DigitalCrafts

EXPLORER

DIGITALCRAFTS

- .vscode
- Week1
- Week2
  - Day 1
  - Day 2

Show All Commands ⇧⌘P

Go to File ⌘T

Find in Files ⇧⌘F

Start Debugging F5

Toggle Terminal ^`

Creating a Class!

PROJECTS

0 0 Combo: 3

person_class.py — DigitalCrafts

EXPLORER

person_class.py

DIGITALCRAFTS

- .vscode
- Week1
- Week2
  - Day 1
  - Day 2
    - person_class.py

```
1
```

PROJECTS

0 0 Combo: 3

Ln 1, Col 1 Space

First we want to create a new file to store our class in

person_class.py — DigitalCrafts

EXPLORER

DIGITALCRAFTS
- .vscode
- Week1
- Week2
  - Day 1
  - Day 2
    - person_class.py

person_class.py

```
1  class Person(object):
2      ....
```

Like a function, we want to make a new definition but instead of 'def' to define a function, we use 'class' to define a class.

PROJECTS

0 0 Combo: 2 Ln 2, Col 5 Space

person_class.py — DigitalCrafts

EXPLORER

DIGITALCRAFTS
- .vscode
- Week1
- Week2
  - Day 1
  - Day 2
    - person_class.py

person_class.py

```
1  class Person(object):
2      def __init__(self):
3          
```

Now, inside of that class, we can create our methods (functions) and properties (data).

PROJECTS

0 0 Combo: 1 Ln 3, Col 9 Space

person_class.py — DigitalCrafts

EXPLORER

DIGITALCRAFTS

- .vscode
- Week1
- Week2
  - Day 1
  - Day 2
    - person_class.py

person_class.py

```
1  class Person(object):
2      def __init__(self):
3          
```

Now, inside of that class, we can create our methods (functions) and properties (data).

First, lets look at the '__init__' function.

PROJECTS

0 0 Combo: 1 Ln 3, Col 9 Space

person_class.py — DigitalCrafts

EXPLORER

DIGITALCRAFTS

- .vscode
- Week1
- Week2
  - Day 1
  - Day 2
    - person_class.py

person_class.py

```
1  class Person(object):
2      def __init__(self):
3          
```

The '__init__' class will be called as soon as a new instance of that class is created (sometimes called construction, or initialization).

We can use this function to set the default state of our class

PROJECTS

0 0 Combo: 1 Ln 3, Col 9 Space

person_class.py — DigitalCrafts

EXPLORER

DIGITALCRAFTS
- .vscode
- Week1
- Week2
  - Day 1
  - Day 2
    - person_class.py

person_class.py

```
1  class Person(object):
2      def __init__(self):
3          self.name = 'Janice'
4
5
```

PROJECTS

0 0 Combo: 1 Ln 5, Col 5 Space

In this case, we know we want our Person class to have a name, so we set it in the constructor.

person_class.py — DigitalCrafts

EXPLORER

DIGITALCRAFTS

- .vscode
- Week1
- Week2
  - Day 1
  - Day 2
    - person_class.py

person_class.py

```
1  class Person(object):
2      def __init__(self):
3          self.name = 'Janice'
4
5      def greet(self):
6          print 'Hello, %s! Nice to meet you!' % (self.name)
7
```

Now we can create a method that can be called off that object.

Let's make 'greet'

PROJECTS

0 0 Combo: 3 Ln 7, Col 1 (1 selected) Space

person_class.py — DigitalCrafts

EXPLORER

person_class.py

```
class Person(object):
```

1. python (Python)

python (Python) ⌘1

```
~/Projects/DigitalCrafts/Week2/Day 2 $ python                    [23:41:10]
Python 2.7.10 (default, Jul 15 2017, 17:16:57)
[GCC 4.2.1 Compatible Apple LLVM 9.0.0 (clang-900.0.31)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Now let's test that in the python console

PROJECTS

0 0 Combo: 3 Ln 1, Col 22 Space

person_class.py — DigitalCrafts

EXPLORER

person_class.py

1. python (Python)

python (Python) ⌘1

