KOLT Python

Introduction

Ahmet Uysal

Monday 18th February, 2019

Agenda

1. Program Information

2. Logistics

3. Installations

4. Introduction

5. References

Course Outcomes

- Apply basic programming concepts using Python

Course Outcomes

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

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

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

Why Python?

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

Why Python?

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

```java
import java.util.Scanner;

public class Example{
    public static void main(String[] args){
        // Create a scanner object
        Scanner scanner = new Scanner(System.in);
        // Print descriptive text to console
        System.out.print("Enter your name: ");
        // Read user input and assign it to a variable
        String name = scanner.nextLine();
        // Greet user
        System.out.println("Hello from Java, " + name);
    }
}
```

Why Python?

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

Some of the Usage Areas[2]

- Data Analysis

Some of the Usage Areas[2]

- Data Analysis
- Web Development

Some of the Usage Areas[2]

- Data Analysis
- Web Development
- System Administration

Some of the Usage Areas[2]

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

Some of the Usage Areas[2]

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

Some of the Usage Areas[2]

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

Some of the Usage Areas[2]

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

Some of the Usage Areas[2]

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

Some of the Usage Areas[2]

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

Python at Koç University

- COMP341: Introduction to Artificial Intelligence

Python at Koç University

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

Python at Koç University

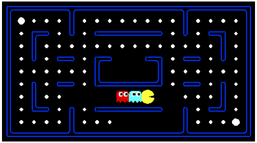
- COMP341: Introduction to Artificial Intelligence
- COMP421/521: Introduction to Machine Learning
- ENGR350(Selected Topics - Summer18/Spring19): Introduction to Programming for Data Science

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

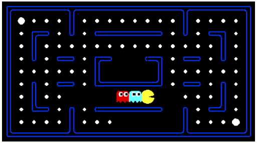
Python at Koç University

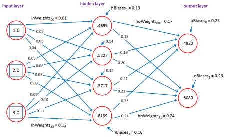
- COMP341: Introduction to Artificial Intelligence
- COMP421/521: Introduction to Machine Learning
- ENGR350(Selected Topics - Summer18/Spring19): Introduction to Programming for Data Science
- INTL450(Selected Topics - Spring19): Advanced Data Analysis in Python



Python at Koç University

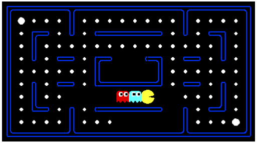
- COMP341: Introduction to Artificial Intelligence
- COMP421/521: Introduction to Machine Learning
- ENGR350(Selected Topics - Summer18/Spring19): Introduction to Programming for Data Science
- INTL450(Selected Topics - Spring19): Advanced Data Analysis in Python

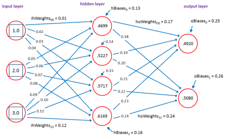




Python at Koç University

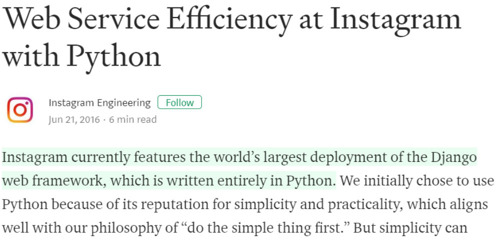
- COMP341: Introduction to Artificial Intelligence
- COMP421/521: Introduction to Machine Learning
- ENGR350(Selected Topics - Summer18/Spring19): Introduction to Programming for Data Science
- INTL450(Selected Topics - Spring19): Advanced Data Analysis in Python





Python at Industry

Python at Industry[3]

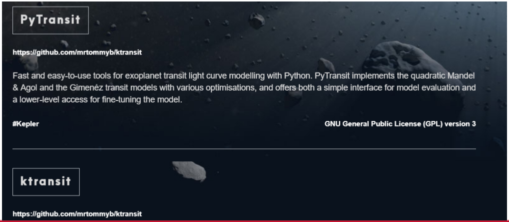


Web Service Efficiency at Instagram with Python

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/mrtommyb/ktransit

Fast and easy-to-use tools for exoplanet transit light curve modelling with Python. PyTransit implements the quadratic Mandel & Agol and the Gimenéz 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/mrtommyb/ktransit

Team

Ahmet Uysal
auysal16@ku.edu.tr

Team

Ahmet Uysal
auysal16@ku.edu.tr

İpek Köprülülü
ikoprululu16@ku.edu.tr

Weekly Schedule

`Monday 17:30-18:45` **`Lecture`**

Weekly Schedule

Monday	17:30–18:45	**Lecture**
Wednesday	17:30–18:45	**Contest/Review**

Weekly Schedule

Monday 17:30–18:45 **Lecture**

Wednesday 17:30–18:45 **Contest/Review**

Every Week **HackerRank Contest**

Weekly Schedule

Monday 17:30-18:45 **Lecture**
Wednesday 17:30-18:45 **Contest/Review**
Every Week **HackerRank Contest**

HackerRank

We will use HackerRank a lot this semester, create your account if you have not already!

