



FULL SAIL
UNIVERSITY

Encapsulation

Keep private things private...

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





FULL SAIL
UNIVERSITY

Encapsulation

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





FULL SAIL
UNIVERSITY

Encapsulation

Hiding things from
your future self.

Design Patterns for Web Programming

Web Design & Development Bachelor of Science Degree





FULL SAIL
UNIVERSITY

Why bother?

So why do we have to do this again?

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





FULL SAIL
UNIVERSITY

Why bother?

So why do we have to do this again?

- Why not just declare a variable as public?





Why bother?

So why do we have to do this again?

- Why not just declare a variable as public?
- Protect information from others or even yourself





Why bother?

So why do we have to do this again?

- Why not just declare a variable as public?
- Protect information from others or even yourself
- Bank Metaphor





FULL SAIL
UNIVERSITY

Properties in Python

Opening and restricting access to attributes

Design Patterns for Web Programming

Web Design & Development Bachelor of Science Degree





Properties in Python

Opening and restricting access to attributes

- ➊ Properties are special attribute





Properties in Python

Opening and restricting access to attributes

- Properties are special attribute
- Created when we need to give special access to a particular attribute





Properties in Python

Opening and restricting access to attributes

- Properties are special attribute
- Created when we need to give special access to a particular attribute
- Created using Decorators





Properties in Python

Opening and restricting access to attributes

- Properties are special attribute
- Created when we need to give special access to a particular attribute
- Created using Decorators
- `@decorator`





Properties in Python

Opening and restricting access to attributes

- Properties are special attribute
- Created when we need to give special access to a particular attribute
- Created using Decorators
- `@decorator`
- This means this function will be treated like a variable!!





FULL SAIL
UNIVERSITY

Properties in Python

Decorate the code

Design Patterns for Web Programming

Web Design & Development Bachelor of Science Degree





FULL SAIL
UNIVERSITY

Properties in Python

Decorate the code

- ◉ Decorators for properties





Properties in Python

Decorate the code

- ◉ Decorators for properties
 - `@property`





Properties in Python

Decorate the code

① Decorators for properties

- `@property`
- `@x.setter`





Properties in Python

Decorate the code

① Decorators for properties

- `@property`
- `@x.setter`
- `@x.deleter`





FULL SAIL
UNIVERSITY

Getter and Setters

Helper functions

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





FULL SAIL
UNIVERSITY

Getter and Setters

Helper functions

○ Getters

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





Getter and Setters

Helper functions

○ Getters

- Special function designed to grab data





Getter and Setters

Helper functions

○ Getters

- Special function designed to grab data
- READ ONLY





Getter and Setters

Helper functions

○ Getters

- Special function designed to grab data
- READ ONLY

○ The Setter





Getter and Setters

Helper functions

○ Getters

- Special function designed to grab data
- READ ONLY

○ The Setter

- Special function designed to change data





Getter and Setters

Helper functions

○ Getters

- Special function designed to grab data
- READ ONLY

○ The Setter

- Special function designed to change data
- Write ONLY





FULL SAIL
UNIVERSITY

Benefit #1

Getters and Setters

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





Benefit #1

Getters and Setters

- You can validate information before it changes important attributes.





Benefit #1

Getters and Setters

- You can validate information before it changes important attributes.
- Ex:





Benefit #1

Getters and Setters

- You can validate information before it changes important attributes.
- Ex:
 - Setter can check the value





Benefit #1

Getters and Setters

- You can validate information before it changes important attributes.
- Ex:
 - Setter can check the value
 - If the value is incorrect the setter can throw an error - or -





Benefit #1

Getters and Setters

- You can validate information before it changes important attributes.
- Ex:
 - Setter can check the value
 - If the value is incorrect the setter can throw an error - or -
 - the setter can correct the value





FULL SAIL
UNIVERSITY

Benefit #2

Getters and Setters

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





FULL SAIL
UNIVERSITY

Benefit #2

Getters and Setters

- You can set information to read-only or write-only access.