```
~/Projects/DigitalCrafts/Week2/Day 2 $ python                    [23:42:09]
Python 2.7.10 (default, Jul 15 2017, 17:16:57)
[GCC 4.2.1 Compatible Apple LLVM 9.0.0 (clang-900.0.31)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from person_class import Person
>>>
```

First we import our class

PROJECTS

0 0 Combo: 3 Ln 1, Col 22 Space

person_class.py — DigitalCrafts

EXPLORER

person_class.py

```
class Person(object):
```

1. python (Python)

python (Python) ⌘1

```
~/Projects/DigitalCrafts/Week2/Day 2 $ python                    [23:42:09]
Python 2.7.10 (default, Jul 15 2017, 17:16:57)
[GCC 4.2.1 Compatible Apple LLVM 9.0.0 (clang-900.0.31)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from person_class import Person
>>> janice = Person()
>>>
```

PROJECTS

0 0 Combo: 3 Ln 1, Col 22 Space

Then we make a new instance of that class

person_class.py — DigitalCrafts

EXPLORER

person_class.py

1. python (Python)

python (Python) ⌘1

```
~/Projects/DigitalCrafts/Week2/Day 2 $ python                    [23:42:09]
Python 2.7.10 (default, Jul 15 2017, 17:16:57)
[GCC 4.2.1 Compatible Apple LLVM 9.0.0 (clang-900.0.31)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from person_class import Person
>>> janice = Person()
>>> print janice.name
Janice
>>>
```

Now we can print the name directly

PROJECTS

0 0 Combo: 3 Ln 7, Col 1 Space

person_class.py — DigitalCrafts

EXPLORER

person_class.py

1. python (Python)

python (Python) ⌘1

```
~/Projects/DigitalCrafts/Week2/Day 2 $ python                    [23:42:09]
Python 2.7.10 (default, Jul 15 2017, 17:16:57)
[GCC 4.2.1 Compatible Apple LLVM 9.0.0 (clang-900.0.31)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from person_class import Person
>>> janice = Person()
>>> print janice.name
Janice
>>> janice.greet()
Hello, Janice! Nice to meet you!
>>>
```

Or we can call the 'greet' method

PROJECTS

0 0 Combo: 3 Ln 7, Col 1 Space

person_class.py — DigitalCrafts

EXPLORER

DIGITALCRAFTS
- .vscode
- Week1
- Week2
  - Day 1
  - Day 2
    - person_class.py
    - person_class.pyc

PROJECTS

person_class.py

```
1  class Person(object):
2      def __init__(self, name):
3          self.name = name
4
5      def greet(self):
6          print 'Hello, %s! Nice to meet you!' % (self.name)
7
```

Now, not every person is called Janice, so let's make that accept a new variable on initialization

0 0 Combo: 4 2 selections (4 characters selected) Space

person_class.py — DigitalCrafts

EXPLORER

person_class.py

1. python (Python)

python (Python) ⌘1

```
~/Projects/DigitalCrafts/Week2/Day 2 $ python                    [23:47:52]
Python 2.7.10 (default, Jul 15 2017, 17:16:57)
[GCC 4.2.1 Compatible Apple LLVM 9.0.0 (clang-900.0.31)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from person_class import Person
>>> bob = Person('Bob')
>>> karen = Person('Karen')
>>> janice = Person('Janice')
>>>
```

PROJECTS

0 0 Combo: 1 Ln 4, Col 1 Space

Now when we define a person, we can give them a unique name.

As you can see, I've created three new objects of the Person class with different names

person_class.py — DigitalCrafts

EXPLORER

person_class.py

1. python (Python)

python (Python) ⌘1

```
~/Projects/DigitalCrafts/Week2/Day 2 $ python                    [23:47:52]
Python 2.7.10 (default, Jul 15 2017, 17:16:57)
[GCC 4.2.1 Compatible Apple LLVM 9.0.0 (clang-900.0.31)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from person_class import Person
>>> bob = Person('Bob')
>>> karen = Person('Karen')
>>> janice = Person('Janice')
>>> bob.greet()
Hello, Bob! Nice to meet you!
>>> karen.greet()
Hello, Karen! Nice to meet you!
>>> janice.greet()
Hello, Janice! Nice to meet you!
>>>
```

So when I call the greet method, they perform the same action, with slightly different results

PROJECTS

0 0 Combo: 1 Ln 4, Col 1 Space

person_class.py — DigitalCrafts

EXPLORER

DIGITALCRAFTS

- .vscode
- Week1
- Week2
  - Day 1
  - Day 2
    - person_class.py
    - person_class.pyc

person_class.py

