# 12 Reunião 12: 08/OUT/2020

## 12.1 Reuniões passadas

- treino de raciocício (sempre!)
- decomposição de um problema em problemas mais simples (sempre!): fatoração de inteiros

- indicadores de passagem: números primos
- repetições encaixadas: `while ...` dentro de `while ...` dentro de ...

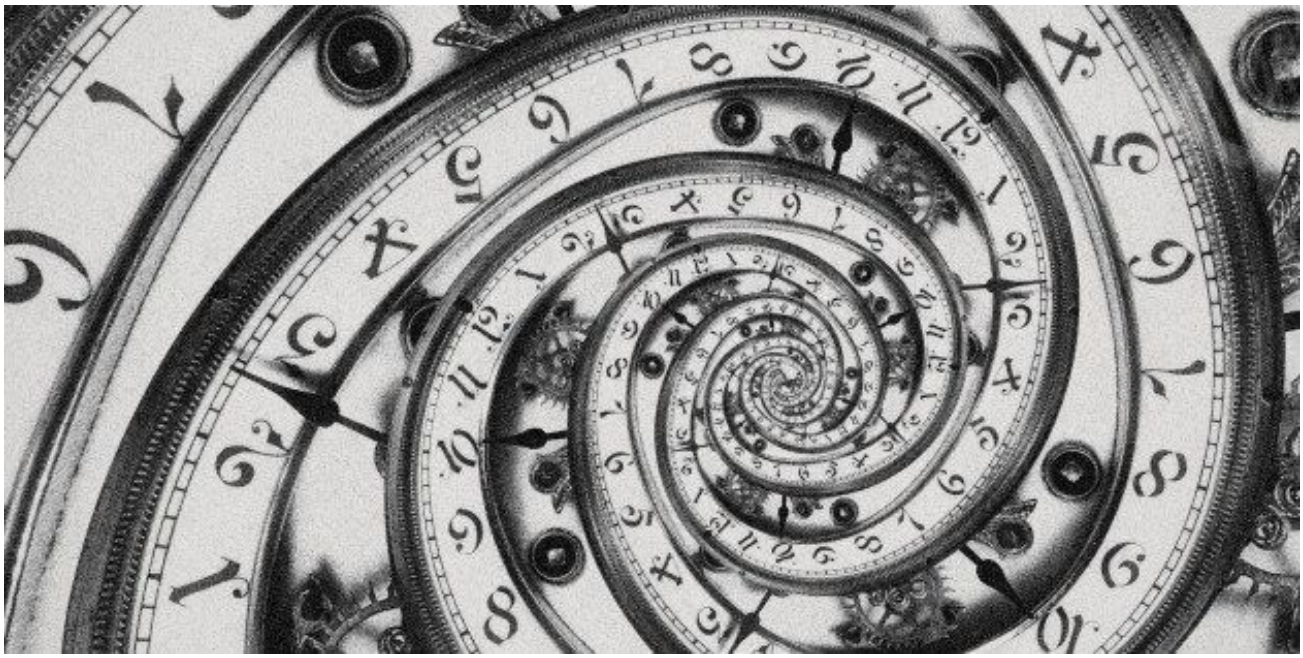


Figure 1: Fonte: Alex teuscher via Getty Images

## 12.2 Hoje

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


Figure 2: Fonte: `https://makeameme.org/`

## 12.3 Exercício: binomial

Escreva um programa que leia dois inteiros `m` e `n` e calcule o coeficiente binomial

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

### 12.3.1 Exemplos

Para `n = 3` e `m = 5` seu programa deve imprimir

```
Digite m: 5
Digite n: 3
binomial(5,3) = 10

Digite m: 4
Digite n: 2
binomial(4,2) = 6

Digite m: 3
Digite n: 1
binomial(3,1) = 3
```

### 12.3.2 Solução 1

Solução simples e na *raça*.

```
def main():
    m = int(input("Digite m: "))
    n = int(input("Digite n: "))

    # calcule m!
    m_fat = 1
    i = 2
    while i <= m:
        m_fat *= i
        i += 1

    # calcule n!
    n_fat = 1
    i = 2
    while i <= n:
        n_fat *= i
        i += 1

    # calcule (m-n)!
    mn_fat = 1
    i = 2
    while i <= m-n:
        mn_fat *= i
        i += 1

    binomial = m_fat//(n_fat * mn_fat)

    print(f"binomial({m},{n}) = {binomial}")
```

