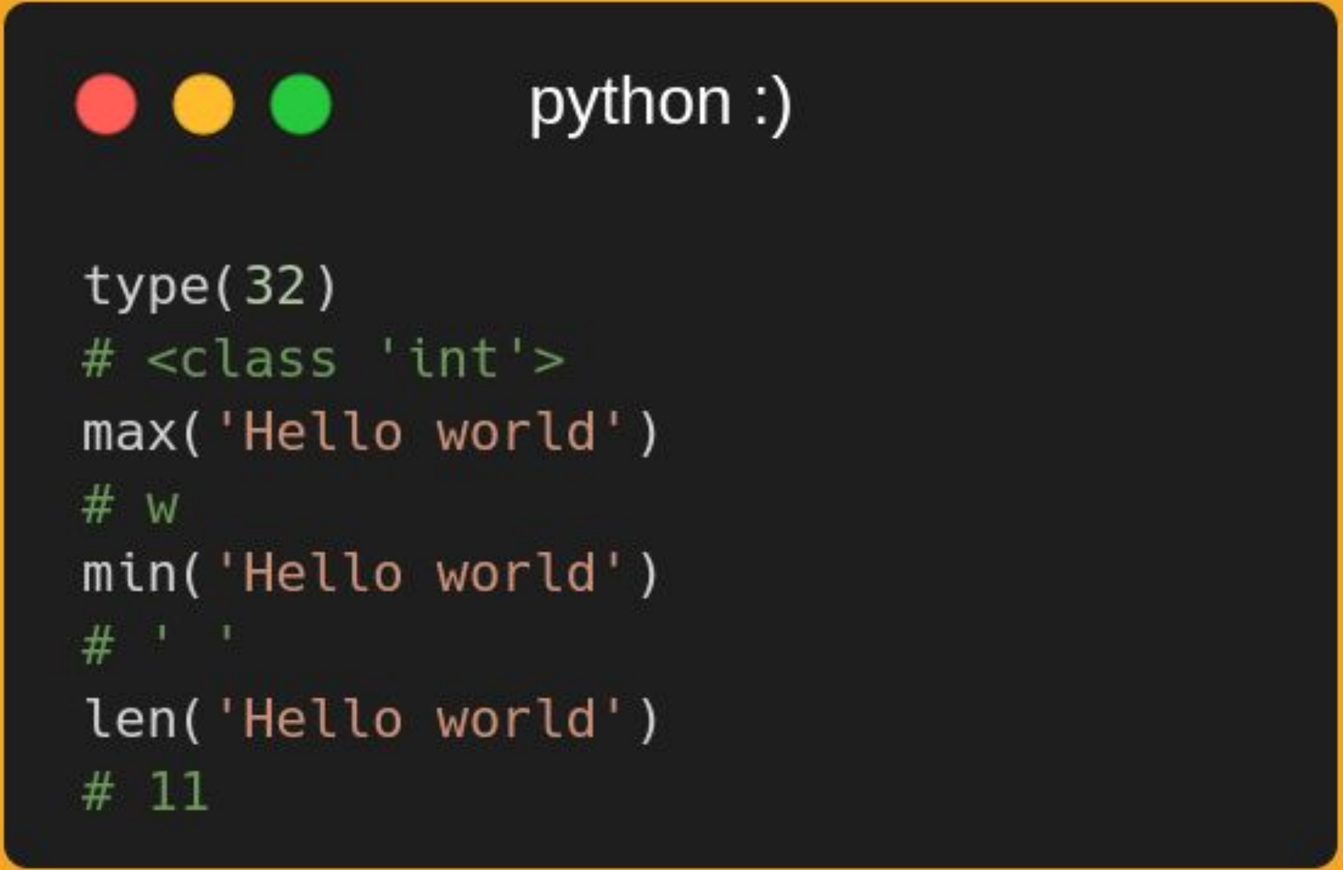
Python Course

Functions and Loops

Feb 2023
Matin KarimPour

Functions

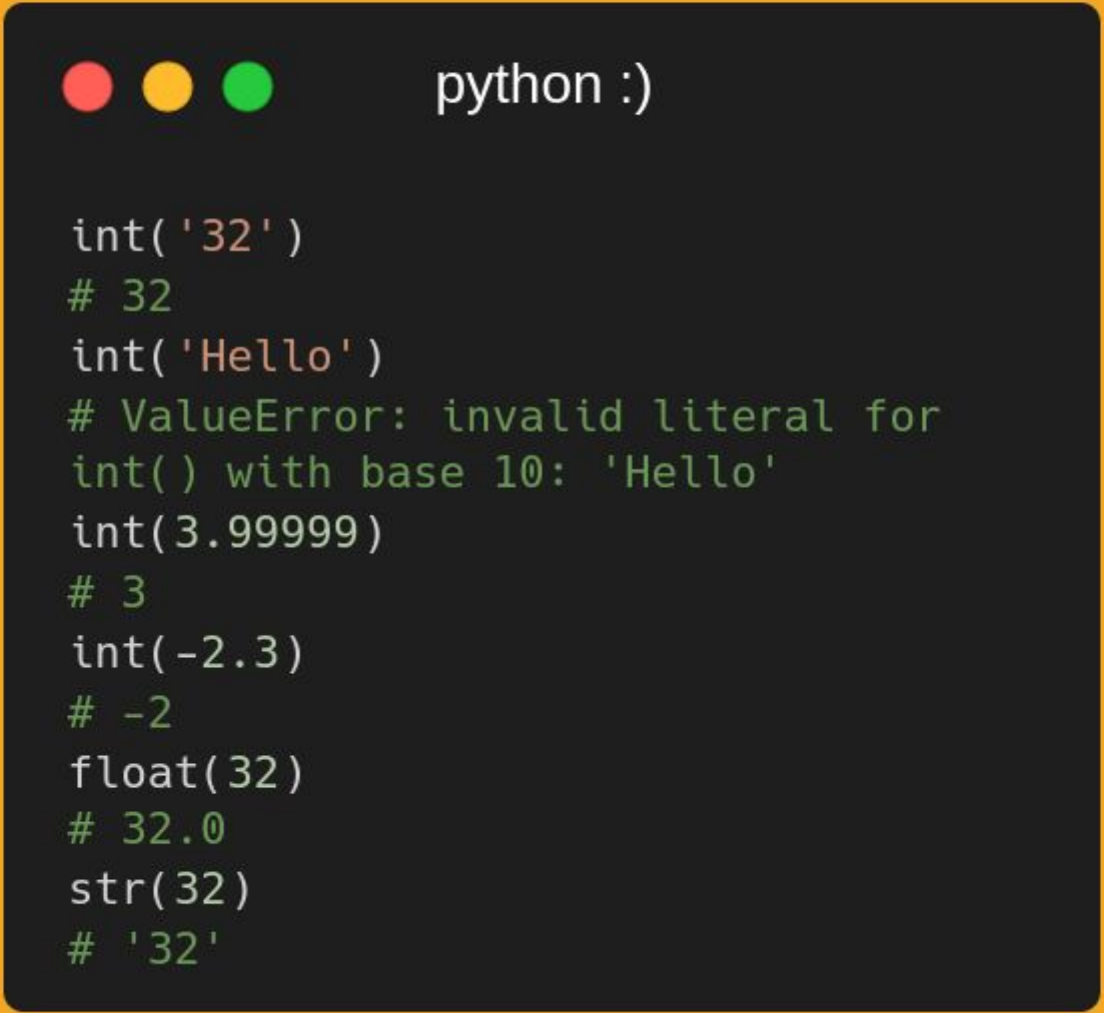
In the context of programming, a function is a named sequence of statements that performs a computation. When you define a function, you specify the name and the sequence of statements. Later, you can “call” the function by name.



```
type(32)
# <class 'int'>
max('Hello world')
# w
min('Hello world')
# ' '
len('Hello world')
# 11
```

