

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
o

4. Introduction
oooooooooooo

5. References
o

KOLT Python

Introduction

Ahmet Uysal

Monday 23rd September, 2019

KOLT

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
o

4. Introduction
oooooooooooo

5. References
o

Agenda

1. Program Information

2. Logistics

3. Installations

4. Introduction

5. References

1. Program Information
●○○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

Course Outcomes

- Apply basic programming concepts using Python

1. Program Information
●○○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

Course Outcomes

- Apply basic programming concepts using Python
- Demonstrate how Python can be used in different areas or disciplines

1. Program Information
●○○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

Course Outcomes

- Apply basic programming concepts using Python
- Demonstrate how Python can be used in different areas or disciplines
- Create code that is easy to understand

1. Program Information
●○○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

Course Outcomes

- Apply basic programming concepts using Python
- Demonstrate how Python can be used in different areas or disciplines
- Create code that is easy to understand
- **Implement practical challenges** by gaining experience in Python

1. Program Information
●○○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

Why Python?

1. Program Information
●○○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

Why Python?

- Easy Syntax

1. Program Information
●●○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

Why Python?

- Easy Syntax
- Beginner Friendly -most popular language for introductory CS courses in top universities[1]-

1. Program Information
●●○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

Why Python?

- Easy Syntax
- Beginner Friendly -most popular language for introductory CS courses in top universities[1]-
- Wide usage area

1. Program Information
●●○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○○○○○○○○○

5. References
○

Why Python?

- Easy Syntax
- Beginner Friendly -most popular language for introductory CS courses in top universities[1]-
- Wide usage area
- Large and growing community

1. Program Information
○○●○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○○○○○○○○○

5. References
○

Some of the Usage Areas [2]

- Data Analysis

1. Program Information
○○●○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○○○○○○○○○

5. References
○

Some of the Usage Areas [2]

- Data Analysis
- Web Development

1. Program Information
○○●○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○○○○○○○○○

5. References
○

Some of the Usage Areas [2]

- Data Analysis
- Web Development
- System Administration

1. Program Information
○○●○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○○○○○○○○○

5. References
○

Some of the Usage Areas [2]

- Data Analysis
- Web Development
- System Administration
- Machine Learning

1. Program Information
○○●○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○○○○○○○○○

5. References
○

Some of the Usage Areas [2]

- Data Analysis
- Web Development
- System Administration
- Machine Learning
- Web Parsers/Scrawlers

1. Program Information
○○●○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○○○○○○○○○

5. References
○

Some of the Usage Areas [2]

- Data Analysis
- Web Development
- System Administration
- Machine Learning
- Web Parsers/Scrawlers
- Testing

1. Program Information
○○●○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○○○○○○○○○

5. References
○

Some of the Usage Areas [2]

- Data Analysis
- Web Development
- System Administration
- Machine Learning
- Web Parsers/Scrawlers
- Testing
- Education

1. Program Information
○○●○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○○○○○○○○○

5. References
○

Some of the Usage Areas [2]

- Data Analysis
- Web Development
- System Administration
- Machine Learning
- Web Parsers/Scrawlers
- Testing
- Education
- Network Programming

1. Program Information
○○●○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○○○○○○○○○

5. References
○

Some of the Usage Areas [2]

- Data Analysis
- Web Development
- System Administration
- Machine Learning
- Web Parsers/Scrawlers
- Testing
- Education
- Network Programming
- ...

1. Program Information
○○○●○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

Python at Koç University

1. Program Information
○○○●○○○

2. Logistics
○○○○

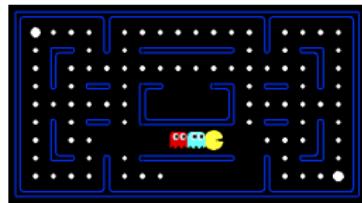
3. Installations
○

4. Introduction
○○○○○○○○○○○○○○

5. References
○

Python at Koç University

- COMP341: Introduction to Artificial Intelligence



1. Program Information
○○○●○○○

2. Logistics
○○○○

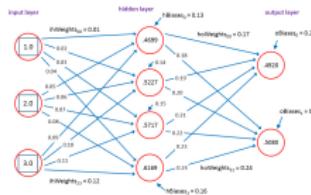
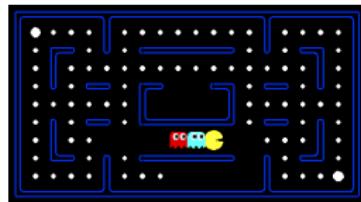
3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

