Andela

Writing Efficient Code with Python Decorator

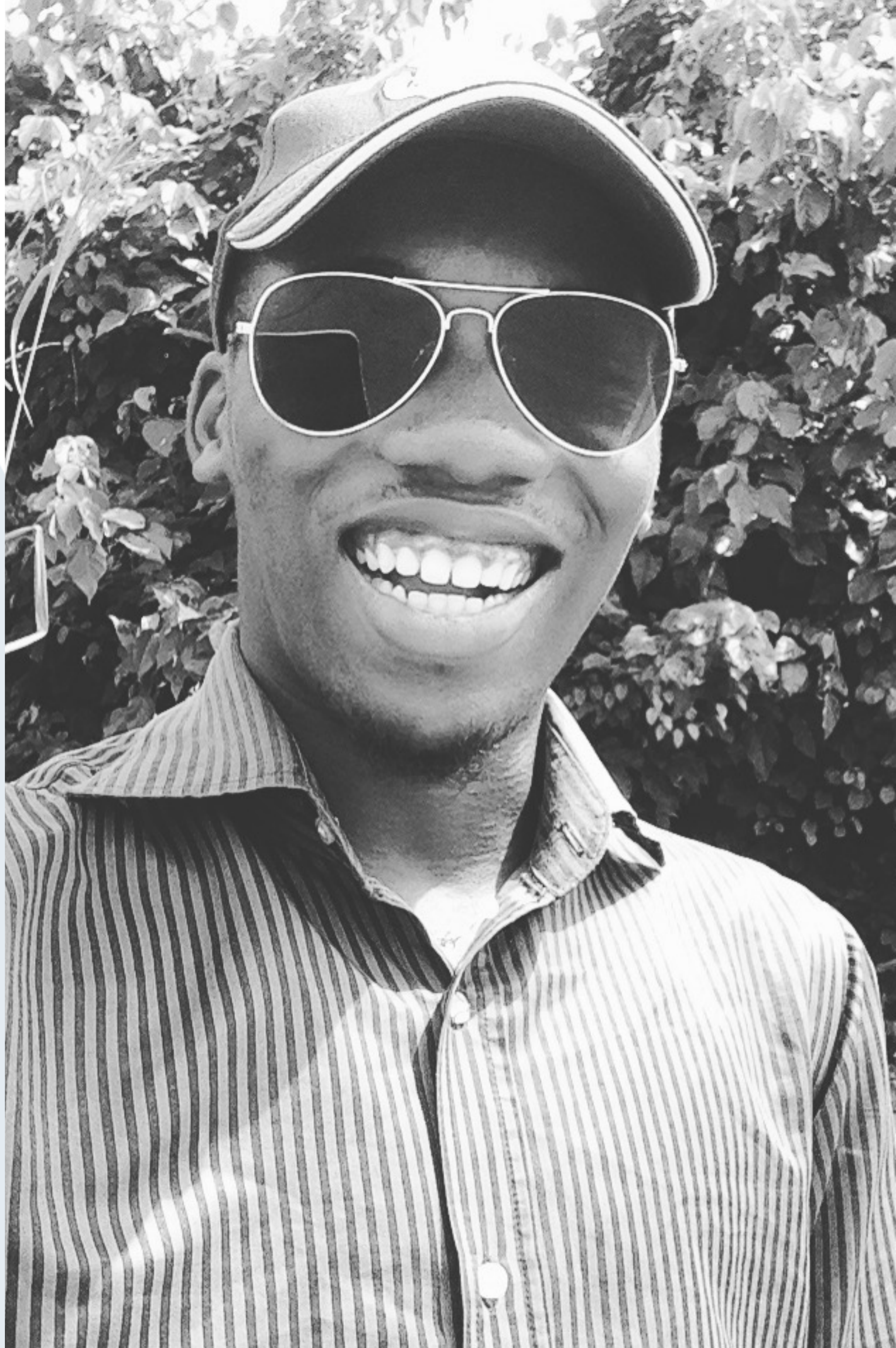
Presented by:
Bolaji Olajide,
Software Developer

September 16, 2017

About Me

- Software Developer @**Andela**
- Pythonista
- ReactJS (occasionally)
- Open-Source Contributor
- Writer (Backticks & Tildes)
- Proton (Positivity)

@cooProton BolajiOlajide @Bolaji___



ABOUT ANDELA

Our mission:

Andela was founded on a simple truth: Brilliance is evenly distributed.