### 12.3.3 Solução 2

Solução deixando evidente os padrões.

```
def main():
    m = int(input("Digite m: "))
    n = int(input("Digite n: "))

    # calcule m!
    k = m  # na chamada da função

    # corpo da função
    k_fat = 1
    i = 2
    while i <= k:
        k_fat *= i
        i += 1

    m_fat = k_fat # valor retornado


    #------------------------------
    # calcule n!
    k = n  # na chamada da função

    # corpo da função
    k_fat = 1
    i = 2
    while i <= k:
        k_fat *= i
        i += 1

    n_fat = k_fat # valor retornado

    #------------------------------
    # calcule m-n!
    k = m-n # na chamada da função

    # corpo da função
    k_fat = 1
    i = 2
    while i <= k:
        k_fat *= i
        i += 1

    mn_fat = k_fat # valor retornado

    binomial = m_fat//(n_fat * mn_fat)

    print(f"binomial({m},{n}) = {binomial}")
```

### 12.3.4 Solução 3

Solução em que os padrões são substituídos por funções.

```
def main():
    '''
    Programa que lê dois inteiros positivos m
    e n e calcula binomial(m,n).
    >>>
    Digite m: 5
    Digite n: 3
    Binomial(5,3) = 10
    >>>
    '''
    m = int(input("Digite m: "))
    n = int(input("Digite n: "))

    # calcule m!
    m_fat = fatorial(m)

    # calcule n!
    n_fat = fatorial(n)

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

    binomial = m_fat//(n_fat * mn_fat)

    print(f"binomial({m},{n}) = {binomial}")

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

    Recebe um inteiro k, k >= 0, 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()
```

## 12.4 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.

## 12.5 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.

## 12.6 Chamada da função

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

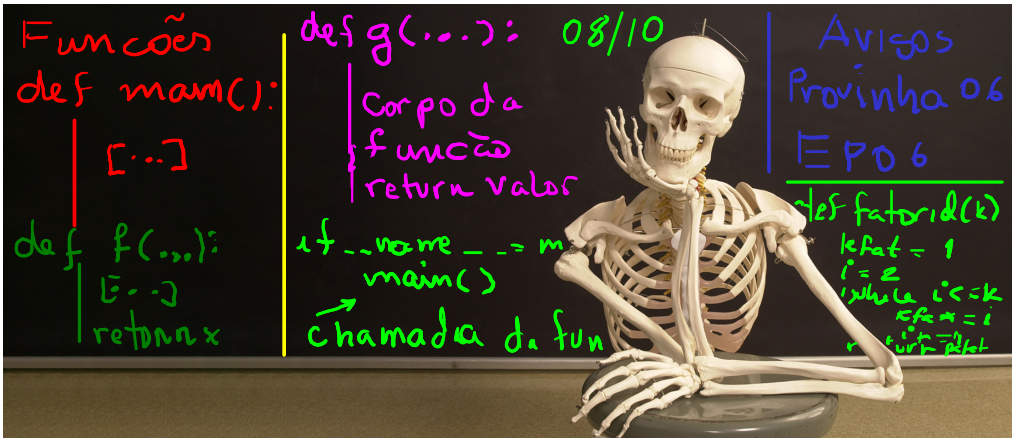
## 12.7 Anatomia de um programa em Python

```
# -------------------------------
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
```



## 12.8 Exercício: Pascal

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

$$\binom{m}{0}, \binom{m}{1}, \binom{m}{2}, \ldots, \binom{m}{m}.$$

### 12.8.1 Exemplos

```
Digite m: 1
binomial(1, 0) = 1
binomial(1, 1) = 1

Digite m: 3
binomial(3, 0) = 1
binomial(3, 1) = 3
binomial(3, 2) = 3
binomial(3, 3) = 1
```

### 12.8.2 Rascunhos