Programming Assignments

- 4-6 Programming Assignments

Programming Assignments

- 4-6 Programming Assignments
- Review sessions will be conducted to **help you**

Programming Assignments

- 4-6 Programming Assignments
- Review sessions will be conducted to **help you**
- Some assignments will have **autograders** to help you find your mistakes and test your code.

Programming Assignments

- 4-6 Programming Assignments
- Review sessions will be conducted 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!**

Programming Assignments

- 4-6 Programming Assignments
- Review sessions will be conducted 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!**
- We can also organize hackathons if requested :)

Programming Contests

- Weekly Online Contests

Programming Contests

- Weekly Online Contests
  - 5-10 Programming questions related to topic we have seen

Programming Contests

- Weekly Online Contests
  - 5-10 Programming questions related to topic we have seen
  - Duration: 1 week (Starts after the lecture on Monday, ends before next lecture)

Programming Contests

- Weekly Online Contests
  - 5-10 Programming questions related to topic we have seen
  - Duration: 1 week (Starts after the lecture on Monday, ends before next lecture)
  - First contest will start just after this lecture

Programming Contests

- Weekly Online Contests
  - 5-10 Programming questions related to topic we have seen
  - Duration: 1 week (Starts after the lecture on Monday, ends before next lecture)
  - First contest will start just after this lecture
- Biweekly Onsite Contests
  - Wednesdays, on weeks which you don’t have an assignment

Programming Contests

- Weekly Online Contests
  - 5-10 Programming questions related to topic we have seen
  - Duration: 1 week (Starts after the lecture on Monday, ends before next lecture)
  - First contest will start just after this lecture
- Biweekly Onsite Contests
  - Wednesdays, on weeks which you don’t have an assignment
  - 2-3 Programming questions related to Monday’s lecture

Programming Contests

- Weekly Online Contests
  - 5-10 Programming questions related to topic we have seen
  - Duration: 1 week (Starts after the lecture on Monday, ends before next lecture)
  - First contest will start just after this lecture
- Biweekly Onsite Contests
  - Wednesdays, on weeks which you don't have an assignment
  - 2-3 Programming questions related to Monday's lecture
  - Duration: 50 minutes

Programming Contests

- Weekly Online Contests
  - 5-10 Programming questions related to topic we have seen
  - Duration: 1 week (Starts after the lecture on Monday, ends before next lecture)
  - First contest will start just after this lecture
- Biweekly Onsite Contests
  - Wednesdays, on weeks which you don’t have an assignment
  - 2-3 Programming questions related to Monday’s lecture
  - Duration: 50 minutes
  - Questions will be solved in remaining 25 minutes

Programming Contests

- Weekly Online Contests
  - 5-10 Programming questions related to topic we have seen
  - Duration: 1 week (Starts after the lecture on Monday, ends before next lecture)
  - First contest will start just after this lecture
- Biweekly Onsite Contests
  - Wednesdays, on weeks which you don’t have an assignment
  - 2-3 Programming questions related to Monday’s lecture
  - Duration: 50 minutes
  - Questions will be solved in remaining 25 minutes
  - Surprize gifts to first three places and problem solvers!

Certificate Requirements

Certificate Requirements

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

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.

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 it!

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 it!
- Complying to *Koç University Code of Conduct*.

5 Minute Break

Installing Python

- Go to *python.org/downloads*

Installing Python

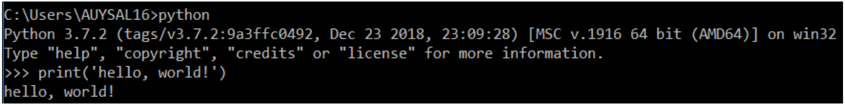
- Go to *python.org/downloads*
- Install the Python 3.7.2 for your operating system

Installing Python

- Go to *python.org/downloads*
- Install the Python 3.7.2 for your operating system
- (Windows only) Make sure to add python to the `environment variables` by checking the corresponding permission on the installation or by hand

Installing Python

- Go to *python.org/downloads*
- Install the Python 3.7.2 for your operating system
- (Windows only) Make sure to add python to the `environment variables` by checking the corresponding permission on the installation or by hand
- Check the installation by running **`python`**(Windows)/**`python3`**(macOS/Linux) in terminal.



```
C:\Users\AUYSAL16>python
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print('hello, world!')
hello, world!
```

Installing an Editor/IDE(Integrated Development Environment)

