



Processamento da Informação

Conceitos básicos de modularização e Estruturas de seleção

Prof. Jesús P. Mena-Chalco
CMCC/UFABC

Q1/2020



Precedência entre operadores

Em português: a **vírgula** é muito importante

VAMOS PERDER, NADA FOI RESOLVIDO.

VAMOS PERDER NADA, FOI RESOLVIDO.

Não queremos saber.

Não, queremos saber.

Ordem nas operações

Na matemática, os **parênteses** destacam a prioridade de cálculo: **as contas dentro de parênteses são resolvidas primeiro.**

$$(2+3) * 4 - 5 = 15$$

Precedência de operadores

Operador	Operação
**	Exponenciação
-	Negação
*	Multiplicação
/	Divisão
//	Divisão inteira
%	Módulo
+	Soma
-	subtração

Exercício

Escreva a seguinte equação usando operadores binários:

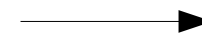
$$x = 7 + \frac{1}{14 - \frac{1}{9}}$$

Exercício

Escreva a seguinte equação usando operadores binários:

$$x = 7 + \frac{1}{14 - \frac{1}{9}}$$

$$x = 7 + 1 / (14 - 1/9)$$



7.072

Da aula anterior

Quais das seguintes instruções dariam erro de sintaxe?

- $6 * \text{-----}8$

- **$8 = \text{alunos}$**

- $((((4 ** 3))))$

- $(-(-(-(-5))))$

- **$4 += 7 / 2$**

O número 8 recebe o valor de uma variável?

A tradução de
 $x += 3$
Seria $x = x+3$

Assim, a tradução seria
 $4 = 4+7/2$
Um número recebe o valor de $4+7/2$

Operadores de atribuição compostos

- $X += 1 \quad \rightarrow \quad X = X+1$
- $X -= 5 \quad \rightarrow \quad X = X-5$
- $X *= 7 \quad \rightarrow \quad X = X*7$
- $X /= 87 \quad \rightarrow \quad X = X/87$

Tambem temos os operadores compostos:

- $**=$
- $//=$
- $/=$
- $\%=$



Questão de estilo?

Questão de estilo?

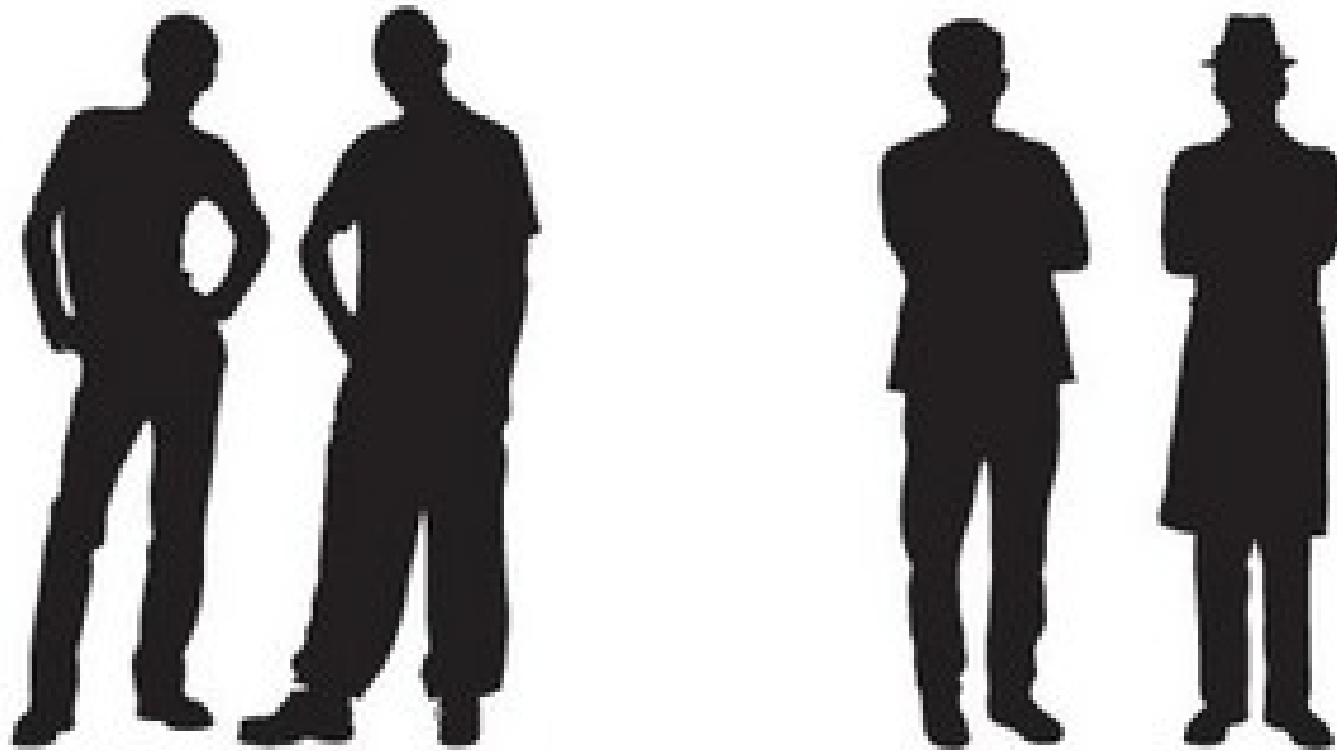
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur vel est augue. Donec aliquam laoreet ipsum, ac sagittis odio aliquam quis. In eleifend est tortor, dapibus posuere odio consectetur id. Donec varius eget est eu luctus. Phasellus scelerisque, nunc laoreet feugiat sagittis, risus neque condimentum nisi, eget feugiat turpis purus quis urna. Mauris tempor eros in turpis tincidunt, bibendum euismod est pretium. Praesent ligula dui, fermentum a porttitor vitae, pulvinar ut odio. Suspendisse vitae pretium dolor, at sodales mauris.

