



FULL SAIL
UNIVERSITY

Access Modifiers

Limiting access to parts of your code

Design Patterns for Web Programming
Web Design & Development Bachelor of Science Degree





Attributes Review

- Attributes of an object are values that describe the characteristics of that object. They are traits
- Special kind of variables - Designed to store data
- Nouns
- Example:
 - rover = Dog()
 - rover.breed = "Sheltie"
- Example of attribute outside of the class it was created:
 - instance.attribute = value
- Example of attribute within its own class:
 - self.attribute = value





Access Modifiers

Limiting access to parts of a class

- Public
 - accessible by any other object
 - **self.attribute**
 - **self.method()** (function call, not declaration here)
- Protected (We will start using these when we talk about Inheritance)
 - accessible by sub-classes
 - **self._attribute**
 - **self._method()**
- Private
 - accessible ONLY within the class it is defined
 - **self.__attribute** (two underscores!!)
 - **self.__method()**





How to access them?

Limiting access to parts of a class

- Private access modifiers mean they are **ONLY ACCESSIBLE WITHIN THE CLASS THEY ARE CREATED!**
- Getters and Setters allow access
 - Getters allow read access
 - Setters allow write access
 - More on these later!
- What if you need to access them?
 - If you need **complete access.. private or protected probably isn't appropriate.**





When to limit access?

Limiting access to parts of a class

- Like security. Limit everything and open access ONLY when necessary.
- Getters and Setters will help grant access to private and protected attributes.
- If you KNOW for sure, you will need read and write access to a variable or function, make them public.
- For example: Data Objects usually contain many public class members.





Access Modifiers in Action

Attributes & Methods

```
class Translator(object):
    def __init__(self):
        self.__greeting = "Hello"
        self.language = "French"
        self.changeGreetingToFrench()

    def changeGreetingToFrench(self):
        self.__greeting = "Bonjour"
```





Access Modifiers in Action

Attributes & Methods

```
class Translator(object):
    def __init__(self):
        self.__greeting = "Hello"
        self.language = "French"
        self.changeGreetingToFrench()

    def changeGreetingToFrench(self):
        self.__greeting = "Bonjour"
```

In another class we have the following:

```
t = Translator() #create instance
```





Access Modifiers in Action

Attributes & Methods

```
class Translator(object):
    def __init__(self):
        self.__greeting = "Hello"
        self.language = "French"
        self.changeGreetingToFrench()

    def changeGreetingToFrench(self):
        self.__greeting = "Bonjour"
```

In another class we have the following:

```
t = Translator()      #create instance
t.__greeting = "Hello" #WONT work.. private!
```





Access Modifiers in Action

Attributes & Methods

```
class Translator(object):
    def __init__(self):
        self.__greeting = "Hello"
        self.language = "French"
        self.changeGreetingToFrench()

    def changeGreetingToFrench(self):
        self.__greeting = "Bonjour"
```

In another class we have the following:

```
t = Translator()      #create instance
t.__greeting = "Hello" #WONT work.. private!
t.greeting = "Hello" #WONT work.. creates new and
completely separate attribute.
```





Access Modifiers in Action

Attributes & Methods

```
class Translator(object):
    def __init__(self):
        self.__greeting = "Hello"
        self.language = "French"
        self.changeGreetingToFrench()

    def changeGreetingToFrench(self):
        self.__greeting = "Bonjour"
```

In another class we have the following:

```
t = Translator()      #create instance
t.__greeting = "Hello" #WONT work.. private!
t.greeting = "Hello" #WONT work.. creates new and
                     completely separate attribute.
t.language = "Turkish" #public - allows access
t.changeGreetingToFrench()
```





Review

- You can use access modifiers to limit access to class members like attributes and methods.
- You can still grant access with other means
- Limiting access is good.

