Classes and Exceptions

April 24, 2016

- fast reflexes
- blind
- pointy head things
- baton
- sonar

Daredevil

Captain America

- fast reflexes
- ageless
- pointy head things
- super strength
- shield

Use classes to group together features(adjectives) and actions(verbs) of an object(nouns).

Classes describe how objects look and behave.

```
class Daredevil:
    def __init__(self):
        self.name = ‘daredevil’
        self.pointy_hat = True
        self.baton = True

    def attack(self):
        print('i\'m coming')

    def defend(self):
        print('stand back')

    def use_senses(self):
        print('ping ping ping')
```

Daredevil

```
class CaptainAmerica:
    def __init__(self):
        self.name = ‘cap’
        self.pointy_hat = True
        self.shield = True

    def attack(self):
        print('i\'m coming')

    def defend(self):
        print('stand back')

    def lift_heavy_stuff(self):
        print('uuurrggg')
```

Captain America

Using a class you can create objects. You can interact with objects in your code.

You can cause objects to perform actions and can interact with an objects properties.

```
dd = Daredevil()

print(dd.pointy_hat)
dd.defend()
dd.use_senses()
```

Daredevil

Captain America

```
cap = CaptainAmerica()

print(cap.pointy_hat)
cap.attack()
cap.lift_heavy_stuff()
```

```
dd = Daredevil()

print(dd.pointy_hat)
dd.defend()
dd.use_senses()

True
stand back
ping ping ping
```

Daredevil

Captain America

```
cap = CaptainAmerica()

print(cap.pointy_hat)
cap.attack()
cap.lift_heavy_stuff()

True
i'm coming
uuurrggg
```

Sometimes we may ask an object to do more than it is capable of.

When this happens we receive an exception.

```
dd = Daredevil()
cap = CaptainAmerica()


def detect_villan_using_radar(hero):
    hero.use_senses()


detect_villan_using_radar(dd)
detect_villan_using_radar(cap)
```

```
ping ping ping

AttributeError: 'CaptainAmerica'
object has no attribute 'use_senses'
```

Daredevil

Captain America

Using

try, **except** blocks

We can deal with these exceptions and let our program continue.

```
dd = Daredevil()
cap = CaptainAmerica()


def detect_villan_using_radar(hero):
    try:
        hero.use_senses()
    except AttributeError:
        print('%s, doesn\'t …’ % hero.name)


detect_villan_using_radar(dd)
detect_villan_using_radar(cap)
```

```
ping ping ping

cap, doesn't have this power
```

Daredevil

Captain America

```
BaseException
 +-- SystemExit
 +-- KeyboardInterrupt
 +-- GeneratorExit
 +-- Exception
      +-- StopIteration
      +-- StandardError
      |    +-- BufferError
      |    +-- ArithmeticError
      |    |    +-- FloatingPointError
      |    |    +-- OverflowError
      |    |    +-- ZeroDivisionError
      |    +-- AssertionError
      |    +-- AttributeError
      |    +-- EnvironmentError
      |    |    +-- IOError
      |    |    +-- OSError
      |    |         +-- WindowsError (Windows)
      |    |         +-- VMSError (VMS)
      |    +-- EOFError
      |    +-- ImportError
      |    +-- LookupError
      |    |    +-- IndexError
      |    |    +-- KeyError
      |    +-- MemoryError
      |    +-- NameError
      |    |    +-- UnboundLocalError
      |    +-- ReferenceError
      |    +-- RuntimeError
      |    |    +-- NotImplementedError
      |    +-- SyntaxError
      |    |    +-- IndentationError
      |    |         +-- TabError
      |    +-- SystemError
      |    +-- TypeError
      |    +-- ValueError
      |         +-- UnicodeError
      |              +-- UnicodeDecodeError
      |              +-- UnicodeEncodeError
      |              +-- UnicodeTranslateError
      +-- Warning
           +-- DeprecationWarning
           +-- PendingDeprecationWarning
           +-- RuntimeWarning
           +-- SyntaxWarning
           +-- UserWarning
           +-- FutureWarning
          +-- ImportWarning
          +-- UnicodeWarning
          +-- BytesWarning
```

```
try:
    x = 4/0
except:
    print('exception happened')

exception happened
```

```
try:
    x = 4/0
except ZeroDivisionError:
    print('wat? divide by zero')
except Exception:
    print('some other exception happened')

wat? divide by zero
```

