana', 'edad': 25} # Creación Diccionario

mi_diccionario['ciudad'] = 'Santiago' # Agregar Elementos

edad = mi_diccionario['edad'] # Rescate de Elementos

usuarios = {                                # Diccionarios Anidados
    'usuario1': {'nombre': 'Ana', 'edad': 25},
    'usuario2': {'nombre': 'Luis', 'edad': 30}
}
```

```
}
```

```
    'usuario2': {'nombre': 'Luis', 'edad': 30}
```

Tuplas en Python

Almacenan una secuencia ordenada de elementos, como una lista, pero no permiten modificaciones. Son útiles para datos constantes.

```
mi_tupla = (1, 2, 3) # Creación Tupla  
primer_elemento = mi_tupla[0] #Rescate Elementos
```

Empaqueta y Desempaqueta de Tuplas

El empaquetado agrupa varios valores en una tupla, y el desempaquetado asigna cada valor de la tupla a una variable individual.

```
tupla = 1, 2, 3
```

```
a, b, c = tupla
```


Sets en Python

Una colección desordenada de elementos únicos. Son eficientes para operaciones de agrupamiento o comparación, sin duplicados.

```
mi_set = {1, 2, 3} # Creación de un set
```

```
set_a = {1, 2, 3}  
set_b = {3, 4, 5}
```

```
# Unión
```

```
union = set_a | set_b # {1, 2, 3, 4, 5}
```

```
# Intersección
```

```
interseccion = set_a & set_b # {3}
```

```
# Diferencia
```

```
diferencia = set_a - set_b # {1, 2}
```

```
difference = set_a - set_b # {1, 2}
```

```
# Diferencia
```

Compresión de Listas, Diccionarios y Sets

Permite crear estructuras de datos en una sola línea, optimizando la legibilidad y rendimiento.

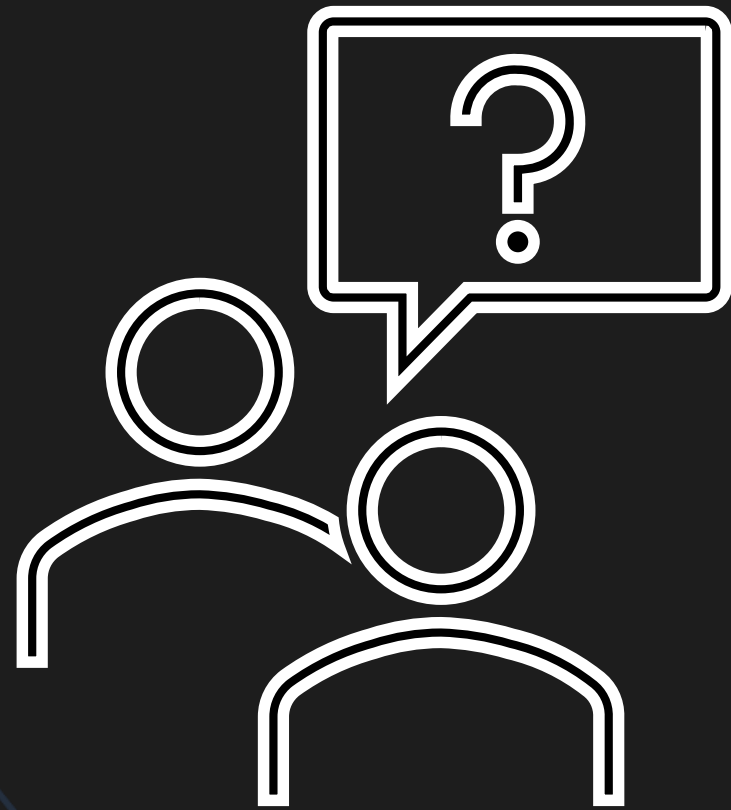
```
lista = [x * 2 for x in range(5)] # [0, 2, 4, 6, 8]
```

```
diccionario = {x: x**2 for x in range(5)} # {0: 0, 1: 1, 2: 4, 3: 9, 4: 16}
```

```
set_comprimido = {x * 2 for x in range(5)} # {0, 2, 4, 6, 8}
```


Preguntas

Sección de preguntas



A background network diagram with blue nodes and connecting lines, creating a web-like structure.

Fundamentos de programación

Python

Para el análisis de datos

Continúe con las
actividades