

Range function

```
my_list = range(10)
print(list(my_list))
# [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

my_list2 = range(1,11)
print(list(my_list2))
# [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

my_list3 = range(0,26,5)
print(list(my_list3))
# [0, 5, 10, 15, 20, 25]
```

- To quickly create a list of consecutive numbers, you can use the **range()** function
- It takes a variable number of arguments
- **range(10)** generates a list from 0 to 9 (up to, not including 10)
- **range(1,11)** goes from 1 to 10
- **range(0,26,5)** goes from 0 to 25, at steps of 5
- **range(start, finish+1, step)**

Dictionaries

- Dictionaries (called “hash maps”, “lookup tables” or sometimes “objects” in other languages) are useful when you need to quickly look up a value, and order isn’t important
- They take the form of key-value pairs **{“key”: “value”}**
- Get a key’s value using square brackets: **dict[“key”]** or **dict.get(“key”)**
- You can also update it by passing in a new dictionary:
dict.update({“key”: “new-value”})

```
dict = {"one": 1, "two": 2, "three": 3}
print(dict["one"]) # => 1
print(dict.get("two")) # => 2
dict.update({"three": 33})
print(dict["three"]) # => 33
```