

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo

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
oooo

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

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

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?



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/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
 - ...

1. Program Information

2. Logistics

3. Installations

4. Introduction

5. References

Python at Koç University

1. Program Information

2. Logistics

3. Installations

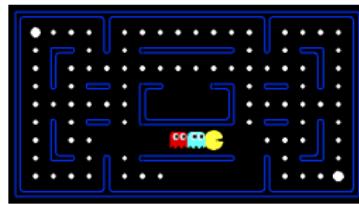
4. Introduction



5. References

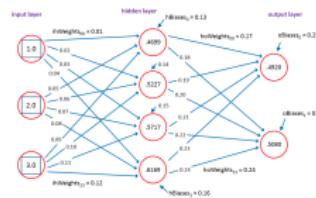
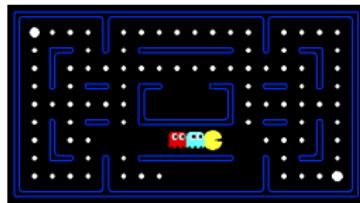
Python at Koç University

- COMP341: Introduction to Artificial Intelligence



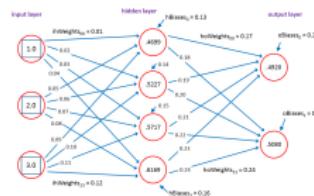
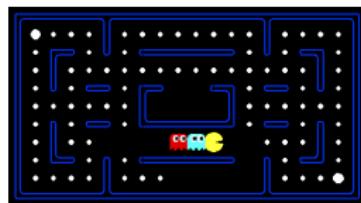
Python at Koç University

- COMP341: Introduction to Artificial Intelligence
 - ENGR421/521: Introduction to Machine Learning



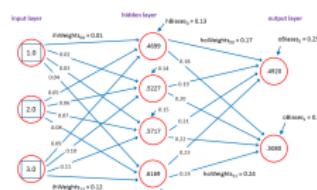
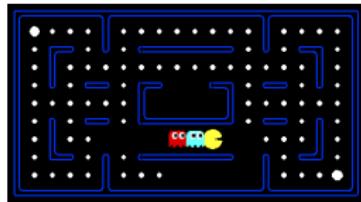
Python at Koç University

- COMP341: Introduction to Artificial Intelligence
 - ENGR421/521: Introduction to Machine Learning
 - ENGR350 (Selected Topics - Summer18/Spring19): Introduction to Programming for Data Science



Python at Koç University

- COMP341: Introduction to Artificial Intelligence
- ENGR421/521: 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
oooooo

2. Logistics
oooo

3. Installations
oooo

4. Introduction
oooooooooooo

5. References
o

Python at Industry



Google



YouTube



1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo

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/mrtommyb/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/mrtommyb/ktransit>

1. Program Information
oooooooo●

2. Logistics
oooo

3. Installations
oooo

4. Introduction
oooooooooooo

5. References
o

Python Everywhere



imgflip.com

1. Program Information
○○○○○○○

2. Logistics
●○○○

3. Installations
○○○○

4. Introduction
○○○○○○○○○○○○○○○○

5. References
○

Who Are We?



Ahmet Uysal
auysal16@ku.edu.tr

1. Program Information
oooooooo

2. Logistics
●ooo

3. Installations
oooo

4. Introduction
oooooooooooo

5. References
o

Who Are We?



Ahmet Uysal
auysal16@ku.edu.tr



Ceren Kocaoğullar
ckocaogullar15@ku.edu.tr

1. Program Information
oooooooo

2. Logistics
●ooo

3. Installations
oooo

4. Introduction
oooooooooooo

5. References
o

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
●○○○

3. Installations
○○○○

4. Introduction
oooooooooooo

5. References
○

Who Are We?



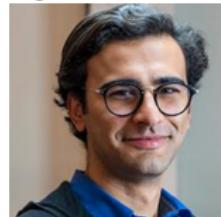
Ahmet Uysal
auysal16@ku.edu.tr



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?

1. Program Information

2. Logistics

3. Installations

4. Introduction



5. References

What Will We Do?

Lecture Monday 11:30-12:45

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(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

2. Logistics

3. Installations

4. Introduction



5. References

Programming Assignments(Don't Be Afraid!)

