

# Cyrill Stechniss

## Python Core Introduction

Date	
Page	

OM  
Student Notebooks

\* `print("Hello World!")`

↳ Print out Hello World! to the screen.

• `name = "Aditya"`

`print("Name: " + name)`

↳ String Concatenation

↳ Variable of type string

⇒ Different type of data you can store in a variable:

→ String (`name = "Aditya"`)

→ Number

↳ int (`age = 25`)

↳ float (`score = 82.4`)

→ Bool (`is_male = True`  
`is_female = False`)

## \* Working with strings

\* \ is escape character

`"\" => "`

\* String Concatenation

`"Aditya" + "Shrivastava" => "AdityaShrivastava"`

\* `phrase = "Aditya"`

`phrase.lower()`  $\Rightarrow$  "aditya"

\* `phrase.upper()`  $\Rightarrow$  "ADITYA"

\* `phrase.isupper()`

$\rightarrow$  Return True if phrase is entirely upper case.

\* `phrase.islower()`

$\rightarrow$  Return True if phrase is entirely lower case.

\* `len(phrase)`  $\Rightarrow$  6

\* `phrase[0]`  $\Rightarrow$  A

\* `phrase.index("t")`  $\Rightarrow$  3

$\rightarrow$  location of first "t" in the phrase.

\* `phrase.index("tya")`  $\Rightarrow$  3

$\rightarrow$  location of first "tya" in the phrase.

\* `phrase = "hello world"`

`phrase.replace("o", "x")`

$\rightarrow$  "hellx wxld"

$\rightarrow$  Same works for more than one character.



## \* Working with Numbers

my\_num = 5

\* **Str(my\_num)**

↳ Converts numbers to string

\* **abs(-5.2)**

↳ 5.2 {absolute value of the number}

\* **Pow(5, 2)**

↳ 25

\* **max(4, 6)**

↳ 6

\* **min(4, 6)**

↳ 4

\* **round(3.3)**

↳ 3

**round(3.7)**

↳ 4

from math import \*

**Floor(3.7)**

↳ 3

**Ceil(3.7)**

↳ 4

**rint(16)**

↳ 4

## \* Getting input from Users

~~name~~ name = input("Enter your name: ")

int("5")

↳ 5 {converts string to integer}

float("5.2")

↳ 5.2 {converts string to float}

## \* Lists

friends = ["Kevin", "Karen", "Jim"]

friends[1]

↳ "Karen"

friends[-1]

↳ "Jim"

friends[1:]

↳ ["Karen", "Jim"]

char = ["A", "B", "C", "D", "E"]

char[1:3]

↳ ["B", "C"]



## ★ List functions

Char ~~list~~ = ["A", "B", "C"]  
Int = [1, 2, 3]

Char.extend(int)

↳ ["A", "B", "C", 1, 2, 3]

Char.append("D")

↳ ["A", "B", "C", "D"]

Char.insert(1, "a")

↳ ["A", "a", "B", "C"]

Char.remove("A")

↳ ["B", "C"]

Char.clear()

↳ []

Char.pop()

↳ ["A", "B"]

Char.index("C")

↳ 2

char\_count("A")

↳ 1 [Number of "A" in the list]

g\_num = [5, 7, 2, 1]

g\_num.sort()

↳ [1, 2, 5, 7]

g\_num.reverse()

↳ [1, 2, 7, 5]

g\_num = int.copy()

↳ [1, 2, 3]

## ★ Tuples

↳ It is immutable.

Coordinate = (4, 5)

Coordinate[0]

↳ 4



## ★ Functions

↳ Collection of code that performs a specific task.

```
def SayHi():  
    print("Hi")
```

```
def SayHi(name):  
    print("Hello" + name)
```

```
SayHi("Mike")
```

→ Hello Mike (Print on Screen)

```
def cube(num):  
    return num * num * num
```

## ★ If Statement

```
* if condition1:  
    <code>  
elif condition2:  
    <code>  
else:  
    <code>
```

→ Condition can be joined by using "and" & "or" not

<, >, =, >=, <=, ==, !=

{Comparison operators}



## ★ Dictionaries

```
month_conversion = {
    "Jan": "January",
    "Feb": "February",
    : : : : :
}
```

→ Keys should be unique.

```
month_conversions["Jan"]
```

→ "January"

```
month_conversion.get("Dec")
```

→ "December"

```
month_conversion.get("Lov", "Not Valid")
```

→ "Not Valid"

## ★ While loop

While Condition:

<code>

## ★ For loop

for letter in "Gisette Academy":

print(letter)

0 1 2 3 4 5 6 7 8 9  
3 4 5 6 7 8 9

range(10)

range(3, 10)

→ string

→ list

→ Number



## ★ Comments

⇒ Add "#" in the front of the line to comment it.

## ★ Try Except

try:

<code>

except:

<code>

If it throws any exception

⇒ You can be specific with except:

→ except ZeroDivisionError:

→ except ValueError:

⇒ You can also store the exception in a variable:

except ZeroDivisionError as err:

print(err)

## ★ Read files

\* open("Path to file", "r")

→ To open a file

→ In read only mode

"w"

→ Writing to file

"a"

→ append to file

"rt"

→ read + write to file



\* `file.close()`

↳ To Close the file

\* `file.readable()`

↳ returns true if file is readable.

\* `file.read()`

↳ read all the information in the file

\* `file.readline()`

↳ Will read first line when called first time  
will read second line when called second time  
and so on.

\* `file.readlines()`

↳ Will store all the lines in the file in a list.

## \* Writing to Files.

⇒ Open in append mode

\* `file.write("text")`

↳ text will be added to  
the end of the file.

⇒ If open in write mode  
"text" will be the only  
text in the file

↳ It will be  
overwritten



## \* Modules and pip

⇒ Module is just a python file that you can import into your current python file.

① File: `useful-tools.py`

② file: `app.py`

\* `import useful-tools`

\* `from useful-tools import <something>`

⇒ `pip install <package-name>`

⇒ `pip uninstall <package-name>`

## \* Classes & Object

Class Student:

`def __init__(self, name, major, gpa):`

`self.name = name`

`self.major = major`

`self.gpa = gpa`

{Attributes}

## \* Object functions

Cont.

`def on_honor_roll(self):`

`if self.gpa >= 3.5:`

`return True`

`else:`

`return False`



## \* Inheritance

Class BioStudent (Student):

↳ BioStudent inherits from student.

~~\_\_\_\_\_~~