Sed consequat purus nec bibendum suscipit. Donec ultricies euismod enim, quis interdum mi ornare et. Ut facilisis elit vitae elementum fringilla. Pellentesque sed orci iaculis, imperdiet elit et, mollis enim.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur vel est augue. Donec aliquam laoreet ipsum, ac sagittis odio aliquam quis. In eleifend est tortor, dapibus posuere odio consectetur id. Donec varius eget est eu luctus. Phasellus scelerisque, nunc laoreet feugiat sagittis, risus neque condimentum nisi, eget feugiat turpis purus quis urna. Mauris tempor eros in turpis tincidunt, bibendum euismod est pretium. Praesent ligula dui, fermentum a porttitor vitae, pulvinar ut odio. Suspendisse vitae pretium dolor, at sodales mauris.

Sed consequat purus nec bibendum suscipit. Donec ultricies euismod enim, quis interdum mi ornare et. Ut facilisis elit vitae elementum fringilla. Pellentesque sed orci iaculis, imperdiet elit et, mollis enim.

Questão de estilo?



Questão de estilo na escrita

Esta É Uma FRASE Que D3ver14 SeR s1mples de lEr?

Questão de estilo na escrita

Vamos tentar nos organizar para ter um mesmo estilo na escrita de nossas variáveis, dos nossos programas...

As **convenções** são importantes para que times de programação possam interagir de forma mais rápida.

Por exemplo, ao usar uma variável de represente o valor máximo de um conjunto de números, podemos ter:

- `maximo`
- `maxVal`
- `max_val`
- `valor_maximo`

Regras que tentaremos usar <https://wiki.python.org.br/GuiaDeEstilo> [pep8]



Módulos / Funções / Métodos

Motivação

Exemplo de módulos
(componentes) para
Arduino:

- Entrada.
- Saída.



Microphone sensor



Hall sensor



Laser-transmit



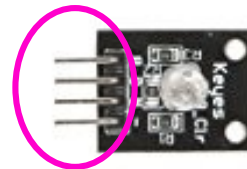
Passiver buzzer



Infrared-transmit



Two-color commoncathode LED



RGB LED



Relay module



Magnet-ring sensor



Obstacle avoidance sensor



Magic-ring



Analog-temperature sensor

Exemplo de uso de funções

```
1 n = int(input())
2 p = abs(n) + pow(n, 2) + round(n/7, 2)
3 print(p)
```

```
3
12.43
❖
❖ help(pow)
Help on built-in function pow in module builtins
:

pow(base, exp, mod=None)
    Equivalent to base**exp with 2 arguments or
    base**exp % mod with 3 arguments
```

Exemplo de definição de uma função

```
1 def dobra(numero):  
2     resposta = numero*2  
3     return resposta  
4  
5 print ( dobra(7) )  
-
```

Def é uma palavra reservada e utilizada para **definir** uma função.

