

All Weeks



Week 1

Week 2

Week 3

Week 4

Week 5

Week 6

Week 7

Week 8

Week 9

Week 10

Week 11

Week 12

Search



Hey

[Slides](#)[Resources](#)

Object-Oriented Python I

Let's check out an example in Python

We have a class Car

Class names are capitalized, and singular (e.g., we don't have a class Cars)

Classes inherit from object because any new class you write will be a type of object.

We give our class some attributes, e.g., speed, direction, color. Every new car we make will start off with speed 0, direction 0 and color red.

You instantiate a new instance of class Car by calling Car(), which you can assign to a variable.

c is now a Car object stored in a location in your computer's memory (the string of ugly letters and numbers)

Object-Oriented Python I

Another Example Slide

We have a class Car

Class names are capitalized, and singular (e.g., we don't have a class Cars)

Classes inherit from object because any new class you write will be a type of object.

We give our class some attributes, e.g., speed, direction, color. Every new car we make will start off with speed 0, direction 0 and color red.

You instantiate a new instance of class Car by calling Car(), which you can assign to a variable.

c is now a Car object stored in a location in your computer's memory (the string of ugly letters and numbers)

Hey



Slides

Resources

Let's check out an example in Python



Mike Godlewski

Hey I'm leaving a comment on this.
Check out my solution to the exercise
and this awesome link.
<http://css-tricks.com/svg-shape-morphing-works/>

 My awesome solution
Posted on Friday at 6:00pm

A different slide possibly from a different deck or week



Rudy Mutter

Hey I'm leaving a comment on this.
Check out my solution to the exercise
attached below.

 My awesome solution
Posted on Friday at 6:00pm

A different slide possibly from a different deck or week



Mike Godlewski

Hey I'm leaving a comment on this.
Check out my solution to the exercise
and this awesome link.
<http://css-tricks.com/svg-shape-morphing-works/>

 My awesome solution
Posted on Friday at 6:00pm

All Weeks



Week 1

Week 2

Week 3

Week 4

Week 5

Week 6

Week 7

Week 8

Week 9

Week 10

Week 11

Week 12

Week 1

Welcome and Introductions

Python Programs

Object-Oriented Python I

Object-Oriented Python II: Inheritance

Object-Oriented Python II: Methods

Let's check out an example in Python

We have a class Car

Class names are capitalized, and singular (e.g., we don't have a class Cars)

Classes inherit from `object` because any new class you write will be a type of object.

We give our class some *attributes*, e.g., `speed`, `direction`, `color`. Every new car we make will start off with speed 0, direction 0 and color red.

You *instantiate* a new instance of class Car by calling `Car()`, which you can assign to a variable.

`c` is now a Car object stored in a location in your computer's memory (the string of ugly letters and numbers)

bash

```
>>> class Car(object):  
...     speed = 0  
...     direction = 0 # degrees from N  
...     color = "red"  
...  
>>> c = Car()  
>>> c  
<__main__.Car object at 0x1091bdbd0>  
>>>
```

1 - 1 Let's check out an example in Python

1 - 2 Let's check out an example in Python

1 - 3 Let's check out an example in Python



Mike Godlewski

Hey I'm leaving a comment on this.
Check out my solution to the exercise
and this awesome link.

 My awesome solution

Posted on Friday at 6:00pm



Rudy Mutter

Hey I'm leaving a comment on this.
Check out my solution to the exercise
and this awesome link.

 My awesome solution

Posted on Friday at 6:00pm

4 - 1 Another slide with a name

4 - 2 Another slide with a name

5 - 1 Another slide with a name

6 - 1 Another slide with a name

6 - 2 Another slide with a name



Let's check out an example in Python

We have a class Car

Class names are capitalized, and singular (e.g., we don't have a class Cars)

Classes inherit from `object` because any new class you write will be a type of object.

We give our class some *attributes*, e.g., `speed`, `direction`, `color`. Every new car we make will start off with speed 0, direction 0 and color red.

You *instantiate* a new instance of class Car by calling `Car()`, which you can assign to a variable.

`c` is now a Car object stored in a location in your computer's memory (the string of ugly letters and numbers)

bash

```
>>> class Car(object):  
...     speed = 0  
...     direction = 0 # degrees from N  
...     color = "red"  
...  
>>> c = Car()  
>>> c  
<__main__.Car object at 0x1091bdbd0>  
>>>
```

Text area content

Drag Files Here

Save

