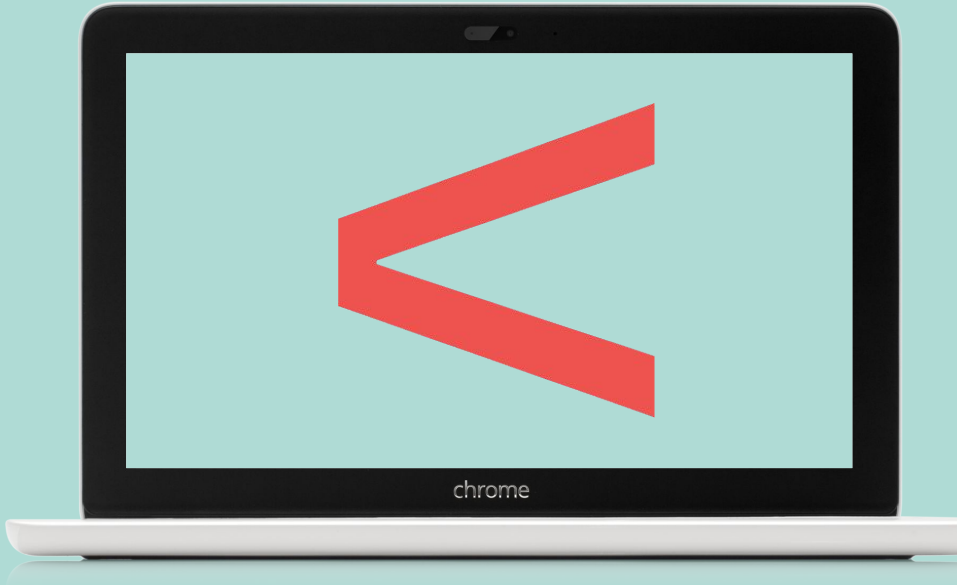
Introduction to Python



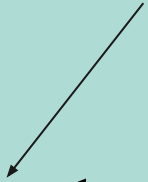
What is Python?

Python is an interpreted,
interactive,
object-oriented
programming language.

Source: https://docs.python.org/3/faq/general.html#what-is-python

What is Python?

Written instructions are interpreted and then executed by the Python Virtual Machine. They are read line by line.



Python is an interpreted, interactive, object-oriented programming language.

Source: https://docs.python.org/3/faq/general.html#what-is-python

What is Python?

We can interact with the python shell and get immediate feedbacks.

Python is an interpreted, interactive, object-oriented programming language.

Source: https://docs.python.org/3/faq/general.html#what-is-python

What is Python?

Python is an interpreted, interactive, object-oriented programming language.

Method of structuring a program by building related properties and behaviors into individual objects. In data science we use mainly *procedural programming*.

Source: https://docs.python.org/3/faq/general.html#what-is-python

Basic types in Python

Numeric Types:

- float: 2.3
- int: 2
- Complex: i + 2

Text Type:

- str: “hello world”

Boolean Type:

- bool: True, False

Each type will behave differently to any operation, for example:

As an integer:

a = 2
b = a*2
b = 4

As a string

a = ‘2’
b = ‘22’

Basic types in Python

Sequence Types:

- List: [2.3, “hello”]
- Tuple: (2.3, “hello”)

* List ≠ tuple: The key difference between tuples and lists is that while the tuples are immutable objects the lists are mutable.

Basic types in Python

- set: {“Peter”, “John”, “Mary”, “Jane” }

A set is an unordered and mutable collection of **unique** elements. Like dictionaries, sets have no index, and can't be considered sequences.

Basic types in Python

Mapping Type:

- dict: {“Name”: “John”}

A dictionary is an unordered (with no index) collection of data values, held as key:value pairs. Dictionaries used to be unordered, but you can now have OrderedDict.

There are more types. Refer to the documentation in python.org

Conditions in Python - comparison operators

As their name imply, comparison operators compare values and, based on a condition, produce a boolean.

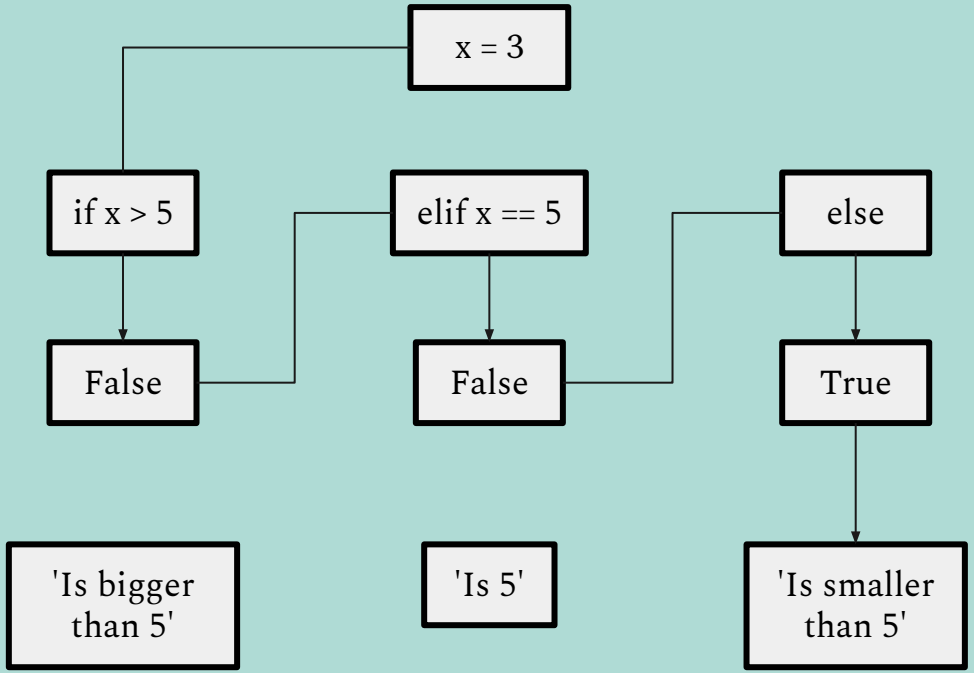
```
a = 5
```

a == 5: True	a > 5: False	a >= 5: True
a != 5: False	a < 5: False	a <= 5: True
a == 6: False	a > 6: False	a >= 6: False
a != 6: True	a < 6: True	a <= 6: True

Conditions in Python - if statement

Allows to run different statements for different inputs.

```
if x > 5:
        print('Is bigger than 5')
elif x == 5:
        print('Is 5')
else:
        print('Is smaller than 5')
```



What is Python?

Always read the documentation!