A função tem apenas um parâmetro (ie., variável de entrada).

Usamos 4 espaços em branco para indicar que as seguintes instruções pertencem à função dobra (linhas 2 e 3)

A função (**return**) devolve um número

Caixa preta

```
1 def caixa_preta(numero):  
2     | | return numero * numero * numero  
3  
4 print ( caixa_preta(3) )
```

Caixa preta



Média de 3 números

```
1  def media(a, b, c):  
2  |   | m = (a + b + c)/3  
3  |   | return m  
4  
5  print ( media(1, 2, 3) )
```

Os números podem ser inteiros ou reais?

Média de 3 números

```
1  def media(a: int, b: int, c: int) -> float:
2  |    m = (a + b + c)/3
3  |    return m
4
5  print ( media(1, 5, 7) )
```

Exemplo de uma função que recebe 3 inteiros e devolve um número **real**.

Note que cada variável tem um tipo de dado. Esse tipo é desejável para melhor compreensão.

Tipos de dados em Python

Os tipos de dados que usaremos são os **primitivos**:

Name	Type	Description
Integers	int	Whole numbers, such as: 3 300 200
Floating point	float	Numbers with a decimal point: 2.3 4.6 100.0
Strings	str	Ordered sequence of characters: "hello" 'Sammy' "2000" "楽しい"
Lists	list	Ordered sequence of objects: [10,"hello",200.3]
Booleans	bool	Logical value indicating True or False

Funções matemáticas em Python

```
1  import math
2
3  def imprimir(n: float):
4      print("Valor absoluto", abs(n))
5      print("Exponenciação ", pow(n, 2))
6      print("Raiz quadrada ", math.sqrt(n))
7      print("Seno          ", math.sin(n))
8      print("Pi           ", math.pi)
9      print("Pi           ", math.e)
10
11  imprimir(float(input()))
```

Funções matemáticas em Python

```
2
Valor absoluto 2.0
Exponenciação 4.0
Raiz quadrada 1.4142135623730951
Seno 0.9092974268256817
Pi 3.141592653589793
Pi 2.718281828459045
```

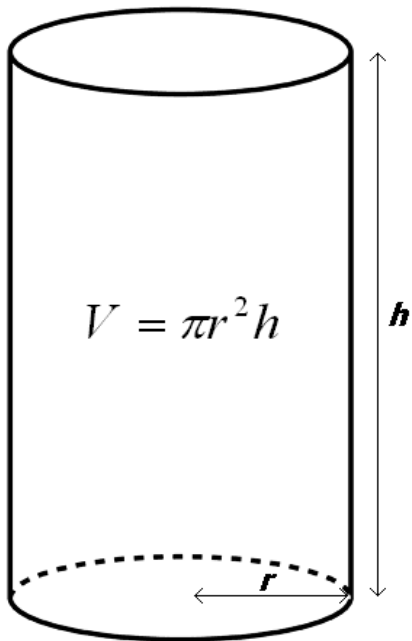
Funções matemáticas em Python

```
-2
Valor absoluto 2.0
Exponenciação 4.0
Traceback (most recent call last):
  File "main.py", line 11, in <module>
    imprimir(float(input()))
  File "main.py", line 6, in imprimir
    print("Raiz quadrada ", math.sqrt(n))
ValueError: math domain error
```

Volume de um cilindro

Crie uma **função**, em Python, que calcule o volume de um cilindro com raio **r** e altura **h**.

