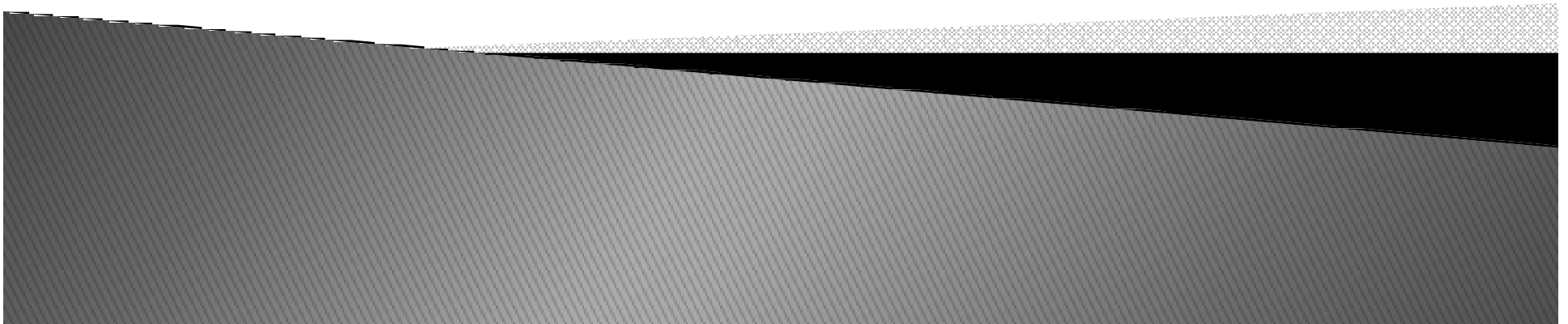


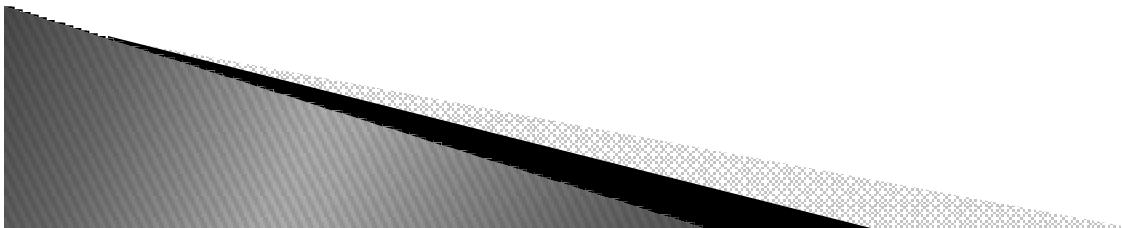
# Testes Unitários

Gustavo Callou  
gcallou@gmail.com



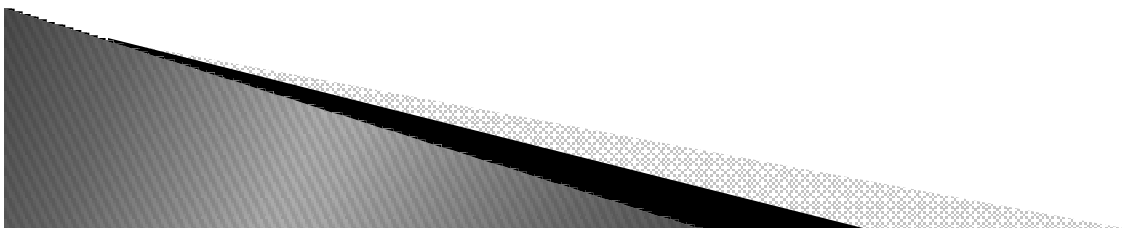
# Testes Unitários

- ▶ Quando acabamos de desenvolver um software, será que acabou o serviço?
- ▶ Como saber se os requisitos foram atendidos?
- ▶ Os testes unitários podem ajudar.
- ▶ É conhecido o comportamento do código para determinadas entradas.
- ▶ Se o código se comportar conforme o esperado para determinada entrada, se admite que ele está correto.



# Testes Unitários

- ▶ Duas formas fáceis de se automatizar a tarefa de testar:
  - usando o módulo doctest
  - ou unittest.
  - Esse módulos já vem com o Python por padrão.