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

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.

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!**

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 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*.

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
●ooo

4. Introduction
oooooooooooo

5. References
o

Installations

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
●ooo

4. Introduction
oooooooooooo

5. References
o

Installations

NO INSTALLATIONS!



1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
●ooo

4. Introduction
oooooooooooo

5. References
o

Installations

NO INSTALLATIONS!



We will use Repl.it, an online IDE.



1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
o●oo

4. Introduction
oooooooooooo

5. References
o

Sign Up to Repl.it

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○●○○

4. Introduction
○○○○○○○○○○○○○○○○

5. References
○

Sign Up to Repl.it

We already sent you an invite, check your email.

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
o●○○

4. Introduction
oooooooooooooooo

5. References
o

Sign Up to Repl.it

We already sent you an invite, check your email.

Repl.it: Join classroom KOLT Python Certificate Program 2019 Inbox



Repl.it <notifications@repl.it> [Unsubscribe](#)
to me ▾

7:26 PM (0 minutes ago)



Hi there,

AHMET UYSAL invited you to participate in classroom "KOLT Python Certificate Program 2019" on Repl.it.

[Join Classroom](#)

Repl.it is the easiest way to get start with programming. It's a cloud coding environment for all major programming languages and a platform for teaching and learning programming.

Good luck! And let us know (contact@repl.it) if you have any problems using our product.

[Unsubscribe](#)

Repl.it, 767 Bryant St #210, San Francisco CA 94107

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo

4. Introduction
oooooooooooo

5. References
o

Sign Up Flow

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo

4. Introduction
oooooooooooo

5. References
o

Sign Up Flow



username	
email	
password	

I'm a teacher

or log in

Sign up

By continuing, you agree to Repl.it's Terms of Service and Privacy Policy, and to receiving emails with updates.

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo

4. Introduction
oooooooooooo

5. References
o

Sign Up Flow



username x

email x

password x

I'm a teacher or log in

Sign up

By continuing, you agree to Repl.it's Terms of Service and Privacy Policy, and to receiving emails with updates.

Sign in with Google

Choose an account
to continue to [repl.it](#)

 Samed Biçer
mbicer14@ku.edu.tr

 [Use another account](#)

To continue, Google will share your name, email address, language preference, and profile picture with repl.it.

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo

4. Introduction
oooooooooooo

5. References
o

Sign Up Flow



username x

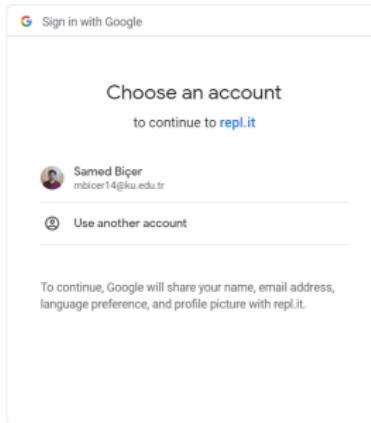
email x

password x

I'm a teacher or log in

Sign up

By continuing, you agree to Repl.it's Terms of Service and Privacy Policy, and to receiving emails with updates.



Classrooms

PYTHON
Certificate Program

KOLT Python Certificate Program 201...

Python

1 assignments

⋮

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo●

4. Introduction
oooooooooooo

5. References
o

Repl.it

back to edit run ►

```
1 # Welcome to KOLT Python Certificate Program
2 # Lines that begin with '#' symbol are single line comments.
3 # Python will ignore these lines. They are purely for humans, like you.
4 """
5 You can also see multiline comments.
6 They start and end with triple quotes.
7 Usually comments are used to document the code and
8 make it more easy to understand.
9 However, we will use comments to guide you or give instructions
10 in exercises and assignments.
11 """
12
13 # modify this function to print your name
14- def print_name():
15     print("Hello, world!")
16
17 print_name()
```

```
Python 3.7.4 (default, Jul  9 2019, 00:06:43)
[GCC 6.3.0 20170516] on linux
>|
```

Due: --

submit

Instructions from your teacher:

Welcome Coder!

This is your first task. Please modify `print_name` function as described.