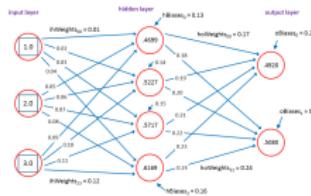
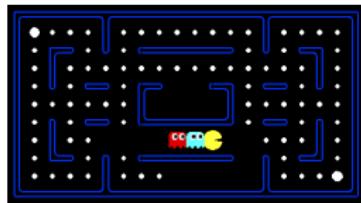
Python at Koç University

- COMP341: Introduction to Artificial Intelligence
- COMP421/521: Introduction to Machine Learning



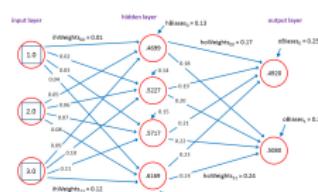
Python at Koç University

- COMP341: Introduction to Artificial Intelligence
- COMP421/521: Introduction to Machine Learning
- ENGR350 (Selected Topics - Summer18/Spring19): Introduction to Programming for Data Science



Python at Koç University

- COMP341: Introduction to Artificial Intelligence
- COMP421/521: Introduction to Machine Learning
- ENGR350 (Selected Topics - Summer18/Spring19): Introduction to Programming for Data Science
- INTL450 (Selected Topics - Spring19): Advanced Data Analysis in Python



1. Program Information
oooo●●○

2. Logistics
○○○○

3. Installations
○

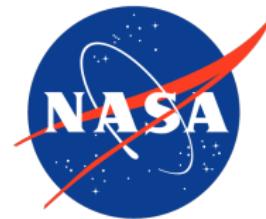
4. Introduction
○○○○○○○○○○○○

5. References
○

Python at Industry



Google



YouTube



1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
o

4. Introduction
oooooooooooo

5. References
o

Python at Industry [3]

Web Service Efficiency at Instagram with Python



Instagram Engineering [Follow](#)

Jun 21, 2016 · 6 min read

Instagram currently features the world's largest deployment of the Django web framework, which is written entirely in Python. We initially chose to use Python because of its reputation for simplicity and practicality, which aligns well with our philosophy of "do the simple thing first." But simplicity can

PyTransit

<https://github.com/mrtommmyb/ktransit>

Fast and easy-to-use tools for exoplanet transit light curve modelling with Python. PyTransit implements the quadratic Mandel & Agol and the Giménez transit models with various optimisations, and offers both a simple interface for model evaluation and a lower-level access for fine-tuning the model.

#Kepler

GNU General Public License (GPL) version 3

ktransit

<https://github.com/mrtommmyb/ktransit>

1. Program Information
oooooooo●

2. Logistics
oooo

3. Installations
○

4. Introduction
oooooooooooo

5. References
○

Python Everywhere



imgflip.com

1. Program Information
oooooooo

2. Logistics
●ooo

3. Installations
○

4. Introduction
oooooooooooo

5. References
○

Who Are We?



Ahmet Uysal
auysal16@ku.edu.tr

1. Program Information
oooooooo

2. Logistics
●ooo

3. Installations
○

4. Introduction
oooooooooooo

5. References
○

Who Are We?



Ahmet Uysal
auysal16@ku.edu.tr



Ceren Kocaoğullar
ckocaoğullar15@ku.edu.tr

1. Program Information
oooooooo

2. Logistics
●ooo

3. Installations
○

4. Introduction
oooooooooooo

5. References
○

Who Are We?



Ahmet Uysal
auysal16@ku.edu.tr



Gül Sena Altıntaş
galtintas17@ku.edu.tr



Ceren Kocaoğullar
ckocaogullar15@ku.edu.tr

1. Program Information
oooooooo

2. Logistics
●ooo

3. Installations
○

4. Introduction
oooooooooooo

5. References
○

Who Are We?