Type conversion functions

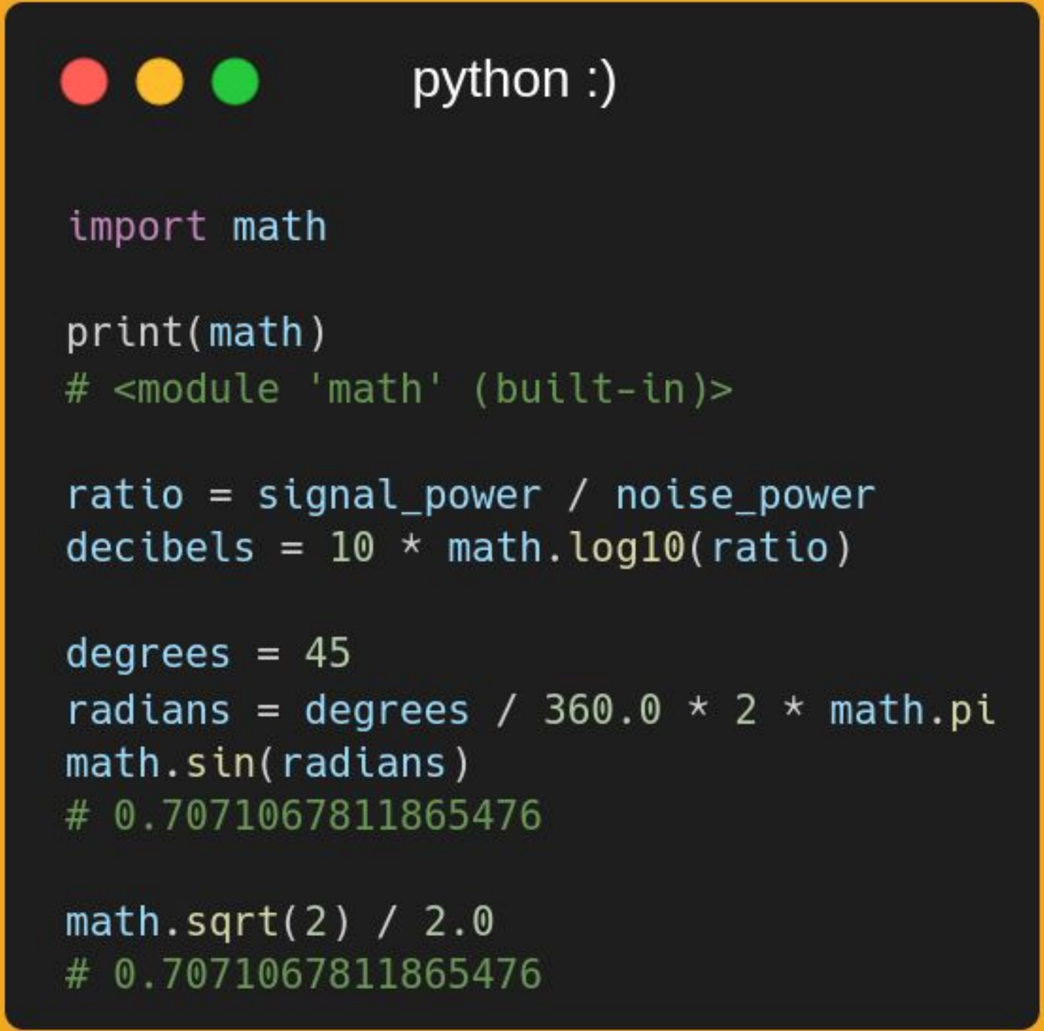
Python also provides built-in functions that convert values from one type to another.



```
int('32')
# 32
int('Hello')
# ValueError: invalid literal for
int() with base 10: 'Hello'
int(3.99999)
# 3
int(-2.3)
# -2
float(32)
# 32.0
str(32)
# '32'
```

