



Universidade de Brasília

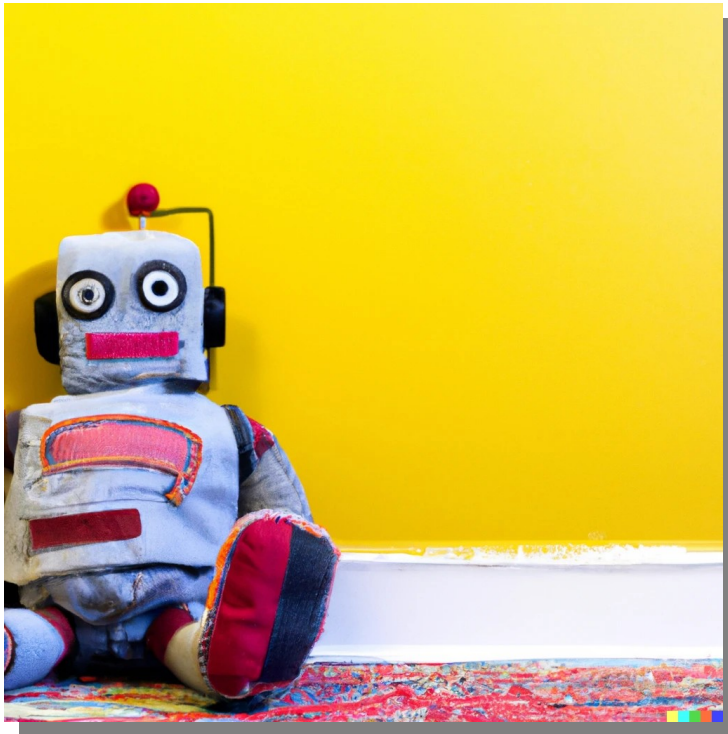
Departamento de Ciência da Computação

Semana 05

Algoritmos: **Procedimentos, Variáveis,** **Expressões e Instruções**

CIC0004

Algoritmos e Programação de Computadores



Prof. Pedro Garcia Freitas

<https://pedrogarcia.gitlab.io/>

pedro.garcia@unb.br

Brasília



Este conjunto de slides não deve ser utilizado ou republicado sem a expressa permissão do autor.

This set of slides should not be used or republished without the author's express permission.



Exercício

Exercício: A série de Taylor é uma série de funções que é utilizada para reescrevermos uma função contínua como um polinômio! Tal série pode ser descrita da seguinte forma:

$$f(x) = \sum_{n=0}^{\infty} a_n (x-a)^n \text{ onde } a_n = \frac{f^{(n)}(a)}{n!},$$



Exercício

onde $f(x)$ é uma função analítica derivável. Assim, o polinômio de Taylor de ordem n em torno de a de uma função n -vezes diferenciável em $x = a$ é dado por:

$$f(x) \approx p(x) = f(a) + f'(a) \frac{(x-a)^1}{1!} + f''(a) \frac{(x-a)^2}{2!} + \dots + f^{(n)}(a) \frac{(x-a)^n}{n!}$$

No caso particular de $a=0$, o polinômio gerado é conhecido como Série de Maclaurin, expressa por

$$f(x) \approx p(x) = f(0) + f'(0) \frac{x^1}{1!} + f''(0) \frac{x^2}{2!} + \dots + f^{(n)}(0) \frac{x^n}{n!} = \sum_{n=0}^{\infty} \frac{f^{(n)}(0)}{n!} x^n$$



Exercício

Considerando a fórmula acima, resolva os itens a seguir:

- (a) Escreva um algoritmo em pseudocódigo/portugol para calcular o fatorial de um número natural qualquer.
- (b) Escreva um algoritmo em pseudocódigo/portugol que recebe como entrada um valor angular em graus e converte esse valor em radianos.



Exercício

(c) Escreva um algoritmo em pseudocódigo/portugol que calcule o **cosseno** de um valor x por meio da expansão de n termos da série de Maclaurin. Dica: a expansão dessa série de Maclaurin é dada pelo polinômio abaixo:

$$\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots$$

(d) Converta/implemente os algoritmos acima em *linguagem Python* e compare o resultado a função implementada em **math.cos**.