Ahmet Uysal
auysal16@ku.edu.tr



Gül Sena Altıntaş
galtintas17@ku.edu.tr



Ceren Kocaoğullar
ckocaoğullar15@ku.edu.tr



Hasan Can Aslan
haslan16@ku.edu.tr

1. Program Information
○○○○○○○

2. Logistics
○●○○

3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

What Will We Do?

Lecture Monday 11:30–12:45

1. Program Information
○○○○○○○

2. Logistics
○●○○

3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

What Will We Do?

Lecture Monday 11:30-12:45

Section Wednesday 14:30-15:45 **or**

Section Thursday 11:30-12:45 **or**

Section Thursday 13:00-14:15 **or**

Section Thursday 16:00-17:15

1. Program Information
○○○○○○○

2. Logistics
○○●○

3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

Programming Assignments

1. Program Information
○○○○○○○

2. Logistics
○○●○

3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

Programming Assignments (Don't Be Afraid!)

1. Program Information
○○○○○○○

2. Logistics
○○●○

3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

Programming Assignments (Don't Be Afraid!)

- 4-6 in total

1. Program Information
oooooooo

2. Logistics
oo●o

3. Installations
○

4. Introduction
oooooooooooo

5. References
○

Programming Assignments (Don't Be Afraid!)

- 4-6 in total
- Review sessions to **help you**

1. Program Information
○○○○○○○

2. Logistics
○○●○

3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

Programming Assignments (Don't Be Afraid!)

- 4-6 in total
- Review sessions to **help you**
- Some assignments will have **autograders** to help you find your mistakes and test your code.

1. Program Information
○○○○○○○

2. Logistics
○○●○

3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

Programming Assignments

1. Program Information
○○○○○○○

2. Logistics
○○●○

3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

Programming Assignments (Don't Be Afraid!)

- 4-6 in total
- Review sessions to **help you**
- Some assignments will have **autograders** to help you find your mistakes and test your code.
- Later assignments will be based on **your interests!**

1. Program Information
○○○○○○○

2. Logistics
○○○●

3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

Certificate Requirements

1. Program Information
○○○○○○○

2. Logistics
○○○●

3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

Certificate Requirements

- At most **3 unexcused absences**, including onsite contests and review sessions.

1. Program Information
○○○○○○○

2. Logistics
○○○●

3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

Certificate Requirements

- At most **3 unexcused absences**, including onsite contests and review sessions.
- Working on and submitting all homework assignments. Submissions that do not pass the autograders will be examined by us.

1. Program Information
oooooooo

2. Logistics
oooo●

3. Installations
○

4. Introduction
oooooooooooo

5. References
○

Certificate Requirements

- At most **3 unexcused absences**, including onsite contests and review sessions.
- Working on and submitting all homework assignments. Submissions that do not pass the autograders will be examined by us.
- We do not expect that you ace all programming assignments. But, we expect that you **spend time** on them!

1. Program Information
○○○○○○○

2. Logistics
○○○●

3. Installations
○

4. Introduction
○○○○○○○○○○○○

5. References
○

Certificate Requirements

- At most **3 unexcused absences**, including onsite contests and review sessions.
- Working on and submitting all homework assignments. Submissions that do not pass the autograders will be examined by us.
- We do not expect that you ace all programming assignments. But, we expect that you **spend time** on them!
- Complying to Koç University Code of Conduct.

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
●

4. Introduction
oooooooooooo

5. References
o

Installations

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
o

4. Introduction
●oooooooooooo

5. References
o

Interactive Interpreter

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○●○○○○○○○○○○

5. References
○

Why It Matters?

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○●○○○○○○○○○○

5. References
○

Why It Matters?

- Immediate gratification :)

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○●○○○○○○○○○○

5. References
○

Why It Matters?

- Immediate gratification :)
- Provides a sandboxed environment to experiment

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
o

4. Introduction
oooooooooooo

5. References
o

Why It Matters?

- Immediate gratification :)
- Provides a sandboxed environment to experiment
- You are not sure what something does, **try it!**

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
o

4. Introduction
oooooooooooo

5. References
o

Why It Matters?