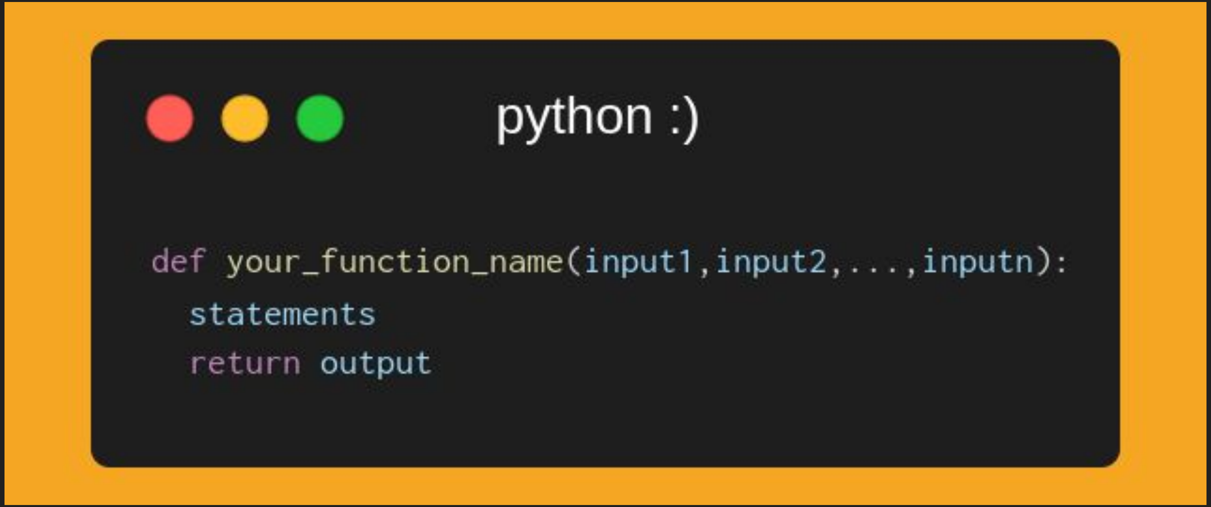
Math functions

Python has a math module that provides most of the familiar mathematical functions.



