

# Computação I - Python

## Aula 2 - Função

### Decomposição de funções

Apresentado por: Julio Barbieri

Produção DCC-UFRJ

Metodologia de referência <https://doi.org/10.5753/wei.2016.9683>



Calcule a área da coroa circular (anel) formada por dois círculos de raios  $r_1$  e  $r_2$  ( $r_1 > r_2$  e  $Pi = 3.14$ ).

# Função

Calcule a área da coroa circular (anel) formada por dois círculos de raios  $r_1$  e  $r_2$  ( $r_1 > r_2$  e  $Pi = 3.14$ ).

```
1 def coroa(r1, r2):  
2     """Funcao que calcula a coroa circular formada pelos circulos  
   de raio r1 e r2 (r1 > r2)"""  
3     return (3.14*r1**2) - (3.14*r2**2)
```

# Função

Calcule a área da coroa circular (anel) formada por dois círculos de raios  $r_1$  e  $r_2$  ( $r_1 > r_2$  e  $Pi = 3.14$ ).

```
1 def coroa(r1, r2):  
2     """Funcao que calcula a coroa circular formada pelos circulos  
   de raio r1 e r2 (r1 > r2)"""  
3     return (3.14*r1**2) - (3.14*r2**2)
```

Calcule a área de um círculo de raio  $R$ .

# Função

Calcule a área da coroa circular (anel) formada por dois círculos de raios  $r_1$  e  $r_2$  ( $r_1 > r_2$  e  $Pi = 3.14$ ).

```
1 def coroa(r1, r2):  
2     """Funcao que calcula a coroa circular formada pelos circulos  
   de raio r1 e r2 (r1 > r2)"""  
3     return (3.14*r1**2) - (3.14*r2**2)
```

Calcule a área de um círculo de raio  $R$ .

```
1 def areac(R):  
2     """Funcao que calcula a area de um circulo de raio R"""  
3     return 3.14*R**2
```

# Função

Calcule a área da coroa circular (anel) formada por dois círculos de raios  $r_1$  e  $r_2$  ( $r_1 > r_2$  e  $Pi = 3.14$ ).

```
1 def coroa(r1, r2):  
2     """Funcao que calcula a coroa circular formada pelos circulos  
   de raio r1 e r2 (r1 > r2)"""  
3     return (3.14*r1**2) - (3.14*r2**2)
```

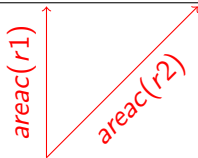
Calcule a área de um círculo de raio  $R$ .

```
1 def areac(R):  
2     """Funcao que calcula a area de um circulo de raio R"""  
3     return 3.14*R**2
```

**O que estas duas funções têm em comum?**

# Função

```
def coroa(r1,r2):  
    return areac(r1) - areac(r2)
```



```
def areac(R):  
    return 3.14*R**2
```

**Posso chamar uma função a partir de outra!**

# Função

```
1 def areac(R):  
2     """ Funcao que calcula a area de um circulo de raio R"""  
3     return 3.14*R**2  
4  
5 def coroa(r1, r2):  
6     """ Funcao que calcula a coroa circular formada pelos circulos  
7     de raio r1 e r2 (r1 > r2)"""  
8     return areac(r1) - areac(r2)
```

**Posso chamar uma função a partir de outra!**



# Função

Chamamos a função *coroa* com os parâmetros 3 e 2

```
coroa(3,2)
```

# Função

Chamamos a função *coroa* com os parâmetros 3 e 2

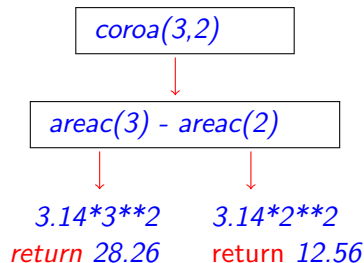
*coroa(3,2)*



*areac(3) - areac(2)*

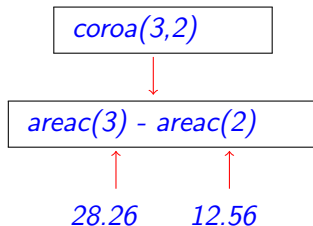
Que chama a função *areac* duas vezes: uma com o parâmetro 3 e outra com o parâmetro 2

# Função



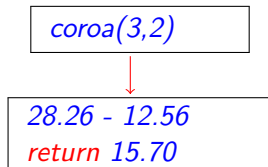
Cada chamada da função *areac* retorna o valor calculado para a função *coroa*

# Função



Cada chamada da função *areac* retorna o valor calculado para a função *coroa*

# Função




A função *coroa* usa os valores retornados pelas chamadas da função *areac* e calcula o valor da coroa.