Code Efficiency

CODE EFFICIENCY

SPEED

RELIABILITY

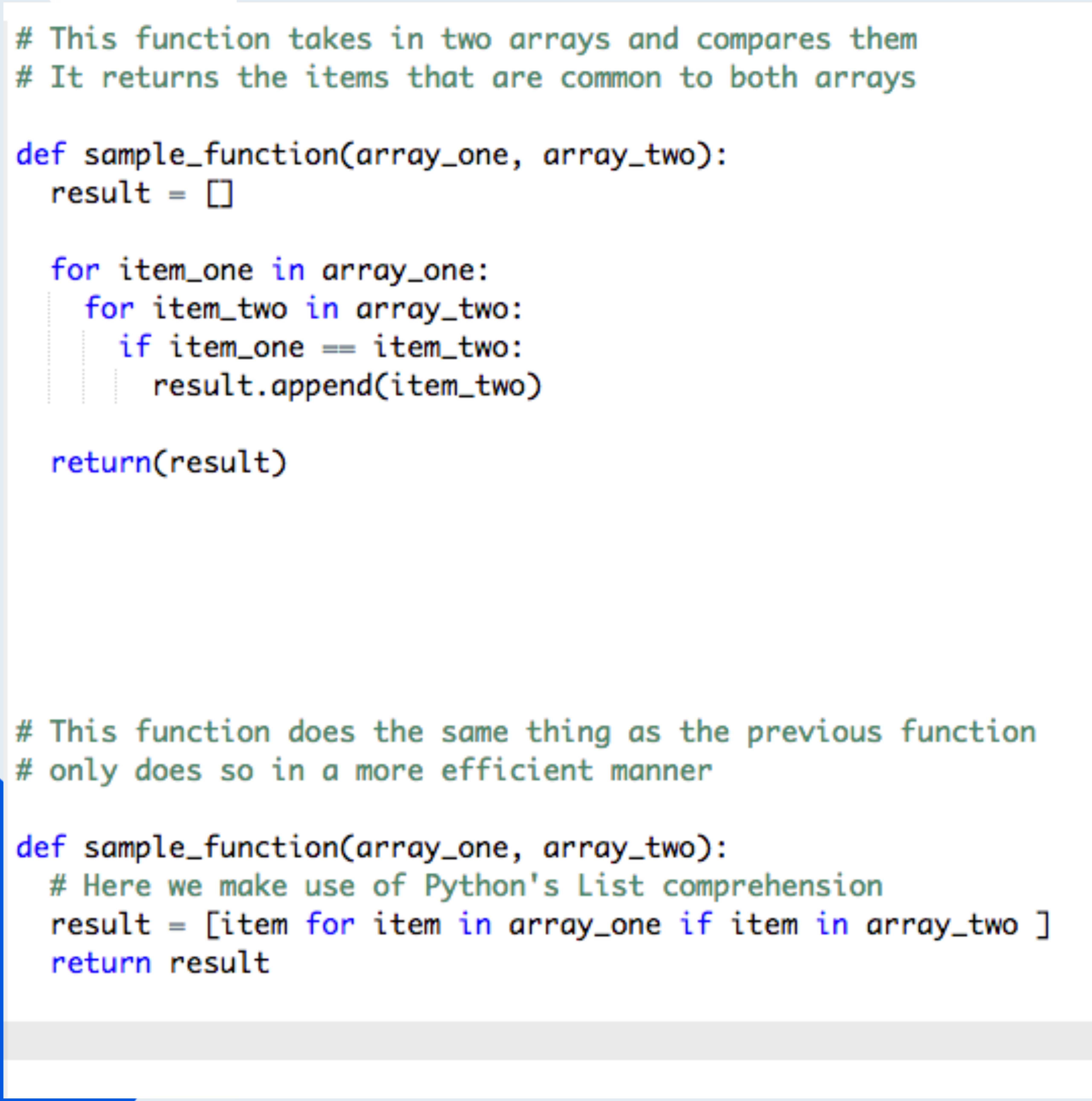
HIGH PERFORMANCE

Why Should My Code Be Efficient?

