KOLT Python

Introduction

Ahmet Uysal

Tuesday 28th January, 2020

KOÇ UNIVERSITY
OFFICE OF LEARNING AND TEACHING

Agenda

1. Program Information
2. Logistics
3. Installations
4. Introduction
5. References

Course Outcomes

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?

Why Python?

- Easy Syntax

Why Python?

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

Why Python?

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

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]

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/Crawlers

Some of the Usage Areas [2]

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

Some of the Usage Areas [2]

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

Some of the Usage Areas [2]

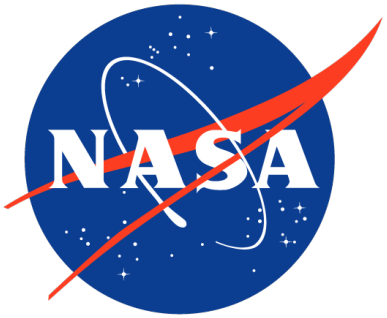
- Data Analysis
- Web Development
- System Administration
- Machine Learning
- Web Parsers/Crawlers
- Testing
- Education
- Network Programming

Some of the Usage Areas [2]

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

Python at Industry

Google



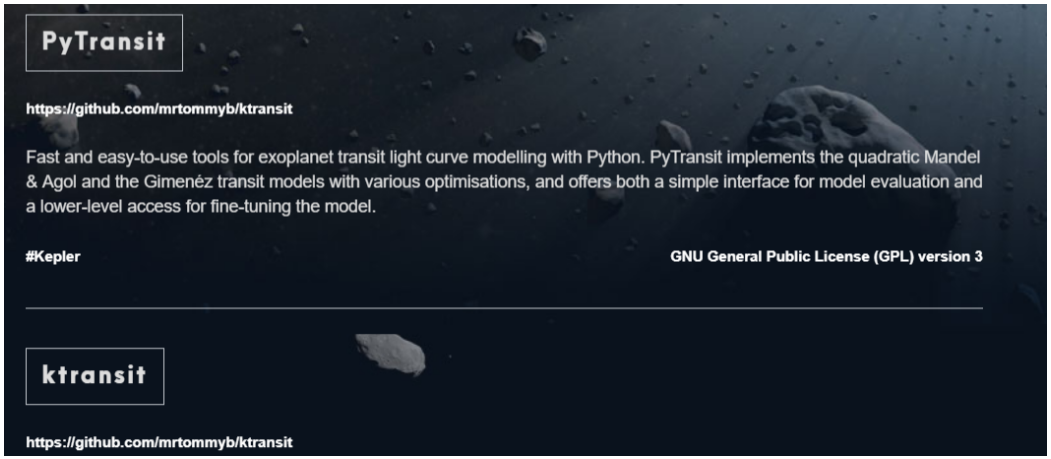
YouTube

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



Python Everywhere

Who Are We?

Ahmet Uysal

Who Are We?

Ahmet Uysal

Ceren Kocaoğullar

Who Are We?

Ahmet Uysal

Ceren Kocaoğullar

Hasan Can Aslan

Who Are We?

Ahmet Uysal

Ceren Kocaoğullar

Hasan Can Aslan

Necla Mutlu

Who Are We?

Ahmet Uysal

Ceren Kocaoğullar

Hasan Can Aslan

Necla Mutlu

Fırat Tamur

Who Are We?

Ahmet Uysal

Ceren Kocaoğullar

Hasan Can Aslan

Necla Mutlu

Fırat Tamur

Ayşe Turşucular

Who Are We?

Ahmet Uysal

Ceren Kocaoğullar

Hasan Can Aslan

Necla Mutlu

Fırat Tamur

Ayşe Turşucular

Haluk Ceyhun Gün

Who Are We?

Ahmet Uysal

Ceren Kocaoğullar

Hasan Can Aslan

Necla Mutlu

Fırat Tamur

Ayşe Turşucular

Haluk Ceyhun Gün

Halil Eralp Koçaş

What Will We Do?

What Will We Do?

Lecture `Tuesday 14:30-15:45` @SNA 158

- New concepts about Python & programming in general
-

What Will We Do?

`Lecture` `Tuesday 14:30-15:45` @SNA 158

- New concepts about Python & programming in general
-

`Section`

- Starting next week
- You will receive a Form with possible Section slots
- Please select **every** option that suits you
- Exercise questions for **you** to solve

We need your feedback to get better, don't hesitate to talk with us!

Programming Assignments

Programming Assignments

- 3-4 in total

Programming Assignments

- 3-4 in total
- Review sessions to **help you**

Programming Assignments

- 3-4 in total
- Review sessions to **help you**
- Some assignments will have **autograders** to help you find your mistakes and test your code.

Programming Assignments

- 3-4 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!** Feel free to recommend subjects.

Certificate Requirements

Certificate Requirements

- At most **3 unexcused absences**, including sections.

Certificate Requirements

- At most **3 unexcused absences**, including sections.
- 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 sections.
- 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!

Certificate Requirements

- At most **3 unexcused absences**, including sections.
- 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*.

Extra

We will also organize complementary events throughout the semester,

- Workshops
- Hackathons
- Programming Contests
- . . .

These are not mandatory, but we highly encourage you to attend.

