

## range(10)

- `range(10)` produces a sequence:

```
python
```

[Copy](#) [Edit](#)

```
0, 1, 2, 3, 4, 5, 6, 7, 8, 9
```

- It's a **lazy object**, which is why `print(range(10))` shows:

```
scss
```

[Copy](#) [Edit](#)

```
range(0, 10)
```

- To view it as a list:

```
python
```

[Copy](#) [Edit](#)

```
list(range(10))
```

## ✅ List Comprehensions

### 1. Squares of numbers:

```
python
```

[Copy](#) [Edit](#)

```
squared_num = [x**2 for x in range(10)]
```

- This builds a new list where each number is squared.
- Output:

```
python
```

[Copy](#) [Edit](#)

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

### 2. Cubes of numbers (0 to 4):

```
python
```

[Copy](#) [Edit](#)

```
cube_num = [y**3 for y in range(5)]
```

- Output:

```
python
```

[Copy](#) [Edit](#)

```
[0, 1, 8, 27, 64]
```

## 💡 Extra Tips You Might Try Next

### ✅ Filter even numbers:

```
python
```

[Copy](#) [Edit](#)

```
even_nums = [x for x in range(10) if x % 2 == 0] # [0, 2, 4, 6, 8]
```

### ✅ Combine with functions:

```
python
```

[Copy](#) [Edit](#)

```
def square(x): return x * x squares = [square(i) for i in range(5)]
```

### ✅ Nested comprehensions (for 2D lists):

```
python
```

[Copy](#) [Edit](#)

```
matrix = [[i * j for j in range(3)] for i in range(3)] # [[0, 0, 0], [0, 1, 2], [0, 2, 4]]
```