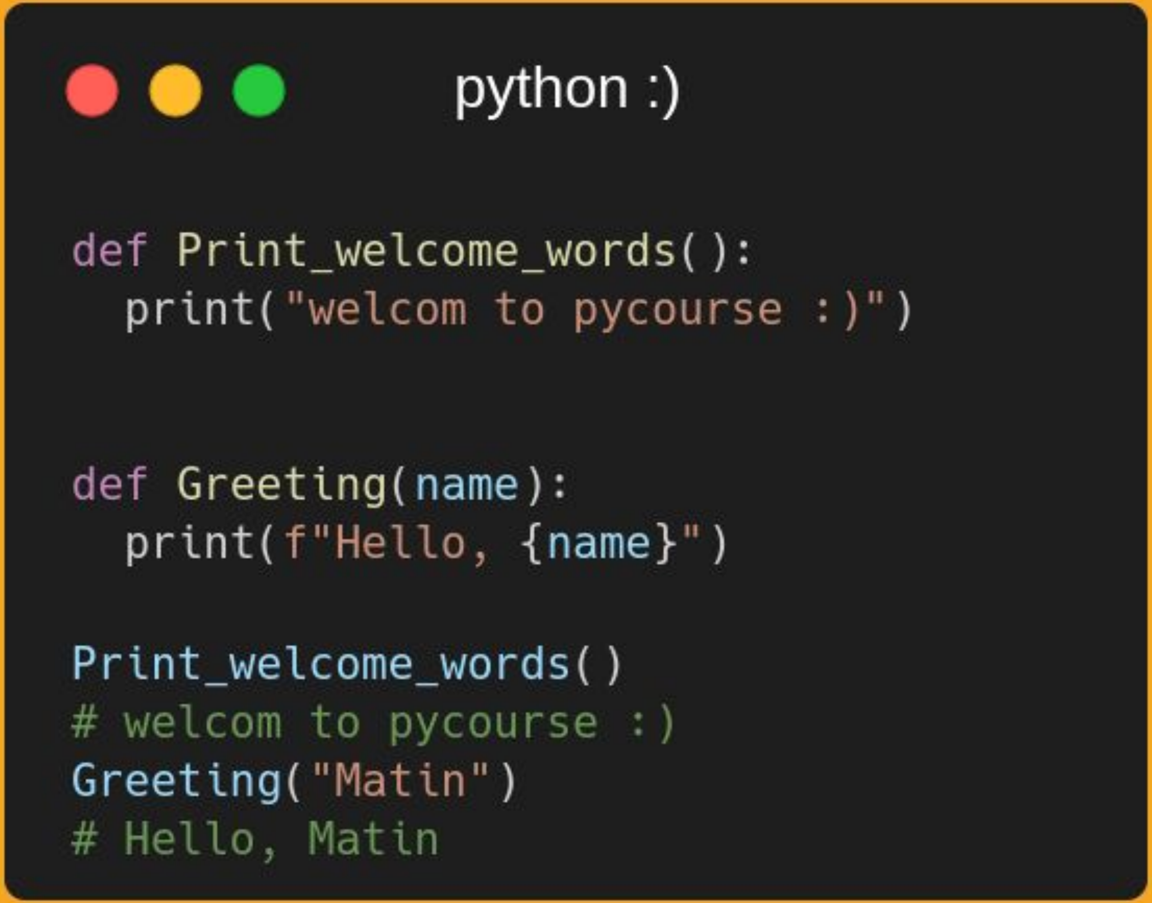
```
import math

print(math)
# <module 'math' (built-in)>

ratio = signal_power / noise_power
decibels = 10 * math.log10(ratio)

degrees = 45
radians = degrees / 360.0 * 2 * math.pi
math.sin(radians)
# 0.7071067811865476

math.sqrt(2) / 2.0
# 0.7071067811865476
```

Adding new functions



```
