

## 13 Reunião 13: 13/OUT/2020

### 13.1 Reunião passada

- funções
- protótipos de funções
- estruturas de funções
- chamada de funções
- passagem de parâmetros
- esqueleto de um programa com funções



Figure 1: Fonte: <https://makeameme.org/>

## 13.2 Hoje

Funções, hoje, amanhã e sempre!

- protótipos de funções
- estruturas de funções
- parâmetros
- chamada de funções
- argumentos
- **toda** função retorna um valor
- nosso esqueleto de programa com funções

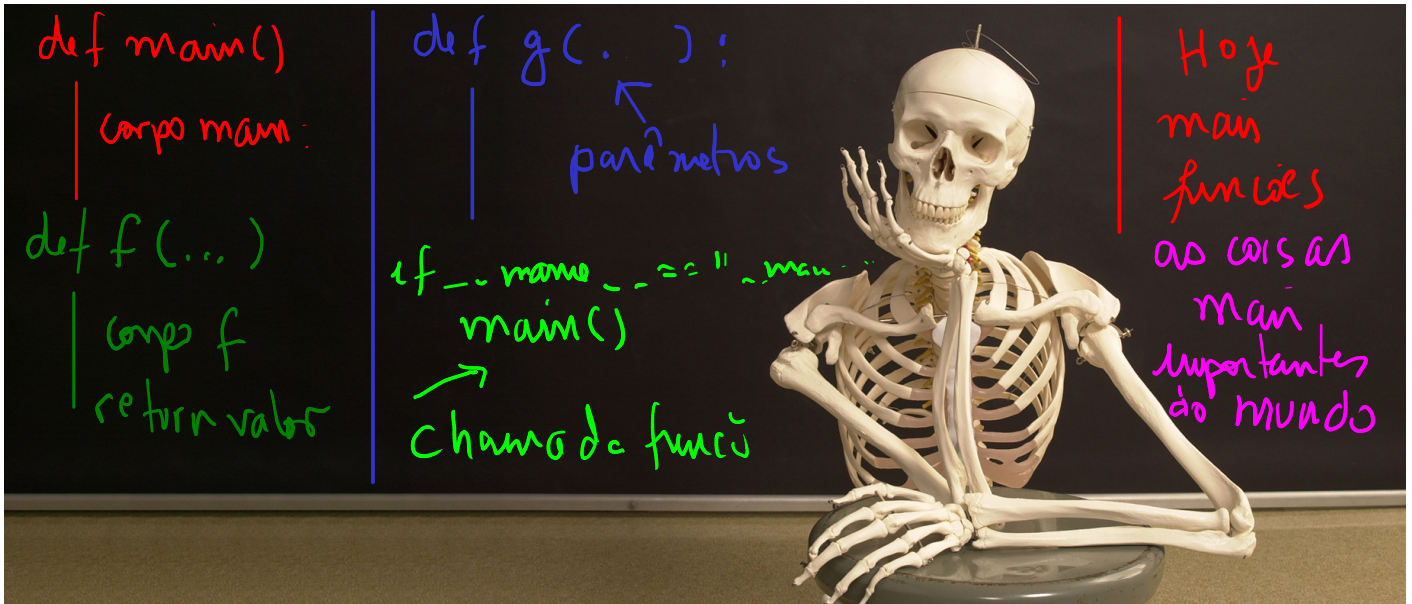
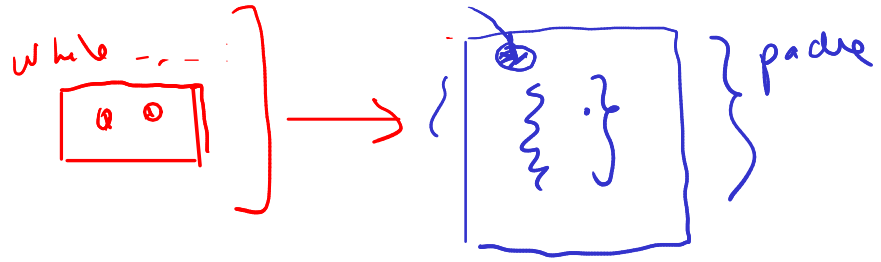
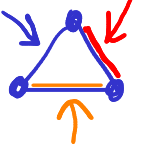


Figure 2: Fonte: <https://www.institutoclaro.org.br/>

### 13.3 Exercício: Pascal

Dado um número inteiro positivo  $m$ , determine os inteiros na linha  $m$  do triângulo de pascal

$$\binom{m}{0}, \binom{m}{1}, \binom{m}{2}, \dots, \binom{m}{m} \quad \binom{3}{2} = 3$$