Weavon
Wig
Clipper
Headphone
Bag

Clipper
Headphone
iPod
Wrist-watch
Weavon

Why Should My Code Be Efficient?



```
# This function takes in two arrays and compares them
# It returns the items that are common to both arrays

def sample_function(array_one, array_two):
  result = []

  for item_one in array_one:
    for item_two in array_two:
      if item_one == item_two:
        result.append(item_two)

  return(result)




# This function does the same thing as the previous function
# only does so in a more efficient manner

def sample_function(array_one, array_two):
  # Here we make use of Python's List comprehension
  result = [item for item in array_one if item in array_two ]
  return result
```

Decorators

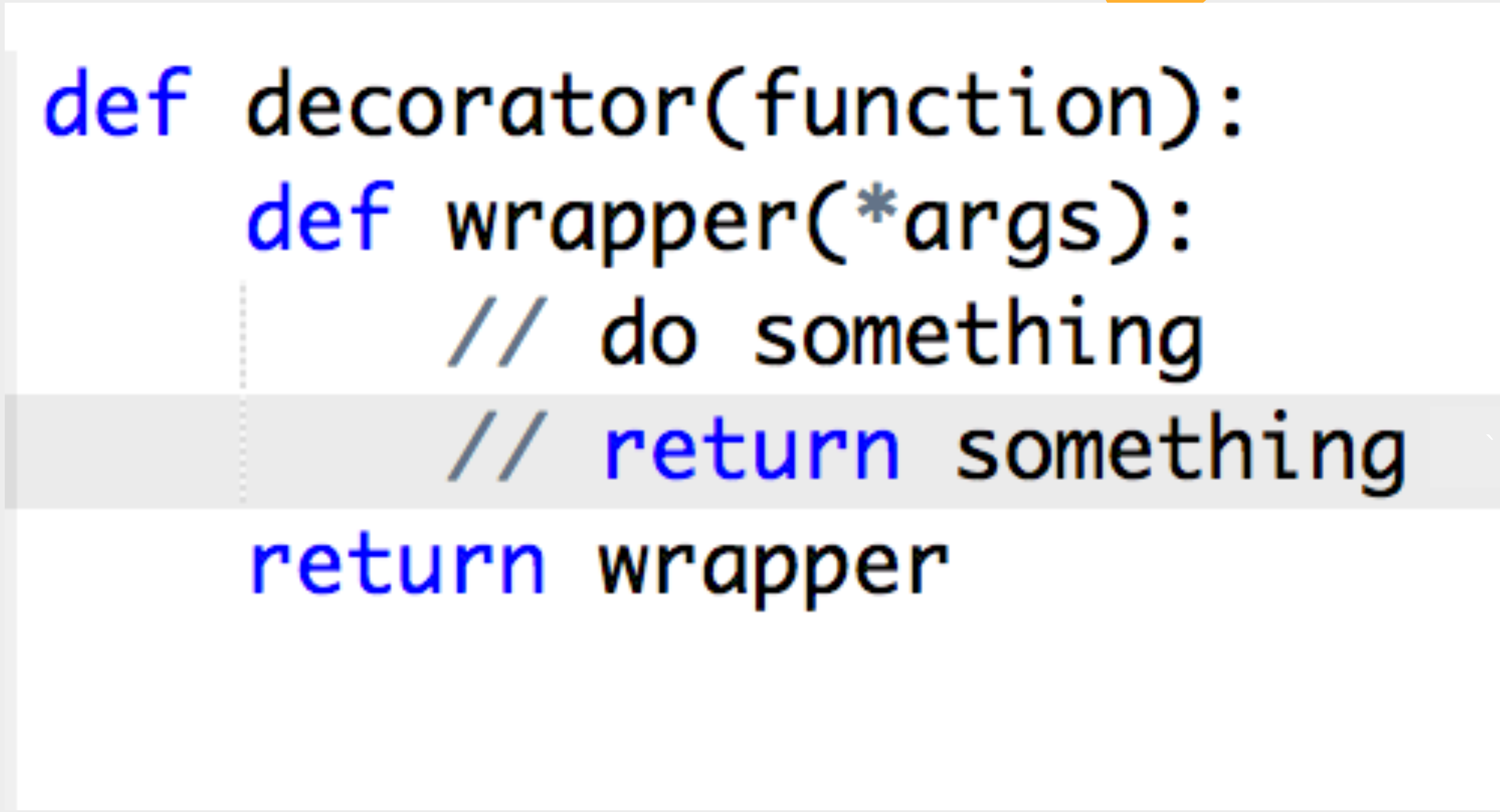
Why Decorators?