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo●

4. Introduction
oooooooooooo

5. References
o

Repl.it

back to edit run ▶

```
1 # Welcome to KOLT Python Certificate Program
2 # Lines that begin with '#' symbol are single line comments.
3 # Python will ignore these lines. They are purely for humans, like you.
4 """
5 You can also see multiline comments.
6 They start and end with triple quotes.
7 Usually comments are used to document the code and
8 make it more easy to understand.
9 However, we will use comments to guide you or give instructions
10 in exercises and assignments.
11 """
12
13 # modify this function to print your name
14- def print_name():
15     print("Hello, world!")
16
17 print_name()
```

Python 3.7.4 (default, Jul 9 2019, 00:06:43)
[GCC 6.3.0 20170516] on linux
▶

Due: -- submit

Instructions from your teacher:

Welcome Coder!

This is your first task. Please modify `print_name` function as described.

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo●

4. Introduction
oooooooooooo

5. References
o

Repl.it

4 back to edit run ▶ Due: -- Submit

```
1 # Welcome to KOLT Python Certificate Program!
2 # Lines that begin with '#' symbol are single line comments.
3 # Python will ignore these lines. They are purely for humans, like you.
4 """
5 You can also see multiline comments.
6 They start and end with triple quotes.
7 Usually comments are used to document the code and
8 make it more easy to understand.
9 However, we will use comments to guide you or give instructions
10 in exercises and assignments.
11 """
12
13 # modify this function to print your name
14 def print_name():
15     print('Hello, world!')
16
17 print_name()
```

Instructions from your teacher:
Welcome Coder!
This is your first task. Please modify `print_name` function as described.

Python 3.7.4 (default, Jul 9 2019, 08:06:43)
[GCC 6.3.0 20170516] on linux
▶

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo●

4. Introduction
oooooooooooo

5. References
o

Repl.it

The screenshot shows a Repl.it interface. On the left, there's a code editor with Python code. A red circle highlights the 'run' button. On the right, there's a terminal window showing a Linux environment with Python 3.7.4 installed.

```
1 # Welcome to KOLT Python Certificate Program
2 # Lines that begin with '#' symbol are single line comments.
3 # Python will ignore these lines. They are purely for humans, like you.
4 """
5 You can also see multiline comments.
6 They start and end with triple quotes.
7 Usually comments are used to document the code and
8 make it more easy to understand.
9 However, we will use comments to guide you or give instructions
10 in exercises and assignments.
11 """
12
13 # modify this function to print your name
14- def print_name():
15     print("Hello, world!")
16
17 print_name()
```

```
Python 3.7.4 (default, Jul  9 2019, 00:06:43)
[GCC 6.3.0 20170516] on linux
> |
```

Due: -- submit

Instructions from your teacher:
Welcome Coder!
This is your first task. Please modify `print_name` function as described.

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo●

4. Introduction
oooooooooooo

5. References
o

Repl.it

The screenshot shows the Repl.it interface. On the left, there is a code editor with the following Python code:

```
1 # Welcome to KOLT Python Certificate Program!
2 # Lines that begin with '#' symbol are single line comments.
3 # Python will ignore these lines. They are purely for humans, like you.
4 """
5 You can also see multiline comments.
6 They start and end with triple quotes.
7 Usually comments are used to document the code and
8 make it more easy to understand.
9 However, we will use comments to guide you or give instructions
10 in exercises and assignments.
11 """
12
13 # modify this function to print your name
14 def print_name():
15     print('Hello, world!')
16
17 print_name()
```

On the right, there is a terminal window with a red border containing the output of the code:

```
Python 3.7.4 (default, Jul  9 2019, 00:06:43)
[GCC 6.3.0 20170916] on linux
> ●
```

At the top right of the interface, there is a green "submit" button.

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo

4. Introduction
●oooooooooooo

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
oooooooo

2. Logistics
oooo

3. Installations
oooo

4. Introduction
○●oooooooooooo

5. References
○

Variables

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo

4. Introduction
○●oooooooooooo

5. References
○

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 the heck are variables?