{ EPot  
Integral  
área  
 $\binom{m}{n} = \frac{m!}{n!(m-n)!}$

#### 13.3.1 Exemplos

Digite m: 1  
 $\binom{1}{0} = 1$   
 $\binom{1}{1} = 1$

Digite m: 3  
 $\binom{3}{0} = 1$   
 $\binom{3}{1} = 3$   
 $\binom{3}{2} = 3$   
 $\binom{3}{3} = 1$

$$\binom{3}{0} \quad \binom{3}{1} \quad \binom{3}{2} \quad \binom{3}{3}$$

1            3            3            1

$$\binom{3}{2} = \frac{3 \times 2 \times 1}{2 \times 1 \times 1} = 3$$

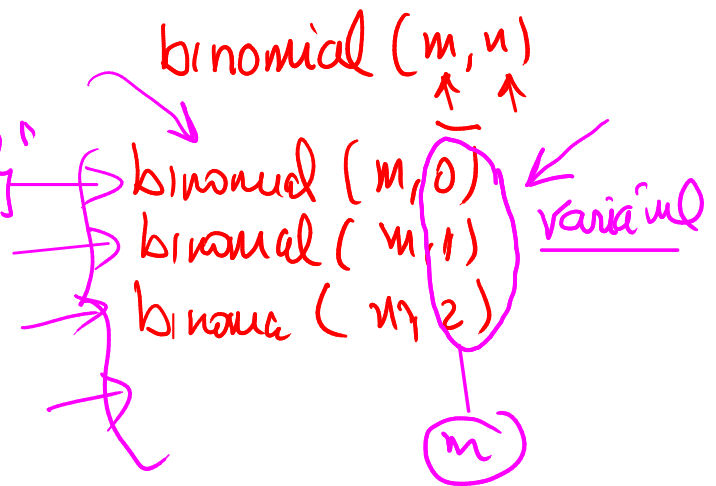
$$\binom{m}{0} \quad \binom{m}{1} \quad \dots \quad \binom{m}{m}$$

$$\binom{m}{0} \quad \binom{m}{1}$$

```
def main():
    m = int(input())
```

```
    r = 0
    while i <= m:
```

```
        print(f" {binomial(m, i)}")
```



### 13.3.3 Solução

```
def main():
    '''
    Programa que lê um número inteiro m e imprime

    binomial(m,0), binomial(m,1),..., binomial(m,m)
    '''
    print("Elementos em uma linha do triângulo de Pascal")
    m = int(input("Digite m: "))
    i = 0
    while i <= m:
        print(f"binomial({m}, {i}) = {binomial(m,i)}")
        i = i + 1

#-----
def binomial(m, n):
    '''(int, int) -> int

    Recebe inteiros não negativos m e n e imprime o
    número de combinações de m, n-a-n.
    '''
    # calcule m!
    m_fat = fatorial(m)

    # calcule n!
    n_fat = fatorial(n)

    # calcule m-n!
    mn_fat = fatorial(m-n)

    return m_fat // (n_fat * mn_fat)

#-----
def fatorial(k):
    '''(int) -> int

    Recebe um inteiro k e retorna k!
    '''
    k_fat = 1
    i = 2
    while i <= k:
        k_fat *= i
        i += 1
    return k_fat

# chamada da função main
if __name__ == "__main__":
    main()
```

## 13.4 Declaração de funções

```
def "nome da função"("parâmetros"):  
    '''Comentários (opcional, recomendado)  
    '''  
    # corpo da função  
    |  
    | bloco de comandos  
    |
```

Variáveis criadas dentro de uma função são locais, isto é, só existem dentro da função. Parâmetros são variáveis locais criadas na chamada da função.

## 13.5 Término da execução

Após a execução do comando `return` a execução da função é abandonada.

```
return "expressão"
```

Uma função também pode ser `void` e nesse caso a função não precisa de `return`.

## 13.6 Chamada da função

```
n_fat = fatorial(n)  
mn_fat = fatorial(m-n)
```

## 13.7 Anatomia de um programa em Python

```
# -----  
def main():  
    ''' Programa principal  
    '''  
    # corpo da função  
    |  
    | bloco de comandos  
    |  
  
# Declaração das funções  
def f(...):  
    # corpo da função f  
    |  
    | bloco de comandos  
    |  
  
def g(...):  
    # corpo da função g  
    |  
    | bloco de comandos  
    |  
  
# início do programa  
if __name__ == "__main__":  
    main() # chamada da função
```

### 13.8 Exercício: frações irredutíveis

Dados dois inteiros  $m$  e  $n$ ,  $n > 0$ , determinar a fração irredutível  $m/n$

} ← função

#### 13.8.1 Exemplos

Digite o num: 8  
Digite o den: 12  
Fracao irredutivel: 2/3

$$\frac{8}{12} = \frac{2}{3}$$

$$\frac{8}{12} = \frac{\cancel{2 \times 2 \times 2}}{\cancel{2 \times 2 \times 3}}$$

Digite o num: 21  
Digite o den: 7  
Fracao irredutivel: 3/1

$$\frac{21}{7} = \frac{\cancel{3 \times 7}}{\cancel{7}} = 3$$

Digite o num: 2  
Digite o den: 9  
Fracao irredutivel: 2/9

Digite o num: 3  
Digite o den: 3  
Fracao irredutivel: 1/1

### 13.8.3 Solução

```
def main():
    ''' Recebe dois inteiros m e n e retorna a fração
        irredutível coorespondente a m/n.

        Pré-condição: supõe m >= 0 e n > 0.
    '''
    m = int(input("Digite o num: "))
    n = int(input("Digite o den: "))

    d = mdc(m,n)
    print(f"Fracao irredutivel: {m//d}/{n//d}")

#-----
def mdc(m, n):
    '''(int, int) -> int
        Recebe dois numeros inteiros m e n e retorna o
        máximo divisor comum de m e n.

        Pré-condição: supõe n > 0. (ou n != 0)
    '''
    d = n
    while m%d != 0 or n%d != 0:
        d = d - 1;
    return d

#-----
if __name__ == "__main__":
    main()
```