- Although you can edit Python(.py) files with any text editor and run them directly through terminal, having a specialized editor/IDE can help a lot.
- We will use Visual Studio Code in lectures but you are free to use any editor/IDE of your choice.
- Get Visual Studio Code from *code.visualstudio.com/Download*

Configuring Visual Studio Code for Python

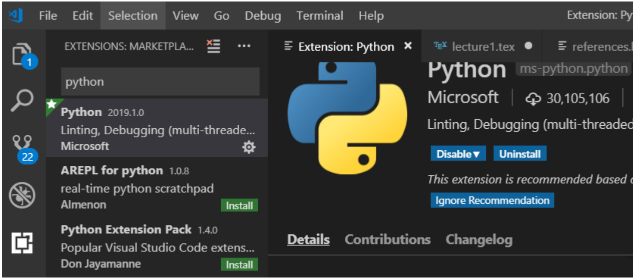
- Install Python extension for VS Code.

Configuring Visual Studio Code for Python

- Install Python extension for VS Code.
- Select the Python(3.7.2) Interpreter in VS Code.

Configuring Visual Studio Code for Python

- Install Python extension for VS Code.
- Select the Python(3.7.2) Interpreter in VS Code.
- For more information, visit *VS Code Python Tutorial*.



Python 2019.1.0
Linting, Debugging (multi-threade...
Microsoft

AREPL for python 1.0.8
real-time python scratchpad
Almenon Install

Python Extension Pack 1.4.0
Popular Visual Studio Code extens...
Don Jayamanne Install

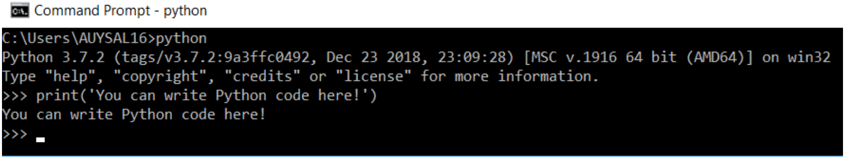
Python ms-python.python
Microsoft | 30,105,106
Linting, Debugging (multi-threaded
Disable ▾ Uninstall
This extension is recommended based
Ignore Recommendation
Details Contributions Changelog

Interactive Interpreter

You can instantly run code on terminal!

Interactive Interpreter

You can instantly run code on terminal!

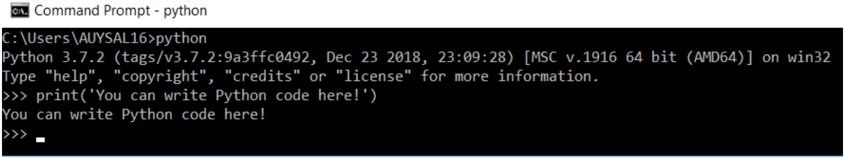


Command Prompt - python

```
C:\Users\AUYSAL16>python
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print('You can write Python code here!')
You can write Python code here!
>>> _
```

Interactive Interpreter

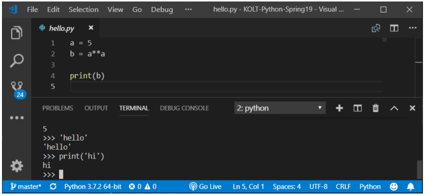
You can instantly run code on terminal!



Command Prompt - python

```
C:\Users\AUYSAL16>python
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print('You can write Python code here!')
You can write Python code here!
>>> _
```

You can also open a terminal inside VS Code



File Edit Selection View Go Debug ···
hello.py - KOLT-Python-Spring19 - Visual ...
hello.py
24
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
2: python
master* Python 3.7.2 64-bit ⊗ 0 ⚠ 0 Go Live Ln 5, Col 1 Spaces: 4 UTF-8 CRLF Python

```
a = 5
b = a**a

print(b)

```

```
5
>>> 'hello'
'hello'
>>> print('hi')
hi
>>> 
```

Why It Matters?

Why It Matters?

- Immediate gratification :)

Why It Matters?

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

Why It Matters?

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

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

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**!

Variables

- How to represent/store values in Python?

Variables

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

Variables

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

Variables

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

Variables

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

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?

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?

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?

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?

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`

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?

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!

Console I/O(Input/Output)

Now we can store the data we know,

Console I/O(Input/Output)

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

Console I/O(Input/Output)

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

`print(), input()`

Console I/O(Input/Output)

Now we can store the data we know,
how about interacting with 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)
```

Console I/O(Input/Output)

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

Console I/O(Input/Output)

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

- Can take arbitrary number of arguments

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!**

Example Program

```
number = input('Please enter a number:')
# Assume user entered 34
result = number * 2
# What will we see in console?
print(result)
```

Example Program

```
number = input('Please enter a number:')
# Assume user entered 34
result = number * 2
# What will we see in console?
print(result)
```

LET’S TRY!!!

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-unive fulltext

[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