def your_function_name(input1,input2,...,inputn):
  statements
  return output
```

Function examples



```python
def Print_welcome_words():
  print("welcom to pycourse :)")


def Greeting(name):
  print(f"Hello, {name}")

Print_welcome_words()
# welcom to pycourse :)
Greeting("Matin")
# Hello, Matin
```

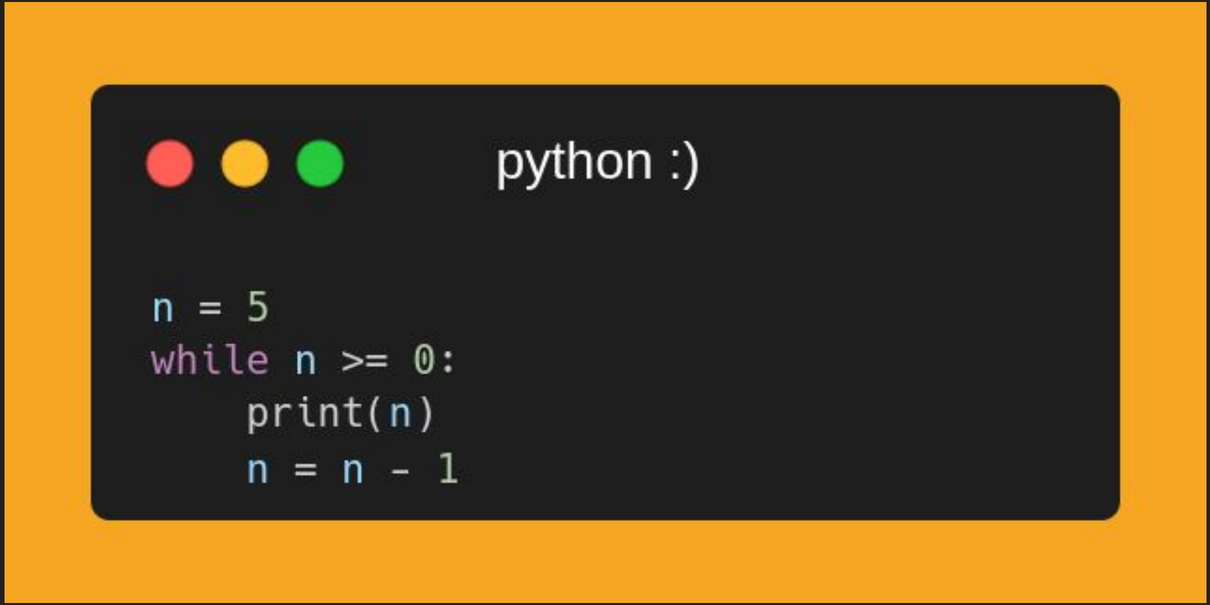
Why functions?