Assinatura: `def volume(r: float, h: float) -> float`



Volume de um cilindro

```
1 import math
2
3 def volume(r: float, h: float) -> float:
4     | | return math.pi * r**2 * h
5
6 print(volume(2, 10))
```

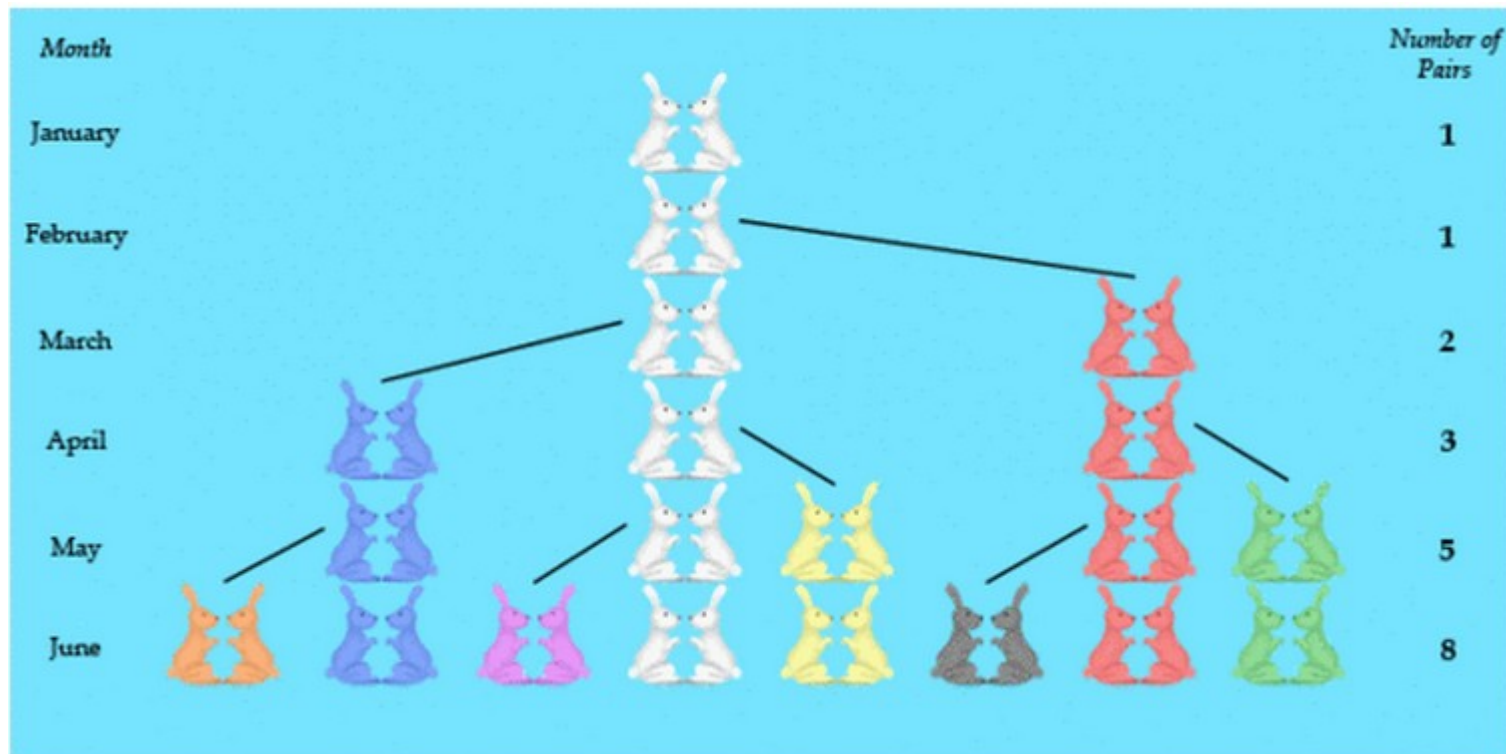
```
125.66370614359172
```



Fibonacci

Números de Fibonacci

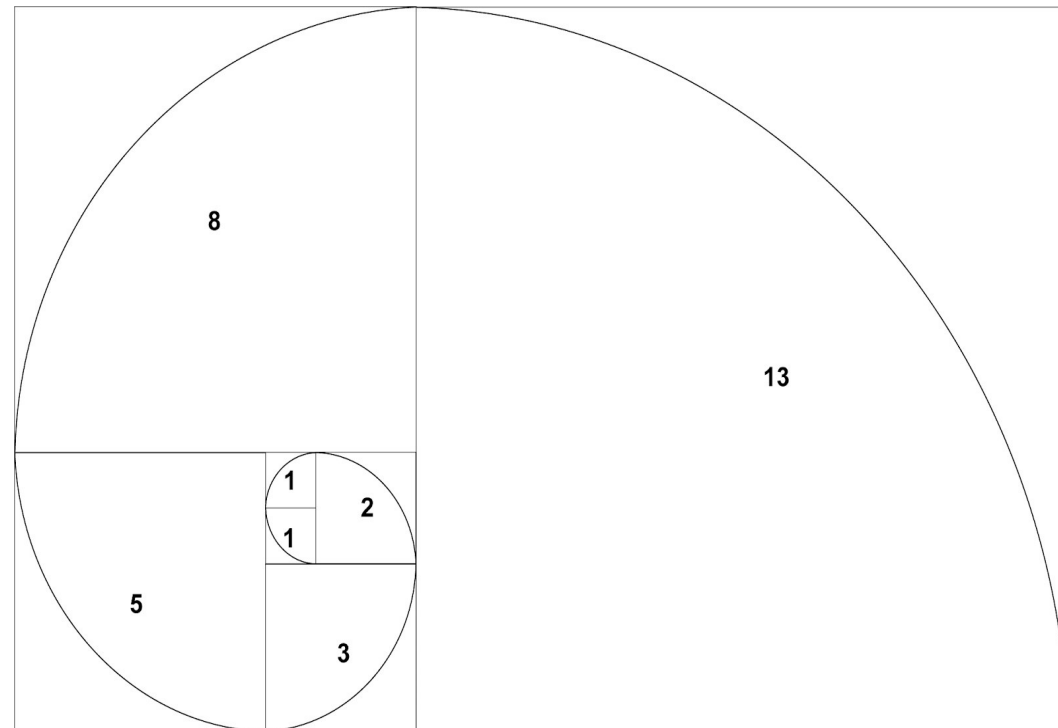
Os números de Fibonacci foram propostos por Leonardo di Pisa (Fibonacci), em 1202, como uma solução para o problema de determinar o tamanho da população de coelhos.



