Introduction to Classes

CSE 160 Data Programming

Questions?

Favorite pokémon? **Icebreaker**

Sli.do **#cse160**

Announcements

- Programming Practice 5 due Sunday, August 2 at 11:59 PM
- Homework Assignment 4 due Monday, August 3 at 11:59 PM
- Written Check-In 6 due tonight, July 31 at 11:59 PM
- Resubmission Cycle 3 opens today

Today's Roadmap

1. Organizing Code
2. Objects (Real and Pythonic)
3. Classes in Python
4. Examples

Jupyter Hub

So Far in CSE 160…

How can we (further) organize all of this?

1. Variables, Expressions, Data Types
2. Loops
3. Conditionals
4. Functions
5. Data Structures

Objects

(Real-world) objects have attributes and can (sometimes) perform actions

Attributes

- Color
- Cost
- Casabrews Model
- Type of Coffee Grounds
- Brew Time
- Shot Type

Actions

- Turn on/off
- Brew 1-shot (Input: coffee grounds)
- Brew 2-shots (Input: coffee grounds)
- Steam (Input: milk)

Python Objects

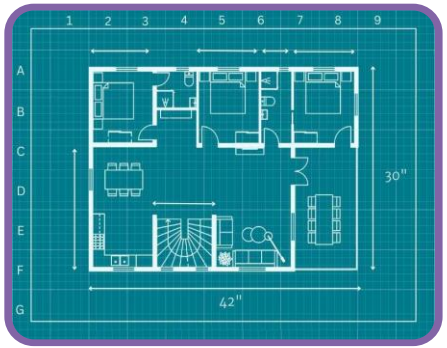
Python objects bundle data and functionality together.

What if built-in data structures don't suffice?
→ Let's create our own!

