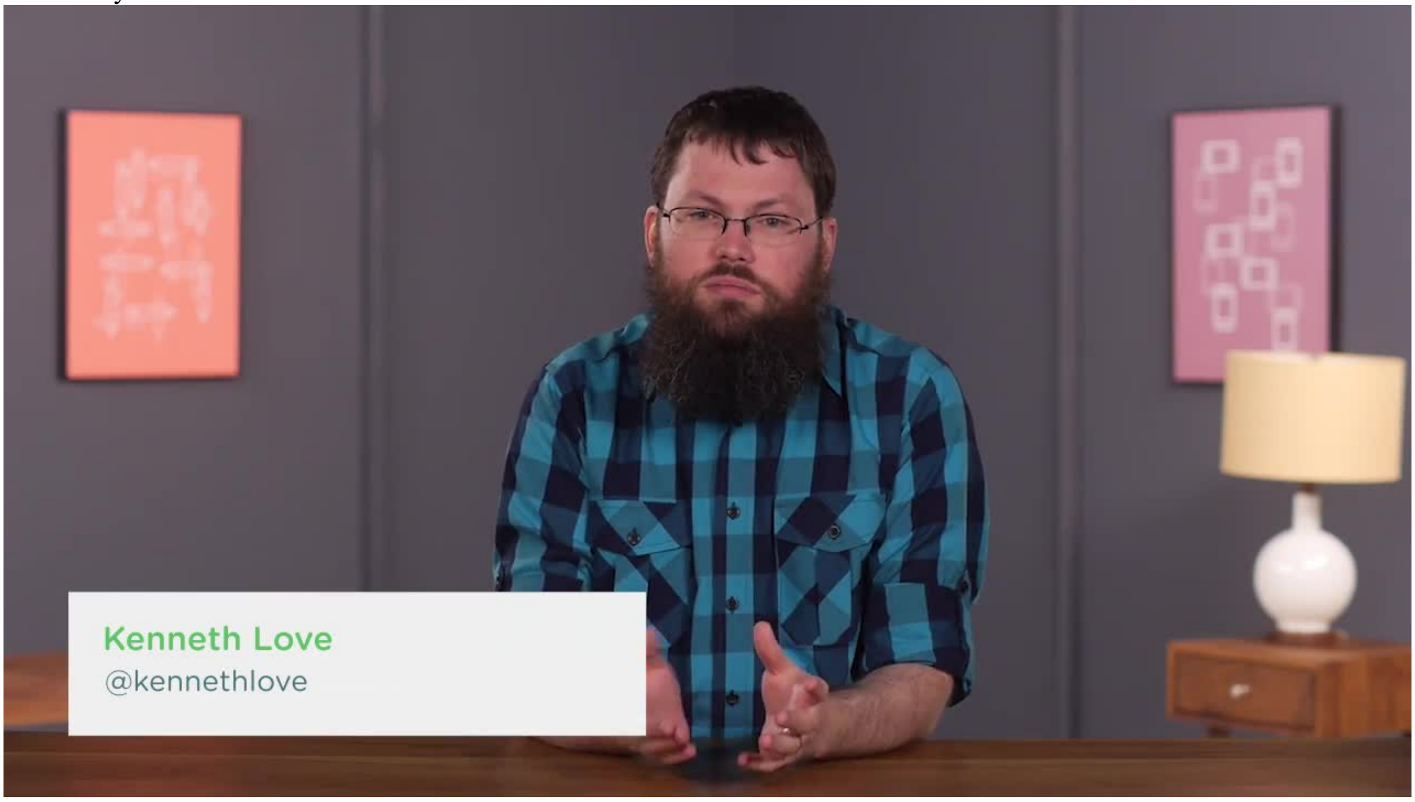
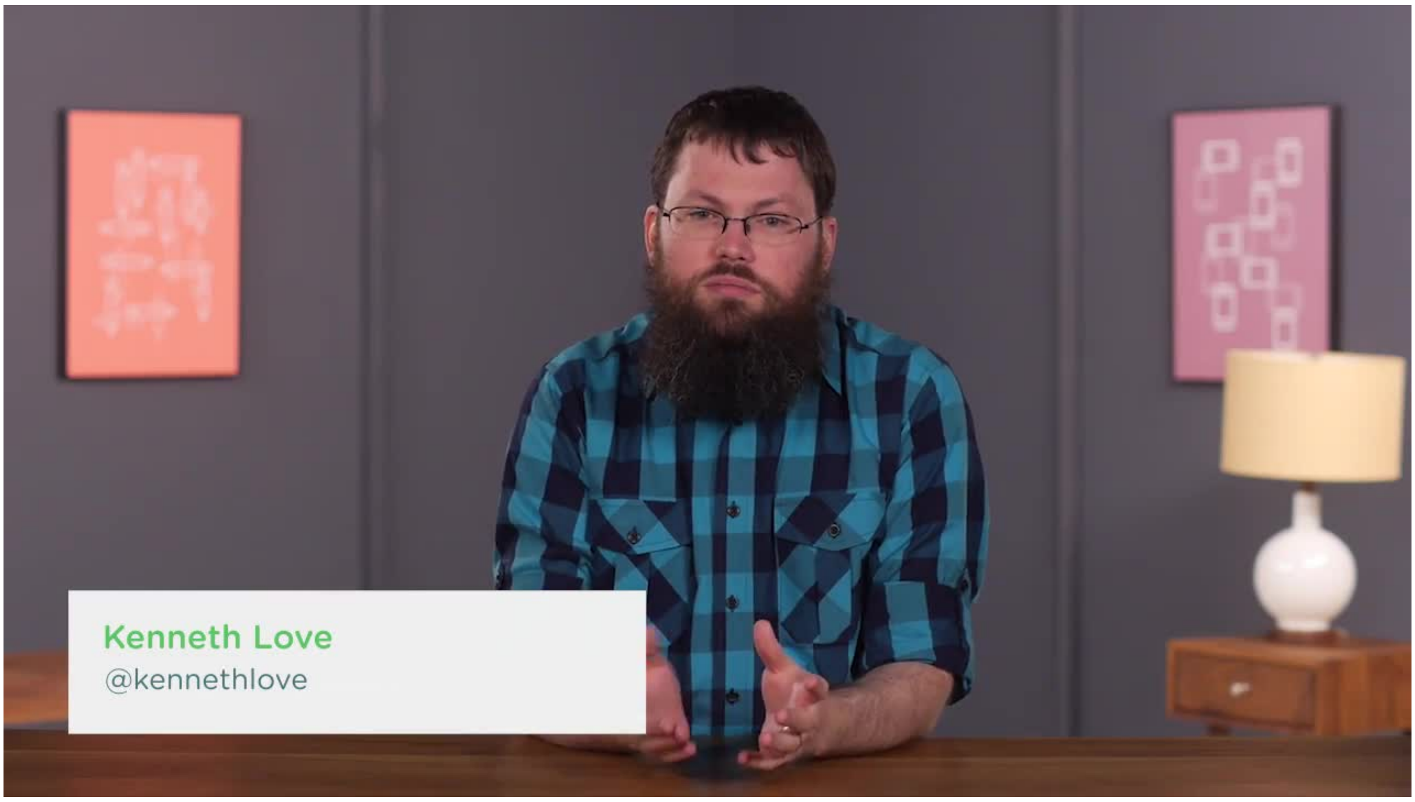
Python Decorators