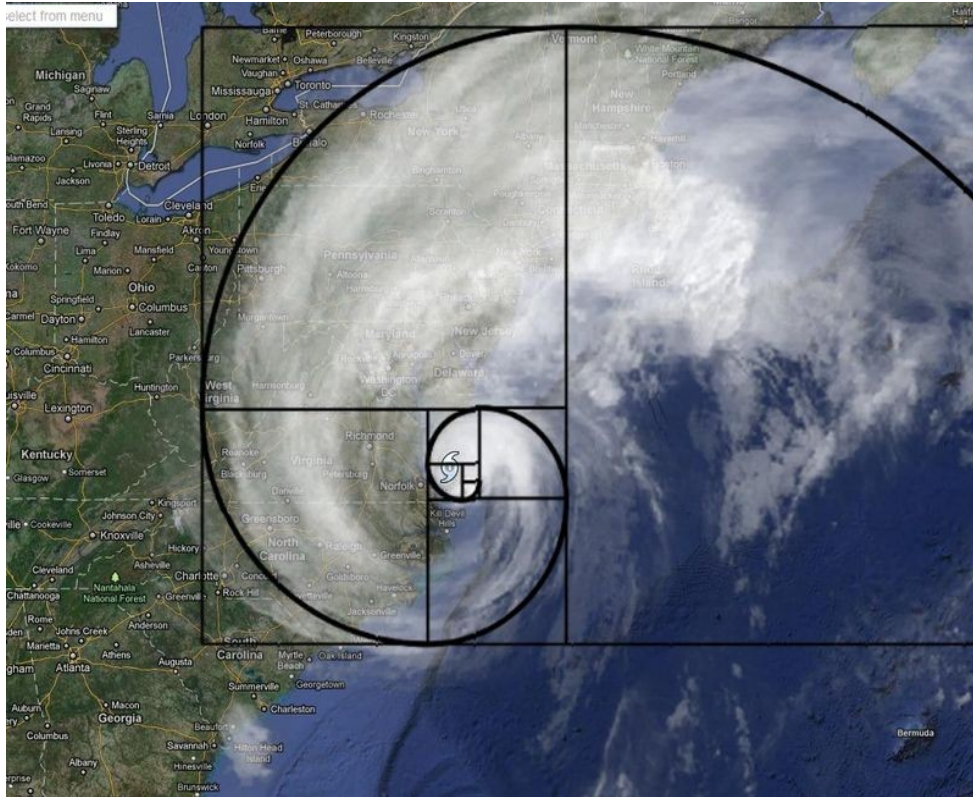
(*) fonte <http://www.oxfordmathcenter.com/drupal7/node/487>

Números de Fibonacci

F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}	F_{17}	F_{18}	F_{19}	F_{20}
0	1	1	2	3	5	8	13	21	34	55	89	144	233	377	610	987	1597	2584	4181	6765



Números de Fibonacci



<https://www.youtube.com/watch?v=SjSHVDfXHQ4>

Números de Fibonacci



(*) fonte <http://britton.disted.camosun.bc.ca/fibslide/jbfibslide.htm>

Números de Fibonacci

Os números de Fibonacci estão relacionados com a razão aurea .