Stay tuned :)

Attendance

Attendance

Fill out the attendance form:

tiny.cc/koltpython

Attendance

Fill out the attendance form:

tiny.cc/koltpython

Password: **Canada**

Installing Python

- Go to *python.org/downloads*

Installing Python

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

Installing Python

- Go to *python.org/downloads*
- Install the latest version of Python 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 latest version of Python 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\auysal>python
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:37:50) [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)

- Having a specialized editor/IDE can help a lot.
- We will use Visual Studio Code 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

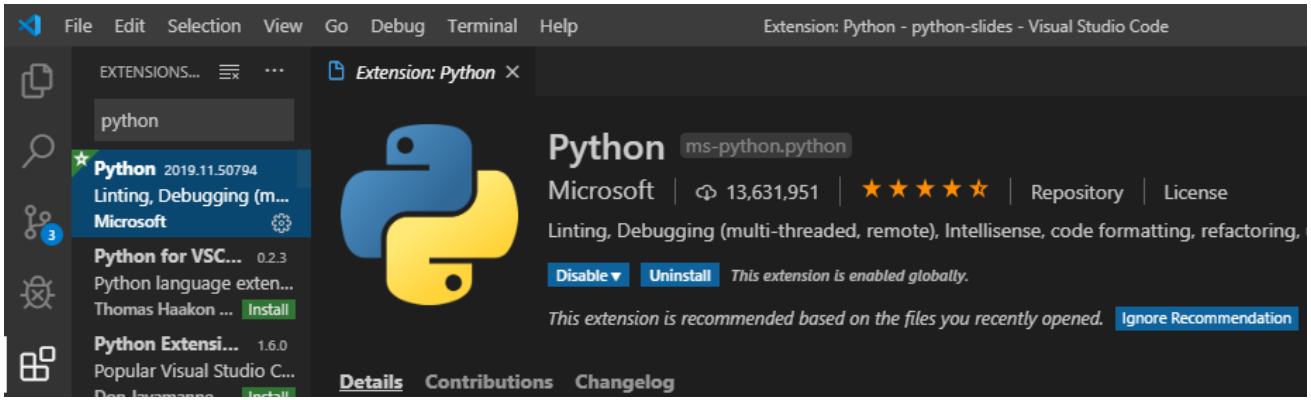
- Install Python extension for VS Code.

Configuring Visual Studio Code for Python

- Install Python extension for VS Code.
- Select the Python(3.8.1) Interpreter in VS Code.

Configuring Visual Studio Code for Python

- Install Python extension for VS Code.
- Select the Python(3.8.1) Interpreter in VS Code.
- For more information, visit *VS Code Python Tutorial*.



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

Variables

- How to represent/store values in Python?

Variables

- How to represent/store values in Python?
- The answer is:
- **VARIABLES!**

Variables

- How to represent/store values in Python?
- The answer is:
- **VARIABLES!**

- But what are variables?

Variables

Think of them as

Variables

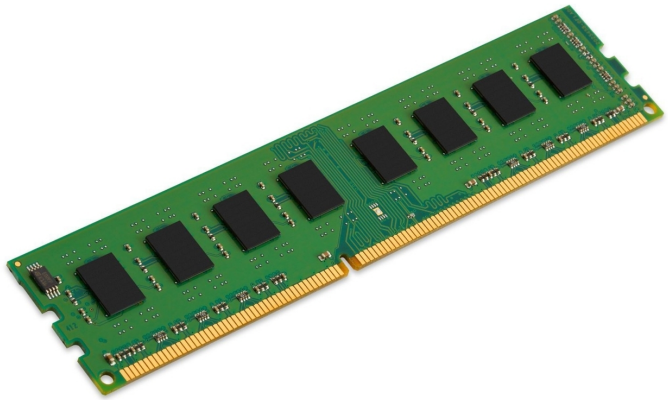
Think of them as

Variables

Tiny boxes in the computer memory

Variables

Tiny boxes in the computer memory



Variables

- Which kind of values we need to keep in these boxes (variables)?

Variables

- Which kind of values we need to keep in these boxes (variables)?
  - Numbers?

Variables

- Which kind of values we need to keep in these boxes (variables)?
  - Numbers?
  - Texts?

Variables

- Which kind of values we need to keep in these boxes (variables)?
  - Numbers?
  - Texts?
  - Individual Characters?

Variables

- Which kind of values we need to keep in these boxes (variables)?
  - Numbers?
  - Texts?
  - Individual Characters?
  - Starting time of the class?

Variables

- Which kind of values we need to keep in these boxes (variables)?
  - Numbers?
  - Texts?
  - Individual Characters?
  - Starting time of the class?
  - Colors?

Variables

- Which kind of values we need to keep in these boxes (variables)?
  - Numbers?
  - Texts?
  - Individual Characters?
  - Starting time of the class?
  - Colors?
  - Truth Values?

Variables

- Which kind of values we need to keep in these boxes (variables)?
  - 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!'
```

Variables

Variables

- Andy could fit all his toys into one box, but this is not the Toy Story.

Variables

- Andy could fit all his toys into one box, but this is not the Toy Story.
- These variable boxes can keep only **one thing at a time.**

How about types 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 the user?

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()`

Let's write our first program!

Console I/O(Input/Output)

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

`print(), input()`

Let's write our first program!

```
# 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('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)
```

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