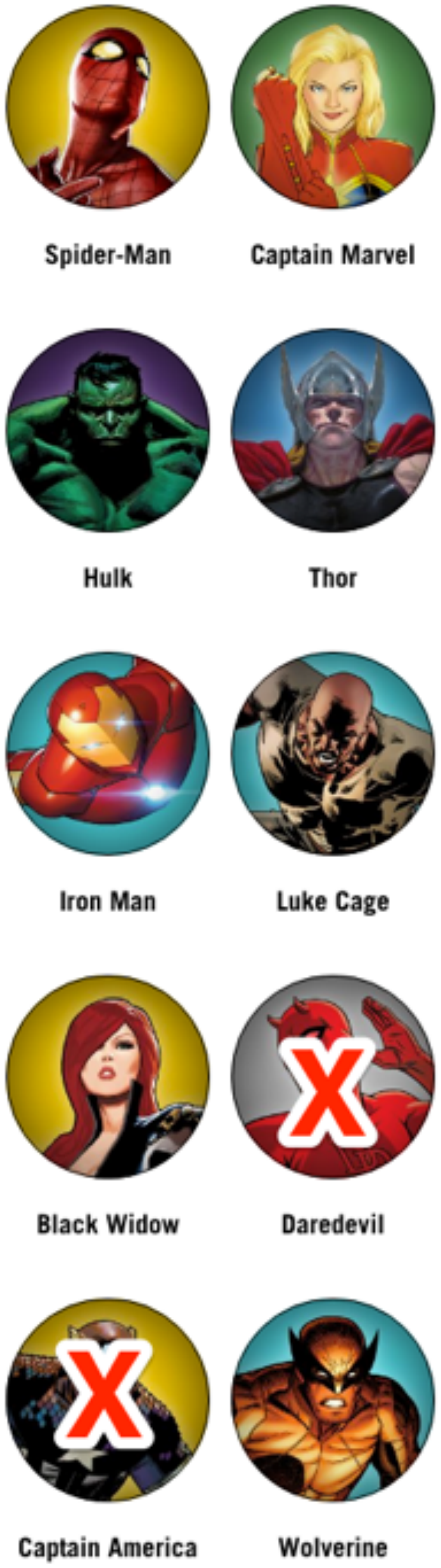
```
try:
    x.do_something()
except ZeroDivisionError:
    print('wat? divide by zero')
except Exception:
    print('some other exception happened')

some other exception happened
```

```
try:
    x.do_something()
except Exception as e:
    print('some exception happened')
    print(str(e))

some exception happened
name 'x' is not defined
```

Exercise



1. Choose a hero marvel or dc.

2. List your hero’s powers and how it looks.

   <u>looks</u> <u>powers</u>

3. Define a class that matches your hero’s description.
   1. descriptive words self.hair_color = ‘black’
   2. translate powers to verbs and functions def use_ray_beam()

4. Write code to create an object from your new class.

5. Access attributes of your class.

6. Try to access an attribute of your class that doesn’t exist.

7. Use try except to trap the exception and to print a message

```
class CaptainAmerica:
    def __init__(self):
        self.name = ‘cap’
        self.pointy_hat = True
        self.shield = True

    def attack(self):
        print('i\'m coming')

    def defend(self):
        print('stand back')

    def lift_heavy_stuff(self):
        print('uuurrggg')
```

Captain America

```
cap = CaptainAmerica()

print(cap.pointy_hat)
cap.attack()
cap.lift_heavy_stuff()

True
i'm coming
uuurrggg
```

```
try:
    cap.use_senses()
except AttributeError:
    print('%s, doesn\'t …’ % cap.name)

cap, doesn't have this power
```