Exercício

(a) Escreva um algoritmo em pseudocódigo/portugol para calcular o fatorial de um número natural qualquer.

```
FUNÇÃO FatorialV1(INTEIRO n)
  VAR INTEIRO fat = 1
  VAR INTEIRO i = n
  SE n > 1 ENTÃO
    ENQUANTO i >= 1 FAÇA
      fat = fat * i
      i = i - 1
    FIM-ENQUANTO
  FIM-SE
  RETORNA fat
FIM-FUNÇÃO
```



Exercício

(a) Escreva um algoritmo em pseudocódigo/portugol para calcular o fatorial de um número natural qualquer.

```
FUNÇÃO FatorialV1(INTEIRO n)
  VAR INTEIRO fat = 1
  VAR INTEIRO i = n
  SE n > 1 ENTÃO
    ENQUANTO i >= 1 FAÇA
      fat = fat * i
      i = i - 1
    FIM-ENQUANTO
  FIM-SE
  RETORNA fat
FIM-FUNÇÃO
```

```
def fact(n):
    fat = 1
    i = n
    if n > 1:
        while i >= 1:
            fat = fat * i
            i = i - 1
    return fat
```




Exercício

(a) Escreva um algoritmo em pseudocódigo/portugol para calcular o fatorial de um número natural qualquer.

```
FUNÇÃO FatorialV2(INTEIRO n)
  SE n > 1 ENTÃO
    RETORNA n * FatorialV2(n - 1)
  SENÃO
    RETORNA 1
  FIM-SE
FIM-FUNÇÃO
```



Exercício

(a) Escreva um algoritmo em pseudocódigo/portugol para calcular o fatorial de um número natural qualquer.

```
FUNÇÃO FatorialV2(INTEIRO n)
  SE n > 1 ENTÃO
    RETORNA n * FatorialV2(n - 1)
  SENÃO
    RETORNA 1
  FIM-SE
FIM-FUNÇÃO
```

```
def fact(n):
    if n > 1:
        return n * fact(n - 1)
    else:
        return 1
```



Exercício

(a) Escreva um algoritmo em pseudocódigo/portugol para calcular o fatorial de um número natural qualquer.

```
FUNÇÃO FatorialV2(INTEIRO n)
  SE n > 1 ENTÃO
    RETORNA n * FatorialV2(n - 1)
  SENÃO
    RETORNA 1
  FIM-SE
FIM-FUNÇÃO
```

```
def fact(n):
    if n > 1:
        return n * fact(n - 1)
    else:
        return 1
```

```
def fact(n):
    return 1 if n < 1 else n * fact(n - 1)
```



Exercício

(a) Escreva um algoritmo em pseudocódigo/portugol para calcular o fatorial de um número natural qualquer.

```
1  def fact(n) :  
2      return 1 if n < 1 else n * fact(n - 1)  
3  
4  
5  print(fact(0))  
6  print(fact(1))  
7  print(fact(2))  
8  print(fact(3))  
9  print(fact(4))  
10 print(fact(5))  
11 print(fact(10))  
12
```

```
Shell x  
>>> %Run -c $EDITOR_CONTENT  
  
1  
1  
2  
6  
24  
120  
3628800
```



Exercício

(b) Escreva um algoritmo em pseudocódigo/portugol que recebe como entrada um valor angular em graus e converte esse valor em radianos.

```
FUNÇÃO GrausEmRadianos (REAL ângulo)  
    REAL PI = 3.141592  
    RETORNA angulo*PI/180  
FIM-FUNÇÃO
```



Exercício

(b) Escreva um algoritmo em pseudocódigo/portugol que recebe como entrada um valor angular em graus e converte esse valor em radianos.

```
FUNÇÃO GrausEmRadianos(REAL ângulo)  
    REAL PI = 3.141592  
    RETORNA angulo*PI/180  
FIM-FUNÇÃO
```

```
def angle2grads(angulo):  
    pi = 3.141592  
    return angulo * pi / 180.0
```