You've already started watching **Python Decorators**

Resume Video Start Over

Video Player





00:00
00:00
00:00

1. 2x 2x
2. 1.75x 1.75x
3. 1.5x 1.5x
4. 1.25x 1.25x
5. 1.1x 1.1x
6. 1x 1x
7. 0.75x 0.75x
8. 0.5x 0.5x

- None
- [✓ English]

Use Up/Down Arrow keys to increase or decrease volume.

Python Decorators

23:09 with Kenneth Love

Sometimes you can just write a function or method or class and be done with it. But sometimes you find that you need to wrap your functions or methods or classes in some sort of new behavior. And since you're a good programmer, you don't want to change every function in the same way over and over again. That's not DRY or a good use of your time! Well, when that happens, you just might need a decorator.

- Teacher's Notes
- Questions?4
- Video Transcript
- Downloads

For more about logging: Write Better Python and the `logging` documentation.

A decorator cookbook.

A great blog post about using `functools.wraps`.

There is no functional difference, as far as Python itself or the interpreter is concerned, between applying a decorator directly or with the `@` symbol.

Examples

Nested function:

```
def outer():
    number = 5

    def inner():
        print(number)
    inner()

outer()  # prints 5
```

First-class functions

```
def apply(func, x, y):
    return func(x, y)

def add(a, b):
    return a + b

print(apply(add, 5, 5))  # prints 10
```

Closures

```
def close():
    x = 5

    def inner():
        print(x)
    return inner

closure = close()  # closure is actually a pointer to inner()
closure()  # prints 5
```

Decorator

```
def logme(func):
    import logging  # because we don't want to require users to import it
    logging.basicConfig(level=logging.DEBUG)

    def inner():
        logging.debug("Called {}".format(func.__name__)
        return func()
    return inner

@logme
def say_hello():
        print("Hello there!")

say_hello()  # logs the call and then prints "Hello there!"
```

Related Discussions

Have questions about this video? Start a discussion with the community and Treehouse staff.

Get Help