



**Universidad Autónoma de Baja  
California  
Facultad de Ingeniería Arquitectura y  
Diseño**



**Materia: Lenguaje de Programación Python**

**Alumno: FABIAN AGUIAR SERGIO**

**Carrera: Ingeniero en computación**

**Fecha: 10/03/2024**

**Matrícula: 374317**

**Maestro: Pedro Nuñez Yepiz**

**Grupo: 432**

**Práctica No. : #5**

**Tema - Unidad : LISTAS, RANGE, RANDOM**

## TEORÍA

### Listas:

Las listas en Python son una colección ordenada y mutable de elementos. Pueden contener cualquier tipo de dato, incluyendo otros objetos de Python, como números, cadenas, listas anidadas, etc. Se crean utilizando corchetes `[ ]` y los elementos se separan por comas. Por ejemplo:

```
-mi_lista = [1, 2, 3, 4, 5]
```

- Las listas en Python son una estructura de datos que almacena una colección ordenada y mutable de elementos.
- Se crean utilizando corchetes `[ ]` y los elementos se separan por comas.
- Pueden contener cualquier tipo de datos, como números, cadenas, booleanos, listas anidadas, etc.
- Los elementos de una lista se pueden acceder mediante índices, empezando desde 0 para el primer elemento y utilizando índices negativos para contar desde el final de la lista.
- Las listas son mutables, lo que significa que se pueden modificar agregando, eliminando o modificando elementos.
- Se pueden realizar diversas operaciones con listas, como concatenación, repetición, búsqueda, longitud, ordenamiento, inversión, entre otras.
- Se pueden iterar sobre los elementos de una lista utilizando bucles `for`.
- Algunas funciones útiles para trabajar con listas incluyen `len()` para obtener la longitud de la lista, `append()` para agregar elementos al final de la lista, `remove()` para eliminar un elemento específico, `pop()` para eliminar y devolver el último elemento, `insert()` para agregar un elemento en una posición específica, entre otras.

### Rango:

La función `range()` en Python se utiliza para generar una secuencia de números. Puede tomar uno, dos o tres argumentos, dependiendo de cómo se quiera utilizar. Si se pasa un solo argumento, genera una secuencia desde 0 hasta el número pasado como argumento (excluyendo ese número). Si se pasan dos argumentos, genera una secuencia desde el primer número hasta el segundo número (excluyendo el segundo número). Y si se pasan tres argumentos, el tercero especifica el paso entre los números. Por ejemplo:

```
-rango_1 = range(5)  
-rango_2 = range(1, 6)  
-rango_3 = range(0, 10, 2)
```

- `range()` es una función que genera una secuencia de números.
- Puede tomar uno, dos o tres argumentos.
- Si se pasa un solo argumento, genera una secuencia desde 0 hasta el número pasado como argumento (excluyendo ese número).
- Si se pasan dos argumentos, genera una secuencia desde el primer número hasta el segundo número (excluyendo el segundo número).

- Si se pasan tres argumentos, el tercero especifica el paso entre los números.
- La secuencia generada por `range()` es un objeto iterable, por lo que comúnmente se convierte en una lista utilizando la función `list()` si se desea ver los valores.
- Se utiliza principalmente en bucles `for` para iterar sobre una secuencia de números.
- Es una forma eficiente de generar secuencias numéricas sin ocupar mucha memoria, ya que genera los números sobre la marcha en lugar de almacenarlos todos en la memoria.
- Es importante recordar que `range()` genera números enteros y no flotantes.

#### Módulo Random:

El módulo `random` en Python proporciona funciones para generar números aleatorios. Algunas de las funciones comunes incluyen `random()`, que genera un número aleatorio entre 0 y 1, y `randint(a, b)`, que genera un número entero aleatorio en el rango entre `a` y `b` (ambos inclusive). Por ejemplo:

```
-import random
```

```
-num_aleatorio = random.random()
```

```
-num_entero = random.randint(1, 100)
```

- `random` es un módulo incorporado en Python que proporciona funciones para trabajar con números aleatorios.
- Se utiliza para generar números aleatorios y realizar operaciones relacionadas con la aleatoriedad.
- Algunas de las funciones más comunes en el módulo `random` incluyen:
  - `random()`: Devuelve un número aleatorio en punto flotante en el rango `[0.0, 1.0)`.
  - `randint(a, b)`: Devuelve un número entero aleatorio en el rango `[a, b]`, ambos inclusive.
  - `choice(seq)`: Devuelve un elemento aleatorio de la secuencia dada.
  - `shuffle(seq)`: Mezcla los elementos de la secuencia dada de forma aleatoria.
  - `sample(population, k)`: Devuelve una lista de `k` elementos únicos seleccionados aleatoriamente de la población dada.
- Antes de usar las funciones del módulo `random`, es necesario importarlo con `import random`.
- Se utiliza en una amplia variedad de aplicaciones, como juegos, simulaciones, generación de datos de prueba, entre otros.
- El módulo `random` utiliza un generador de números pseudoaleatorios para generar secuencias de números que parecen ser aleatorios pero en realidad son deterministas y reproducibles si se conoce la semilla (seed) utilizada.

Estas son solo algunas de las formas en que se pueden utilizar listas, rango y el módulo `random` en Python para realizar diversas tareas.

Git hub

[https://github.com/sergiofabian05/SFA\\_PY\\_ACT7.git](https://github.com/sergiofabian05/SFA_PY_ACT7.git)