# Função

Podemos usar a função **quadrado** que definimos na aula anterior

```
def coroa(r1,r2):  
    return areac(r1) - areac(r2)
```

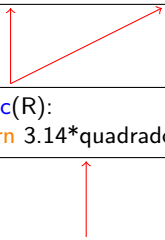
```
def areac(R):  
    return 3.14*R**2
```



# Função

Podemos usar a função **quadrado** que definimos na aula anterior

```
def coroa(r1,r2):  
    return areac(r1) - areac(r2)
```



```
def areac(R):  
    return 3.14*quadrado(R)
```

```
def quadrado(X):  
    return X**2
```

# Função

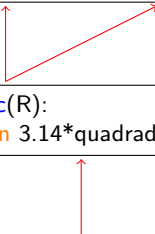
```
1 def quadrado(X):  
2     """Funcao que retorna o quadrado de um numero"""  
3     return X**2  
4  
5 def areac(R):  
6     """Funcao que calcula a area de um circulo de raio R"""  
7     return 3.14*quadrado(R)  
8  
9 def coroa(r1, r2):  
10    """Funcao que calcula a coroa circular formada pelos circulos  
    de raio r1 e r2 (r1 > r2)"""  
11    return areac(r1) - areac(r2)
```



# Função

Pi é bastante usado. Por que não definimos uma função (constante) para ele?

```
def coroa(r1,r2):  
    return areac(r1) - areac(r2)
```



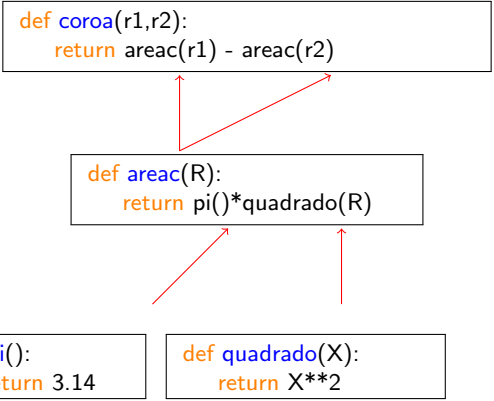
```
def areac(R):  
    return 3.14*quadrado(R)
```

```
def quadrado(X):  
    return X**2
```

# Função

Pi é bastante usado. Por que não definimos uma função (constante) para ele?

```
def coroa(r1,r2):  
    return areac(r1) - areac(r2)
```



```
def areac(R):  
    return pi()*quadrado(R)
```

```
def pi():  
    return 3.14
```

```
def quadrado(X):  
    return X**2
```

# Função

```
1 def pi():
2     """Funcao que define o valor de Pi como sendo 3.14"""
3     return 3.14
4
5 def quadrado(X):
6     """Funcao que retorna o quadrado de um numero"""
7     return X**2
8
9 def areac(R):
10    """Funcao que calcula a area de um circulo de raio R"""
11    return pi()*quadrado(R)
12
13 def coroa(r1, r2):
14    """Funcao que calcula a coroa circular formada pelos circulos
15    de raio r1 e r2 (r1 > r2)"""
16    return areac(r1) - areac(r2)
```

## Autores

- **João C. P. da Silva** ▶ Lattes
- **Carla Delgado** ▶ Lattes
- **Ana Luisa Duboc** ▶ Lattes

## Colaboradores

- **Anamaria Martins Moreira** ▶ Lattes
- **Fabio Mascarenhas** ▶ Lattes
- **Leonardo de Oliveira Carvalho** ▶ Lattes
- **Charles Figueiredo de Barros** ▶ Lattes
- **Fabício Firmino de Faria** ▶ Lattes

# Computação I - Python

## Aula 2 - Função

### Decomposição de funções

Apresentado por: Julio Barbieri

Produção DCC-UFRJ

Metodologia de referência <https://doi.org/10.5753/wei.2016.9683>