# Doctest

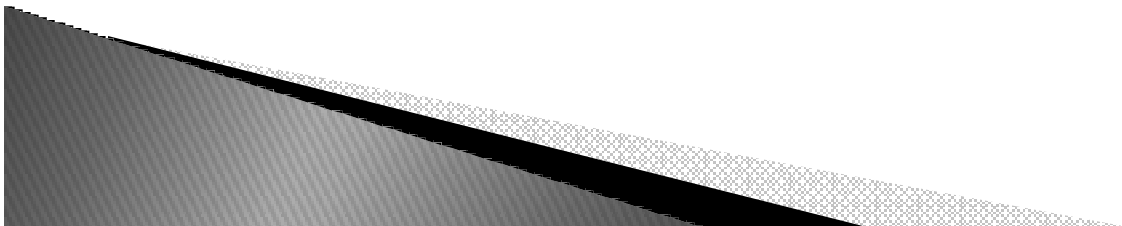
- ▶ Suponha que temos a seguinte função.

```
def fibo(n):  
    if n < 2:  
        return n  
    else:  
        return fibo(n-1) + fibo(n-2)
```

É conhecido o fibonacci de 1 é 1 e o fibonacci de 10 é 55.

Procedimento não automatizado para testar, abrir criar um programa de teste para testar esses valores.

No lugar de sempre criar um programa para testar, vamos usar o Doctest.



# Doctest

- ▶ Com o doctest é possível criar testes unitários ao adicionar linhas de comentários ao seu código e depois fazer uma chamada ao doctest.
- ▶ Exemplo:

```
def fibo(n):  
    """  
    >>> fibo(0)  
    0  
    >>> fibo(1)  
    1  
    >>> fibo(10)  
    55  
    """  
  
    if n < 2:  
        return n  
    else:  
        return fibo(n-1) + fibo(n-2)  
  
import doctest  
doctest.testmod()
```

# Doctest

- ▶ Com o doctest é possível criar testes unitários ao adicionar linhas de comentários ao seu código e depois fazer uma chamada ao doctest.
- ▶ Exemplo:

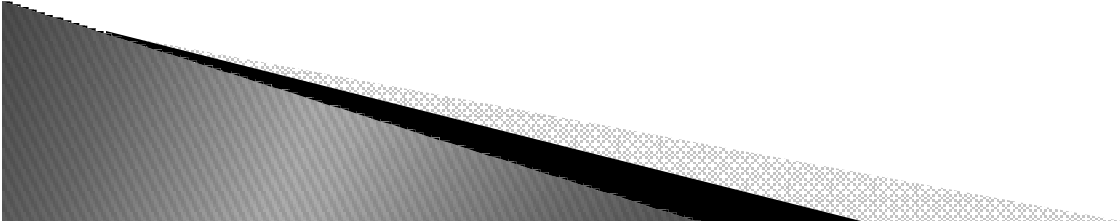
```
def fibo(n):  
    """  
    >>> fibo(0)  
    0  
    >>> fibo(1)  
    1  
    >>> fibo(10)  
    55  
    """  
    if n < 2:  
        return n  
    else:  
        return fibo(n-1) + fibo(n-2)  
  
import doctest  
doctest.testmod()
```

# Doctest

Exemplo do erro ao executar o teste:

```
$ python fibo.py
*****
File "fibo.py", line 7, in __main__.fibo
Failed example:
fibo(10)
Expected:
55
Got:
20
*****

1 items had failures:
1 of 3 in __main__.fibo
***Test Failed*** 1 failures.
```



# Unittest

```
import unittest
from fibo import fibo

class testa_fibonacci(unittest.TestCase):
    def teste_um(self):
        self.assertEqual(fibo(0),0)

    def teste_dois(self):
        self.assertEqual(fibo(1),1)

    def teste_tres(self):
        self.assertEqual(fibo(7),13)

    def teste_quatro(self):
        self.assertEqual(fibo(10),55)

unittest.main()
```





# Exercício

- ▶ Definir uma classe fibo, e fazer 2 testes unitários (1 Doctest e 1 Unittest)