- Immediate gratification :)
- Provides a sandboxed environment to experiment
- You are not sure what something does, **try it!**
- Shortens code-test-debug cycle and speeds up learning

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○●○○○○○○○○○

5. References
○

Comments

```
# Single line comments start with a '#'
```

```
"""
```

Multiline comments can be written between three "s and are often used as function and module comments.

```
"""
```

```
print('Hello, stranger!')
```

Python will basically ignore comments, they are purely written **for humans!**

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○●○○○○○○○○

5. References
○

Variables

- How to represent/store values in Python?

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○●○○○○○○○○

5. References
○

Variables

- How to represent/store values in Python?
- Which kind of values we need to represent?

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○●○○○○○○○○

5. References
○

Variables

- How to represent/store values in Python?
- Which kind of values we need to represent?
 - Numbers?

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○●○○○○○○○○

5. References
○

Variables

- How to represent/store values in Python?
- Which kind of values we need to represent?
 - Numbers?
 - Texts?

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
o

4. Introduction
oooo●oooooooo

5. References
o

Variables

- How to represent/store values in Python?
- Which kind of values we need to represent?
 - Numbers?
 - Texts?
 - Individual Characters?

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○●○○○○○○○○

5. References
○

Variables

- How to represent/store values in Python?
- Which kind of values we need to represent?
 - Numbers?
 - Texts?
 - Individual Characters?
 - Starting time of the class?

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
o

4. Introduction
oooo●oooooooo

5. References
o

Variables

- How to represent/store values in Python?
- Which kind of values we need to represent?
 - Numbers?
 - Texts?
 - Individual Characters?
 - Starting time of the class?
 - Colors?

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○●○○○○○○○○

5. References
○

Variables

- How to represent/store values in Python?
- Which kind of values we need to represent?
 - Numbers?
 - Texts?
 - Individual Characters?
 - Starting time of the class?
 - Colors?
 - Truth Values?

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○●○○○○○○○○

5. References
○

Variables

- How to represent/store values in Python?
- Which kind of values we need to represent?
 - Numbers?
 - Texts?
 - Individual Characters?
 - Starting time of the class?
 - Colors?
 - Truth Values?
 - People?

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
o

4. Introduction
oooo●oooooooo

5. References
o

Variables

Type	Explanation	Examples
int	represent integers	3, 4, 17, -10
float	represent real numbers	3.0, 1.11, -109.123123
bool	represent boolean truth values	True, False
str	A sequence of characters.	'Hello', ", '3'
NoneType	special and has one value, None	None

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
o

4. Introduction
oooo●oooooooo

5. References
o

Variables

Type	Explanation	Examples
int	represent integers	3, 4, 17, -10
float	represent real numbers	3.0, 1.11, -109.123123
bool	represent boolean truth values	True, False
str	A sequence of characters.	'Hello', ", '3'
NoneType	special and has one value, None	None

OK, but how do we create one?

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
o

4. Introduction
oooo●oooo

5. References
o

Variables

```
x = 2
x * 7
# => 14

x
# => 2
x = x * 7


y = 'Hello'
y + ' World!'
# => 'Hello World!'
```

How about type of variables?

Special method called `type()`

```
type(1) # => <class 'int'>
type('Hello') # => <class 'str'>
type(None) # => <class 'NoneType'>
type('') # => <class 'str'>

type(int) # => <class 'type'>
type(type(int)) # => <class 'type'>
```

Python knows variables' type even if you don't know it!

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○●○○○

5. References
○

Console I/O(Input/Output)

Now we can store the data we know,

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○●○○○

5. References
○

Console I/O(Input/Output)

Now we can store the data we know,
how about interacting with the user?

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
o

4. Introduction
oooooooo●oooo

5. References
o

Console I/O(Input/Output)

Now we can store the data we know,
how about interacting with the user?

```
print(), input()
```

Console I/O(Input/Output)

Now we can store the data we know,
how about interacting with the user?

```
print(), input()
```

```
# Print descriptive text to console
# and assign input to variable
name = input('Enter a sentence:')
# Greet user
print('Hello from Python,', name)
```

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
o

4. Introduction
oooooooo●○○

5. References
o

Console I/O(Input/Output)

```
print(*args, sep=' ', end='\n')
```

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○

4. Introduction
○○○○○○○●○○

5. References
○

Console I/O(Input/Output)

```
print(*args, sep=' ', end='\n')
```

- Can take arbitrary number of arguments

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
o

4. Introduction
oooooooo●○○

5. References
o

Console I/O(Input/Output)

```
print(*args, sep=' ', end='\n')
```

- Can take arbitrary number of arguments
- Separates elements with space by default

