COMP 1012

Objects intro

Robert Guderian

Today: Objects

We've seen them!

String, List, Dict, File

Objects are another way of representing data

OO

Object-orientation is a big concept in programming.

```
someString.split()
```

This calls a method (which is a function attached to an object) on an object.

What is an object

A noun!

What is an object

A noun!

- Something we can describe

What is an object

A noun!

- Something we can describe
- What could a person object look like?

What is an object

A noun!

- Something we can describe
- What could a person object look like?
- What about coffee cup?

What is an object

A noun!

- Something we can describe
- What could a person object look like?
- What about coffee cup?
- Car?

The syntax

(There is always new syntax…)

We describe the DNA of our noun, by describing the class. Ask: what do ALL the entities have in common

```
class Human:
    name = "Insert name here"
    age = -1
```

How do we work with it?

```
myHuman = Human()
# set the name
myHuman.name = "Alex"
print(myHuman.age)
```

We're not all named Alex

So, why are we setting everyone's name to "Alex" in our version of `Human`?

We're not all named Alex

So, why are we setting everyone's name to "Alex" in our version of `Human`?

- That's dumb.

We're not all named Alex

So, why are we setting everyone's name to "Alex" in our version of `Human`?

- That's dumb.
- So, let's not do it!

The constructor

Lets us have a way to set default values

```
class Human:
    name = "Insert name here"
    age = -1
    def __init__(self, theName, theAge):
        self.name = theName
        self.age = theAge
```

The constructor

Lets us have a way to set default values

```
class Human:
    name = "Insert name here"
    age = -1
    def __init__(self, theName, theAge):
        self.name = theName
        self.age = theAge
```

- What?

def __init__

Called a ‘dunder’ method (don’t remember that), magic method or double-underscore method. It’s a magic built-in method.

It’s called when we do this:

```
h = Human("Alex", 7)
```

But, what about `self`

That's a "self referential pointer".

But, what about `self`

That's a "self referential pointer".

- literally, this object

But, what about `self`

That's a "self referential pointer".

- literally, this object
- me, myself

Demo

University class object. What can we set when we create the object?

Can we have defaults, and a constructor?

Yeah!

```
class Human:
    age = 0 # born today
    def __init__(self, name)
        self.name = name # namespace collision!
```

Summary

We can represent data with objects

It keeps all the data in one place, for one thing

```
if(condition)
```

```
if(condition == true)
```

```
if(String.valueof
   (condition).equals("true"))
```

// reveal.js plugins