- Analytics & Logging
- Validation & Runtime Checks
- Code Reusability
- Writing cleaner code

Decorator Use-Cases

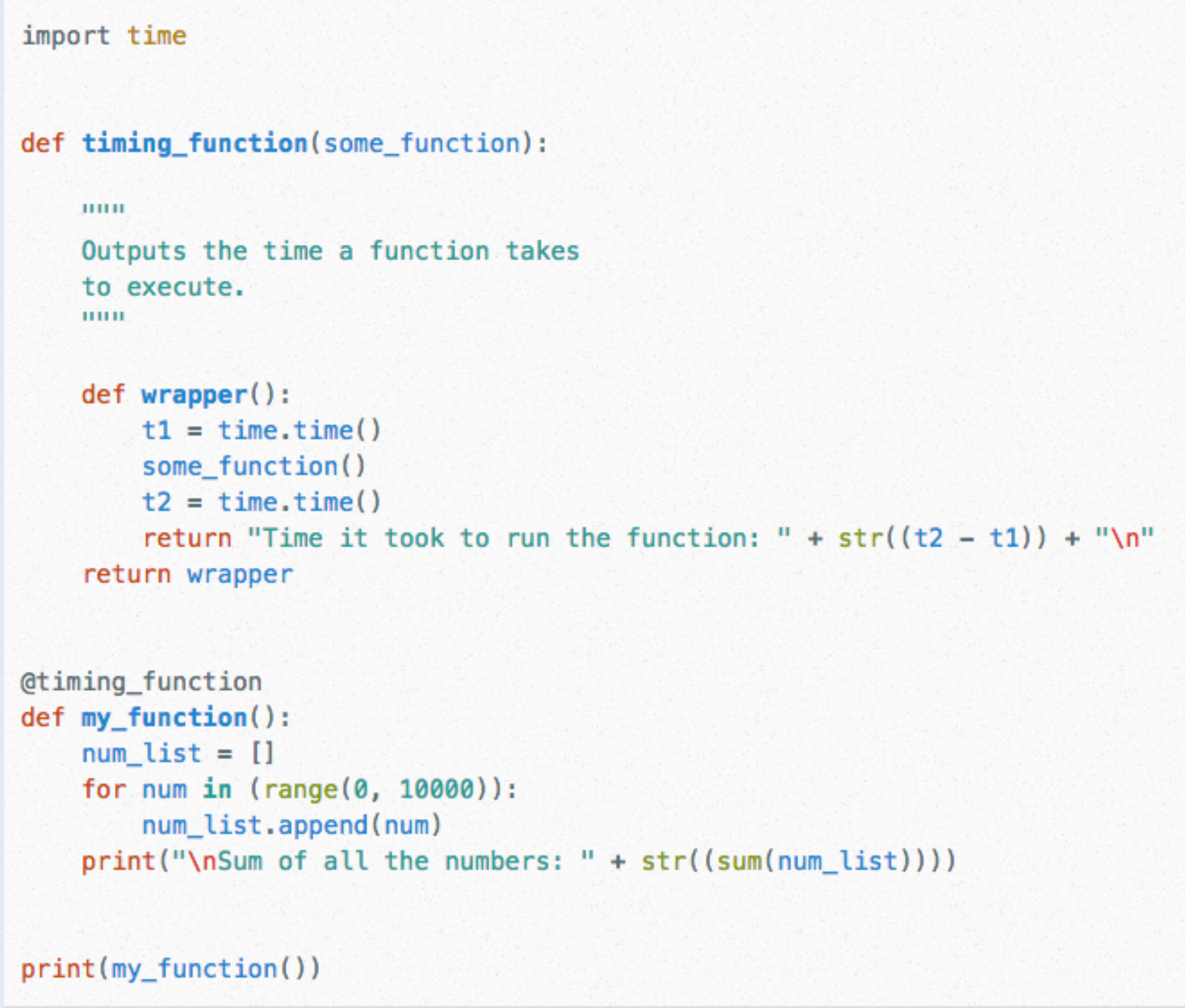
THOSE PYTHON DECORATORS
SO HOT RIGHT NOW

Decorator Syntax



```
def decorator(function):
    def wrapper(*args):
        // do something
        // return something
    return wrapper
```