Console I/O(Input/Output)

```
print(*args, sep=' ', end='\n')
```

- Can take arbitrary number of arguments
- Separates elements with space by default
- Adds newline character '\n' to end by default

Console I/O(Input/Output)

```
print(*args, sep=' ', end='\n')
```

- Can take arbitrary number of arguments
- Separates elements with space by default
- Adds newline character '\n' to end by default

```
input([prompt])
```

Console I/O(Input/Output)

```
print(*args, sep=' ', end='\n')
```

- Can take arbitrary number of arguments
- Separates elements with space by default
- Adds newline character '\n' to end by default

```
input([prompt])
```

- Prints the prompt to Console

Console I/O(Input/Output)

```
print(*args, sep=' ', end='\n')
```

- Can take arbitrary number of arguments
- Separates elements with space by default
- Adds newline character '\n' to end by default

```
input([prompt])
```

- Prints the prompt to Console
- Program is paused until user enters something

Console I/O(Input/Output)

```
print(*args, sep=' ', end='\n')
```

- Can take arbitrary number of arguments
- Separates elements with space by default
- Adds newline character '\n' to end by default

```
input([prompt])
```

- Prints the prompt to Console
- Program is paused until user enters something
- **returns an str object!**

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
o

4. Introduction
oooooooooooo●o

5. References
o

Example Program

```
number = input('Rate us out of 100 :')
# Assume user entered 34
result = number + (100 - number)
# What will we see in console?
print(result)
```

Example Program

```
number = input('Rate us out of 100 :')
# Assume user entered 34
result = number + (100 - number)
# What will we see in console?
print(result)
```



1. Program Information
oooooooo

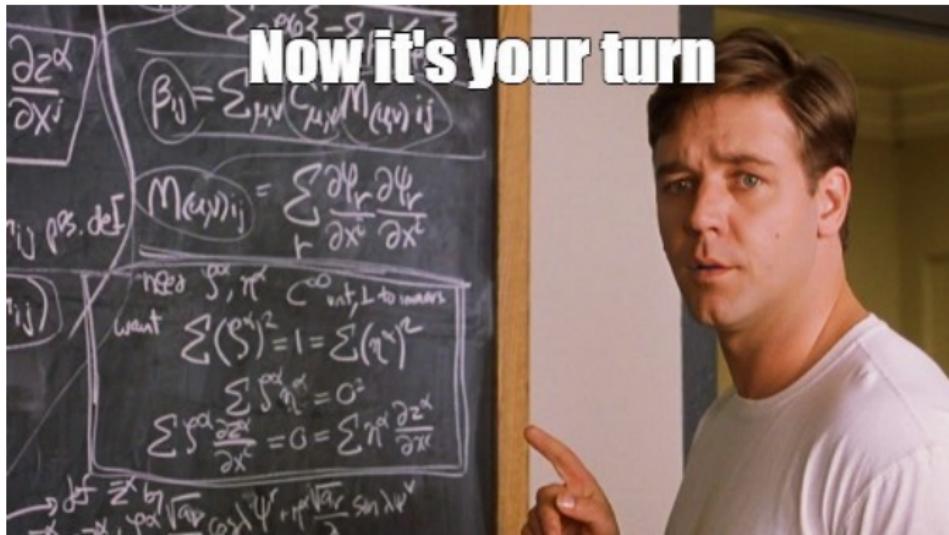
2. Logistics
oooo

3. Installations
o

4. Introduction
oooooooooooo●

5. References
o

Get Your Hands Dirty



1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
o

4. Introduction
oooooooooooo

5. References
●

References

- [1] P. Guo, "Python is now the most popular introductory teaching language at top u.s. universities." [Online]. Available: <https://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-u-s-univefulltext>
- [2] JetBrains, "Python developers survey 2018." [Online]. Available: <https://www.jetbrains.com/research/python-developers-survey-2018/>
- [3] M. Ni, "Web service efficiency at instagram with python." [Online]. Available: <https://instagram-engineering.com/web-service-efficiency-at-instagram-with-python-4976d078e366>