Exercício

(b) Escreva um algoritmo em pseudocódigo/portugol que recebe como entrada um valor angular em graus e converte esse valor em radianos.

```
1 def angle2grads(angulo):  
2     pi = 3.141592  
3     return angulo * pi / 180.0  
4  
5  
6 print(angle2grads(0))  
7 print(angle2grads(90))  
8 print(angle2grads(180))  
9 print(angle2grads(270))  
10 print(angle2grads(360))  
11
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
```

```
0.0  
1.570796  
3.141592  
4.7123880000000001  
6.283184
```

```
>>>
```



Exercício

(c) Escreva um algoritmo em pseudocódigo/portugol que calcule o **cosseno** de um valor x por meio da expansão de n termos da série de Maclaurin. Dica: a expansão dessa série de Maclaurin é dada pelo polinômio abaixo:

$$\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots$$



Exercício

```
FUNÇÃO cos(REAL angulo, INTEIRO n)
  INTEIRO i = 0
  REAL serie = 0
  REAL angulo_grad = GrausEmRadianos(angulo)
  ENQUANTO i <= n FAÇA
    serie = serie +  $((-1)^n / \text{FatorialV2}(2 * i)) * (\text{angulo\_grad}^{2*i})$ 
    i = i + 1
  FIM-ENQUANTO
  RETORNA serie
FIM-FUNÇÃO
```



Exercício

(d) Converta/implemente os algoritmos acima em *linguagem Python* e compare o resultado a função implementada em `math.cos`.

```
def cos_v1(a, n=50):  
    series = 0  
    x = angle2grads(a)  
    for i in range(n):  
        term = ((-1) ** i) / fact(2*i)  
        term *= x ** (2 * i)  
        series += term  
    return series
```

```
def cos_v2(a, n=50):  
    def cos(x, it):  
        if it <= 0:  
            return 1  
        else:  
            term = ((-1) ** it) / fact(2*it)  
            term *= x ** (2 * it)  
            return cos(x, it - 1) + term  
    x = angle2grads(a)  
    return cos(x, n)
```

Ex

```

1 import math
2
3 def angle2grads(angulo):
4     pi = 3.141592
5     return angulo * pi / 180.0
6
7
8
9 def fact(n):
10     return 1 if n < 1 else n * fact(n - 1)
11
12
13 def cos_v1(a, n=50):
14     series = 0
15     x = angle2grads(a)
16     for i in range(n):
17         term = ((-1) ** i) / fact(2*i)
18         term *= x ** (2 * i)
19         series += term
20     return series
21
22
23 def cos_v2(a, n=50):
24     def cos(x, it):
25         if it <= 0:
26             return 1
27         else:
28             term = ((-1) ** it) / fact(2*it)
29             term *= x ** (2 * it)
30             return cos(x, it - 1) + term
31     x = angle2grads(a)
32     return cos(x, n)
33
34
35 for angle in [0, 90, 180, 270, 360]:
36     v1, v2 = cos_v1(angle), cos_v2(angle)
37     ref = math.cos(angle2grads(angle))
38     diff1, diff2 = math.fabs(ref - v1), math.fabs(ref - v2)
39     print(f"angle={angle}, v1={v1:1.3f} d1={diff1:1.3e}, v2={v2:1.3f} d2={diff2:1.3e}")
40

```

hell x

```
>> %Run -c $EDITOR_CONTENT
```

```

angle=0, v1=1.000 d1=0.000e+00, v2=1.000 d2=0.000e+00
angle=90, v1=0.000 d1=8.201e-17, v2=0.000 d2=8.201e-17
angle=180, v1=-1.000 d1=2.220e-16, v2=-1.000 d2=2.220e-16
angle=270, v1=-0.000 d1=1.422e-15, v2=-0.000 d2=1.422e-15
angle=360, v1=1.000 d1=9.104e-15, v2=1.000 d2=9.104e-15

```

```
>>
```

Prof. Pedro Garcia Freitas



Dúvidas?