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo

4. Introduction
oo●oooooooooooo

5. References
o

Variables

Think of them as

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo

4. Introduction
oo●oooooooooooo

5. References
o

Variables

Think of them as



1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo

4. Introduction
oooo●oooooooooooo

5. References
o

Variables

Tiny boxes in the computer memory

1. Program Information
oooooooo

2. Logistics
oooo

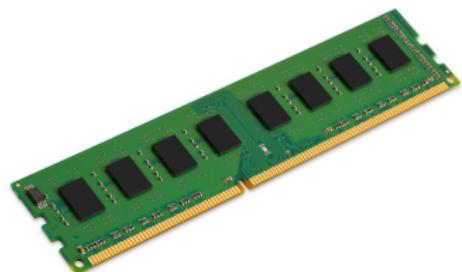
3. Installations
oooo

4. Introduction
oooo●oooooooooooo

5. References
o

Variables

Tiny boxes in the computer memory



1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○○○○

4. Introduction
○○○○●○○○○○○○○

5. References
○

Variables

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

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○○○○

4. Introduction
○○○○●○○○○○○○○

5. References
○

Variables

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

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○○○○

4. Introduction
○○○○●○○○○○○○○

5. References
○

Variables

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

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○○○○

4. Introduction
○○○○●○○○○○○○○

5. References
○

Variables

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

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○○○○

4. Introduction
○○○○●○○○○○○○○

5. References
○

Variables

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

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○○○○

4. Introduction
○○○○●○○○○○○○○

5. References
○

Variables

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

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○○○○

4. Introduction
○○○○●○○○○○○○○

5. References
○

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?

1. Program Information
○○○○○○○

2. Logistics
○○○○

3. Installations
○○○○

4. Introduction
○○○○●○○○○○○○○

5. References
○

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?

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo

4. Introduction
oooo●oooo

5. References
○

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
oooo

4. Introduction
oooo●oooo

5. References
○

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
oooo

4. Introduction
oooooooo●oooooooo

5. References
o

Variables

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

x
# => 2
x = x * 7


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

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo

4. Introduction
oooooooo●oooo

5. References
o

Variables

- BE CAREFUL!

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo

4. Introduction
oooooooo●oooo

5. References
o

Variables

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

Variables

- BE CAREFUL!
 - 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!

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo

4. Introduction
oooooooooooo●oooo

5. References
o

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
oooo

4. Introduction
oooooooooooo●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
oooo

4. Introduction
oooooooooooo●○○

5. References
○

Console I/O(Input/Output)

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

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo

4. Introduction
oooooooooooo●oo

5. References
o

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!**

1. Program Information
oooooooo

2. Logistics
oooo

3. Installations
oooo

4. Introduction
oooooooooooo●○

5. References
○

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
oooo

4. Introduction
oooooooooooo●○

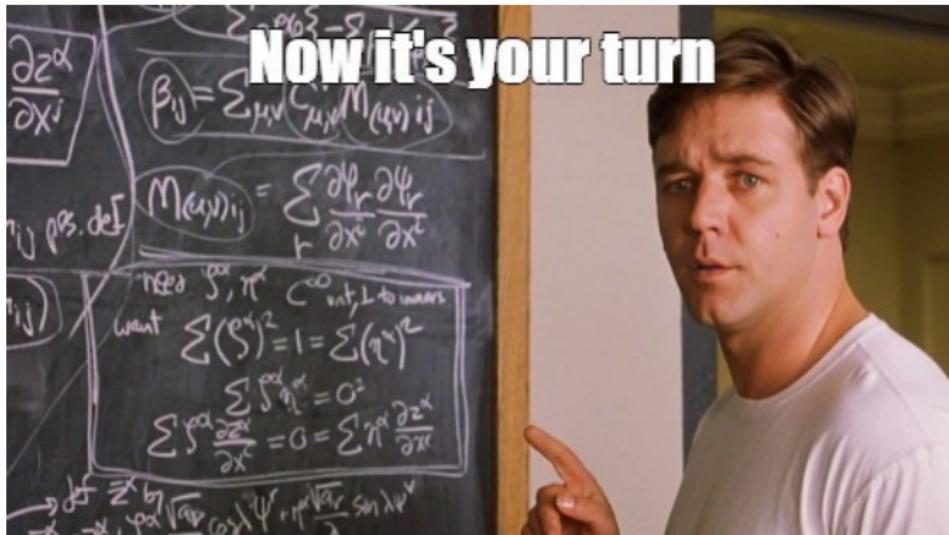
5. References
○

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)
```



Get Your Hands Dirty



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>