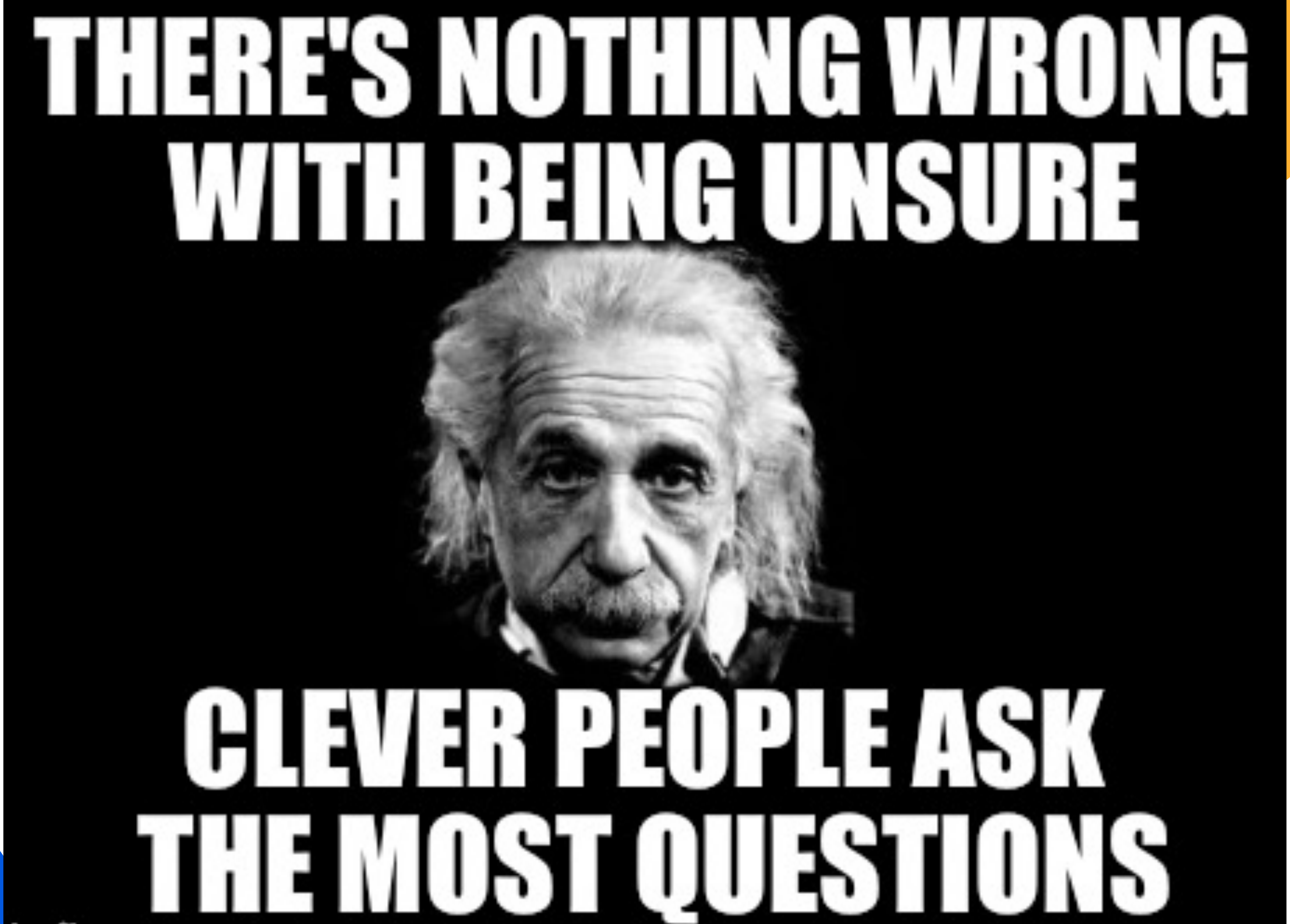
ANALYTICS WITH DECORATORS



```python
import time


def timing_function(some_function):

    """
    Outputs the time a function takes
    to execute.
    """

    def wrapper():
        t1 = time.time()
        some_function()
        t2 = time.time()
        return "Time it took to run the function: " + str((t2 - t1)) + "\n"
    return wrapper


@timing_function
def my_function():
    num_list = []
    for num in (range(0, 10000)):
        num_list.append(num)
    print("\nSum of all the numbers: " + str((sum(num_list))))


print(my_function())
```

BRACE YOURSELVES
A LIVE DEMO IS COMING

PRESENTATION FINISHED
ANY QUESTIONS...


THERE'S NOTHING WRONG WITH BEING UNSURE
CLEVER PEOPLE ASK THE MOST QUESTIONS