- makes your program easier to read, understand, and debug
- Later, if you make a change, you only have to make it in one place.
- Once you write and debug one, you can reuse it.
- debug the parts one at a time and then assemble them into a working whole.

Loops

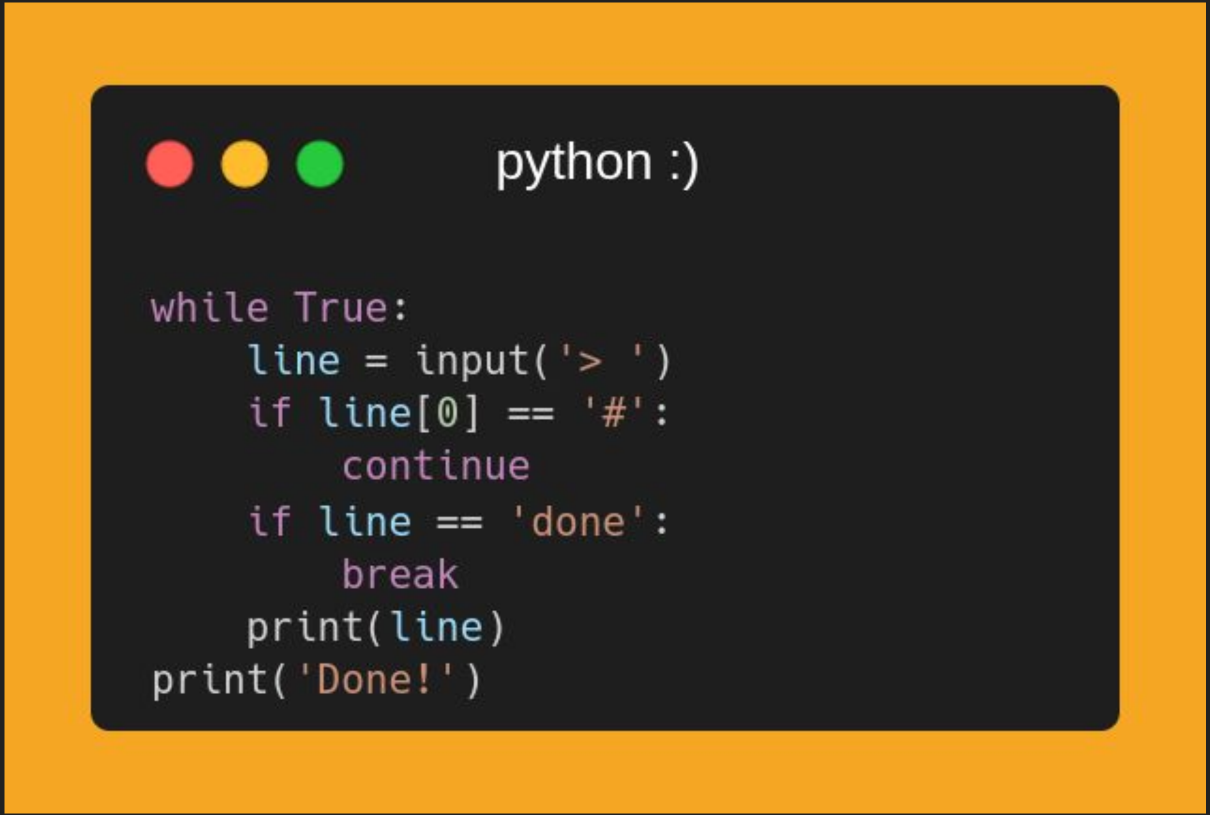
The while statement

Computers are often used to automate repetitive tasks. Repeating identical or similar tasks without making errors is something that computers do well and people do poorly. Because iteration is so common, Python provides several language features to make it easier.



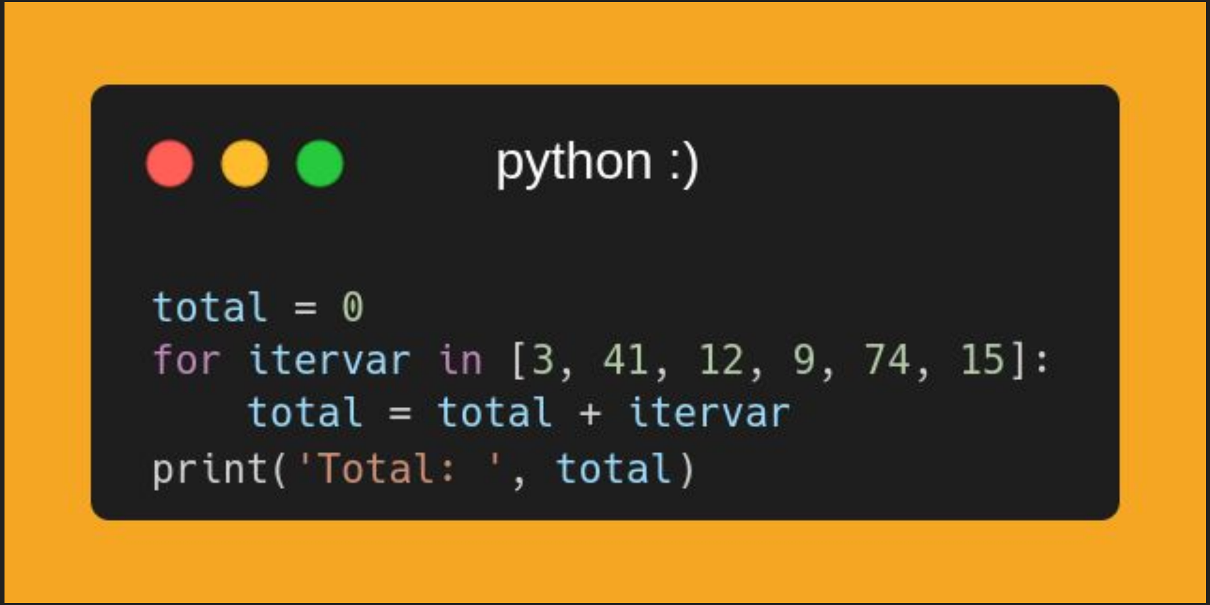
```python
n = 5
while n >= 0:
    print(n)
    n = n - 1
```

Infinite loops



```python
while True:
    line = input('> ')
    if line[0] == '#':
        continue
    if line == 'done':
        break
    print(line)
print('Done!')
```

Loops using for



```python
total = 0
for itervar in [3, 41, 12, 9, 74, 15]:
    total = total + itervar
print('Total: ', total)
```

Any Questions?