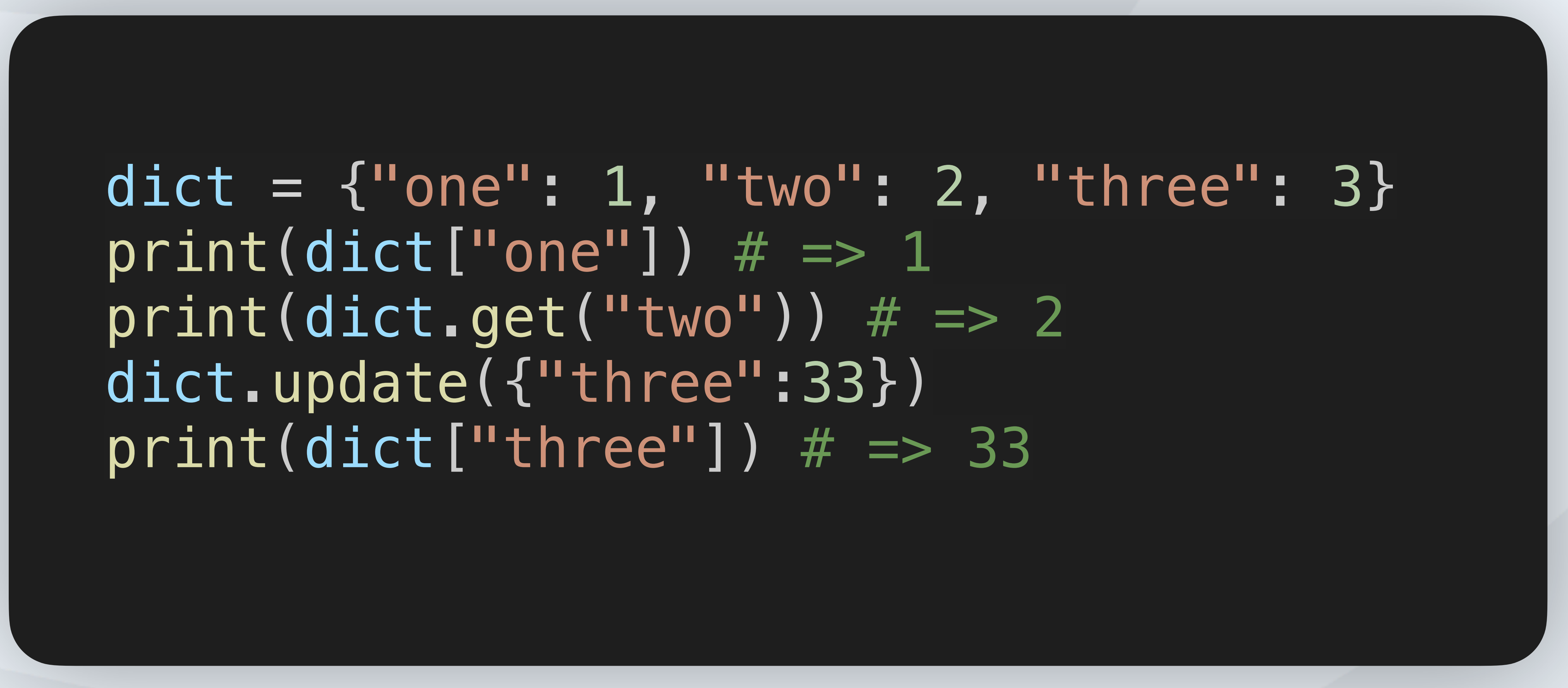
Dictionaries

- 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
```

Loops

[1]

```python
primes = [2, 3, 5, 7]
for prime in primes:
    print(prime)

print()

for n in range(len(primes)):
    print(primes[n])
```

=>

```
2
3
5
7

2
3
5
7
```

[2]

```python
count = 0
while count < 5:
    print(count)
    count += 1  # count = count + 1
```

=>

```
0
1
2
3
4
```

- Use loops to go through lists one by one
- The two main types of loop in Python are **for** [1] and **while** [2] loops
- **for item in list:** is a shorthand that assigns each item to a variable (**item** in this case)
- otherwise, you can use **range(len(list))** to go through each index
- **while** loops keep on going until a condition is met, then stop when it becomes **True**