```
1  class Person(object):
2      def __init__(self, name):
3          self.name = name
4
5      def greet(self):
6          print 'Hello, %s! Nice to meet you!' % (self.name)
7
```

PROJECTS

0 0 Combo: 1 Ln 4, Col 1 Space

Let's add another method to introduce different people to each other, and then set their friends

person_class.py — DigitalCrafts

EXPLORER

DIGITALCRAFTS
- .vscode
- Week1
- Week2
  - Day 1
  - Day 2
    - person_class.py
    - person_class.pyc

PROJECTS

person_class.py

```
1  class Person(object):
2      def __init__(self, name):
3          self.name = name
4
5      def greet(self):
6          print 'Hello, %s! Nice to meet you!' % (self.name)
7
8      def introduce(self, new_friend):
9
10
```

It's going to take the 'self' argument as all class methods do, plus a second 'new_friend' argument

0 0 Combo: 1 Ln 8, Col 37 Space

person_class.py — DigitalCrafts

EXPLORER

DIGITALCRAFTS

- .vscode
- Week1
- Week2
  - Day 1
  - Day 2
    - person_class.py
    - person_class.pyc

person_class.py

```
1  class Person(object):
2      def __init__(self, name):
3          self.name = name
4          self.friends = []
5
6      def greet(self):
7          print 'Hello, %s! Nice to meet you!' % (self.name)
8
9      def introduce(self, new_friend):
10
11
```

Before we add new friends, we need to set the default list of friends to be an empty list

PROJECTS

0 0 Combo: 32 Ln 4, Col 21 Space

```python
class Person(object):
    def __init__(self, name):
        self.name = name
        self.friends = []

    def greet(self):
        print 'Hello, %s! Nice to meet you!' % (self.name)

    def introduce(self, new_friend):
        self.friends.append(new_friend.name)

```

Now we can append the introduced friend to the Person's friends list

person_class.py — DigitalCrafts

EXPLORER

DIGITALCRAFTS
- .vscode
- Week1
- Week2
  - Day 1
  - Day 2
    - person_class.py
    - person_class.pyc

person_class.py

```
1  class Person(object):
2      def __init__(self, name):
3          self.name = name
4          self.friends = []
5
6      def greet(self):
7          print 'Hello, %s! Nice to meet you!' % (self.name)
8
9      def introduce(self, new_friend):
10         self.friends.append(new_friend.name)
11         print '%s made friends with %s' % (self.name, new_friend.name)
12
```

PROJECTS

0 0 Combo: 1 Ln 11, Col 71 Space

Let's add some output to show that it's working

person_class.py — DigitalCrafts

EXPLORER

person_class.py

1. python (Python)

python (Python) ⌘1

```
~/Projects/DigitalCrafts/Week2/Day 2 $ python                    [0:02:12]
Python 2.7.10 (default, Jul 15 2017, 17:16:57)
[GCC 4.2.1 Compatible Apple LLVM 9.0.0 (clang-900.0.31)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from person_class import Person
>>> bob = Person('Bob')
>>> karen = Person('Karen')
>>>
```

riend.name)

PROJECTS

0 0 Combo: 1 Ln 11, Col 71 Space

And then let's open the console again, import our class, define some Persons, and introduce them

person_class.py — DigitalCrafts

EXPLORER

person_class.py

1. python (Python)

python (Python) ⌘1

```
~/Projects/DigitalCrafts/Week2/Day 2 $ python                    [0:02:12]
Python 2.7.10 (default, Jul 15 2017, 17:16:57)
[GCC 4.2.1 Compatible Apple LLVM 9.0.0 (clang-900.0.31)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from person_class import Person
>>> bob = Person('Bob')
>>> karen = Person('Karen')
>>> bob.introduce(karen)
Bob made friends with Karen
>>>
```

```
riend.name)
```

PROJECTS

0 0 Combo: 1 Ln 11, Col 71 Space

And then let's open the console again, import our class, define some Persons, and introduce them