Classes

Classes** provide the blueprint for creating an **object

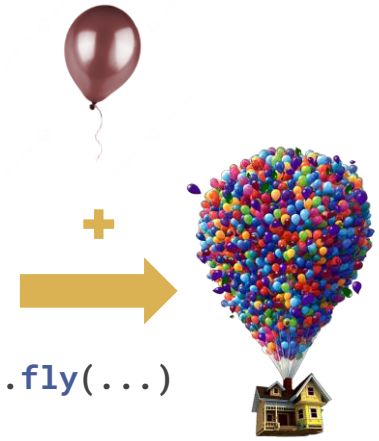
Class Blueprint



- `number_of_rooms`
- `sqr_footage`
- `color`

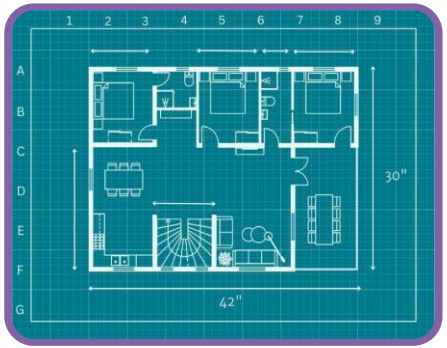
Specific Object

- `number_of_rooms = 5`
- `sqr_footage = 2800`
- `color = "Multicolored"`



Classes

Class Blueprint



- `number_of_rooms`
- `sqr_footage`
- `color`

`.fly(...)`

Classes

Classes** provide the blueprint for creating an **object

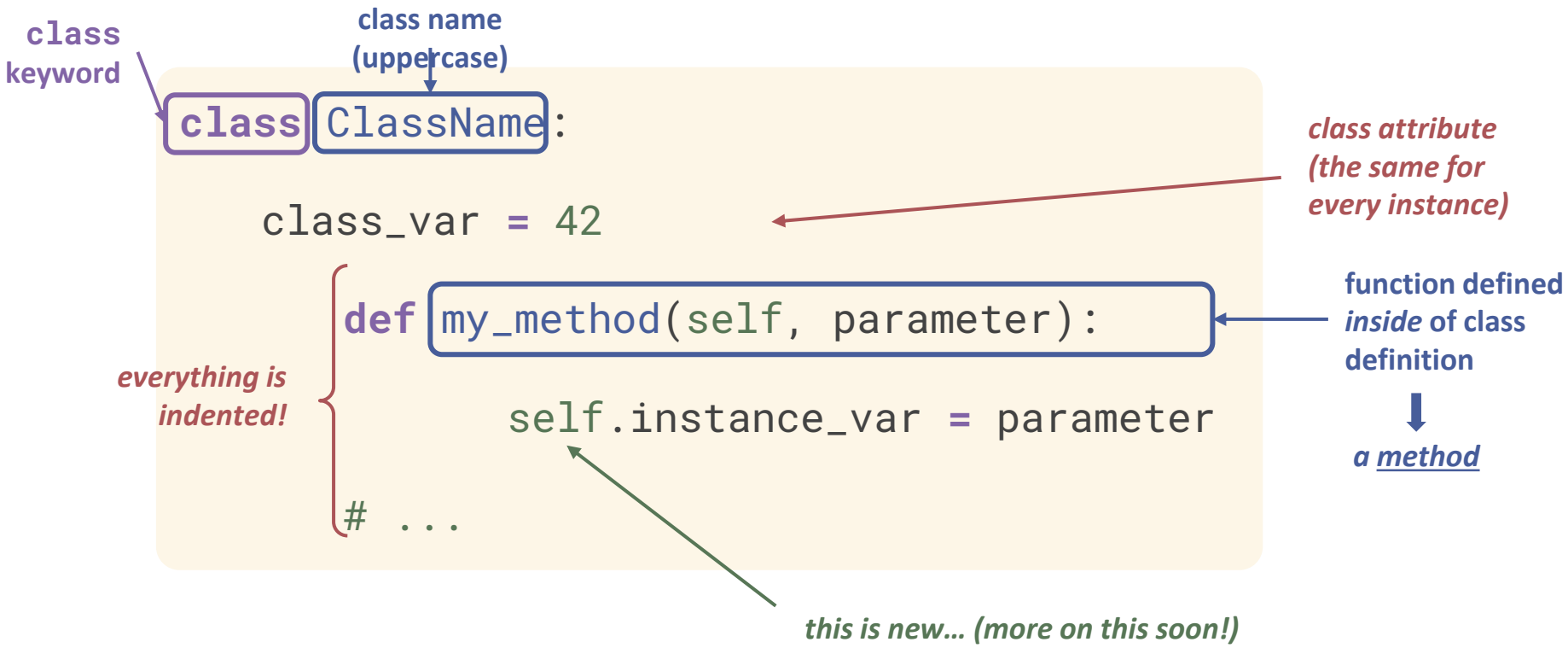
Class Blueprint

- `color`
- `model`
- `brew_time`

Specific Object

- `color = "Black"`
- `model = "CM5418"`
- `brew_time = 30`

Classes in Python



Creating an Object

special method that is called when the object is created

defining the class (template)

```
class ClassName:

    def __init__(self, parameter):

        self.parameter = parameter
```

`self` is a variable that refers to the object *itself*

not the same variable!

An *instance variable* is accessible only to the object (instance)

creating an *instance* of the class (i.e. an object)

```
my_object = ClassName(18)
```

Creating an Object

```
class House:

        def __init__(self, sqr_foot, num_rooms, color):

                self.sqr_foot = sqr_foot
                self.num_rooms = num_rooms
                self.color = color
```

```
carls_house = House(2800, 5, “Multicolored”)
```

Questions?

Live and on Sli.do

Sli.do #cse160

Example: Pokemon

```
class Pokemon:

    hp = 20

        def __init__(self, name, kind):

                self.name = name
                self.type = kind
```


```
psyduck = Pokemon("Psyduck",
                  "Water")
```

Think Pair Share

Consider these two ***instances*** of the Pokemon class:

```
psyduck = Pokemon("Psyduck", "Water")

wooper = Pokemon("Wooper", "Ground")
```

Sli.do #cse160

What is printed when these two lines of code are run?

```
psyduck.evolve()

print(wooper.hp)
```

A 20

B 40

C 60

D 80

E Error

Using an Object

creating an *instance* of the class (i.e. an object)

```
my_object = ClassName(18)
```

```
class ClassName:

    class_var = 160

    def __init__(self, parameter):

        self.parameter = parameter
```

`my_object.parameter` → 18

`my_object.class_var` → 160

`ClassName.class_var` → 160

Example: Universities

What attributes and functionality does a university have?

Think Pair Share

Create a class called **`University`** with the following **attributes** and **methods**:

- **`name`**, a string
- **`mascot`**, a string
- **`enrollment`**, an int
- **`depts`**, a list of strings

- **`welcome()`**, that prints:
  "[mascot] welcomes you to [name]"
- **`add_dept(new_dept)`**, that adds `new_dept` to the `depts` list

University of Washington

Questions?

Live and on Sli.do

Sli.do #cse160

Anatomy of a Class