Benefit #2

Getters and Setters

- You can set information to read-only or write-only access.
- Good for information you don't want altered.





Benefit #2

Getters and Setters

- You can set information to read-only or write-only access.
- Good for information you don't want altered.
- Ex:





Benefit #2

Getters and Setters

- You can set information to read-only or write-only access.
- Good for information you don't want altered.
- Ex:
 - Getter & no Setter: Read Only





Benefit #2

Getters and Setters

- You can set information to read-only or write-only access.
- Good for information you don't want altered.
- Ex:
 - Getter & no Setter: Read Only
 - Setter & no Getter: Write Only





FULL SAIL
UNIVERSITY

Benefit #3

Getters and Setters

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





Benefit #3

Getters and Setters

- You can do more when a property/attribute is accessed.





Benefit #3

Getters and Setters

- You can do more when a property/attribute is accessed.
- Since getters and setters are functions so they can do more lines of code when called (which is more than you can do with a simple variable)





Benefit #3

Getters and Setters

- You can do more when a property/attribute is accessed.
- Since getters and setters are functions so they can do more lines of code when called (which is more than you can do with a simple variable)
- Ex:





Benefit #3

Getters and Setters

- You can do more when a property/attribute is accessed.
- Since getters and setters are functions so they can do more lines of code when called (which is more than you can do with a simple variable)
- Ex:
 - Setter is used to change information in a class





Benefit #3

Getters and Setters

- You can do more when a property/attribute is accessed.
- Since getters and setters are functions so they can do more lines of code when called (which is more than you can do with a simple variable)
- Ex:
 - Setter is used to change information in a class
 - Display is updated right away.





Bank metaphor

Keepin' it secure

○ Like bank tellers:

- Check we are who we say we are.
- Make sure there is enough \$\$
- They can do other stuff while you withdraw (like log the transaction, offer a better account, etc)





FULL SAIL
UNIVERSITY

Property Decorator

Python decorations

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





Property Decorator

Python decorations

```
class LanguageGreeting(object):
    def __init__(self):
        self.__greeting = "Hello"

    @property
    def greeting(self):
        return self.__greeting

    @greeting.setter
    def greeting(self, value):
        self.__greeting = value
```





FULL SAIL
UNIVERSITY

Using the property

Python decorations

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





Using the property

Python decorations

```
lang = LanguageGreeting()  
lang.greeting = "Oy!" #uses setter  
print lang.greeting #uses getter
```

```
class LanguageGreeting:  
    def __init__(self):  
        self.__greeting = "Hello"
```

```
@property  
def greeting(self):  
    return self.__greeting
```

```
@greeting.setter  
def greeting(self, value):  
    self.__greeting = value
```





FULL SAIL
UNIVERSITY

Encapsulation in Python

There's always a caveat

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





Encapsulation in Python

There's always a caveat

- Encapsulation is roughly equivalent to the lock on the door of a bathroom.





Encapsulation in Python

There's always a caveat

- Encapsulation is roughly equivalent to the lock on the door of a bathroom.
- It's not intended to keep anybody out if they really insist on entering.





Encapsulation in Python

There's always a caveat

- Encapsulation is roughly equivalent to the lock on the door of a bathroom.
- It's not intended to keep anybody out if they really insist on entering.
- It is intended as a courtesy to let people know that entering will lead mostly to:





Encapsulation in Python

There's always a caveat

- Encapsulation is roughly equivalent to the lock on the door of a bathroom.
- It's not intended to keep anybody out if they really insist on entering.
- It is intended as a courtesy to let people know that entering will lead mostly to:
 - embarrassment, and





Encapsulation in Python

There's always a caveat

- Encapsulation is roughly equivalent to the lock on the door of a bathroom.
- It's not intended to keep anybody out if they really insist on entering.
- It is intended as a courtesy to let people know that entering will lead mostly to:
 - embarrassment, and
 - a stinking mess.