O i-ésimo número pode ser aproximado pela seguinte equação (formula explícita):

$$F_i = \left\lfloor \frac{\left(\frac{1+\sqrt{5}}{2}\right)^i - \left(\frac{1-\sqrt{5}}{2}\right)^i}{\sqrt{5}} \right\rfloor$$

Crie uma função em Python que receba um número inteiro i, e devolva Fi.

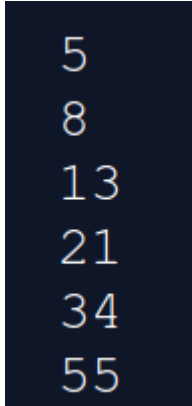
Assinatura: `def fibonacci(i: int) -> int:`

Números de Fibonacci

```
1  import math
2
3  def fibonacci(i: int) -> int:
4      r5 = math.sqrt(5)
5      t1 = math.pow((1+r5)/2, i)
6      t2 = math.pow((1-r5)/2, i)
7
8      return math.floor((t1-t2)/r5)
```

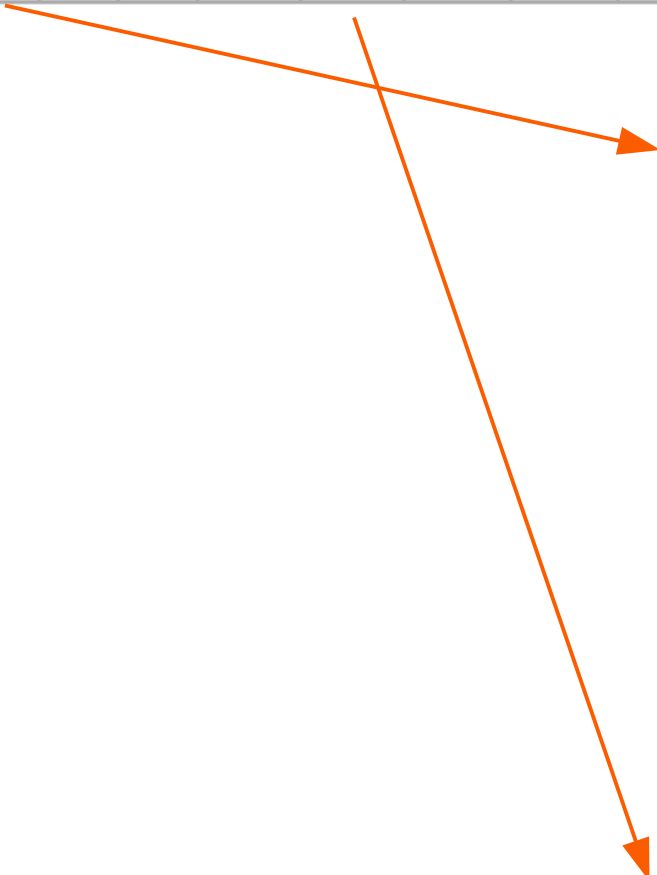
Números de Fibonacci

```
1  import math
2
3  def fibonacci(i: int) -> int:
4      r5 = math.sqrt(5)
5      t1 = math.pow((1+r5)/2, i)
6      t2 = math.pow((1-r5)/2, i)
7
8      return math.floor((t1-t2)/r5)
9
10 print(fibonacci(5))
11 print(fibonacci(6))
12 print(fibonacci(7))
13 print(fibonacci(8))
14 print(fibonacci(9))
15 print(fibonacci(10))
```



5
8
13
21
34
55

F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}	F_{17}	F_{18}	F_{19}	F_{20}
0	1	1	2	3	5	8	13	21	34	55	89	144	233	377	610	987	1597	2584	4181	6765


$$13 \div 8 = 1.625$$

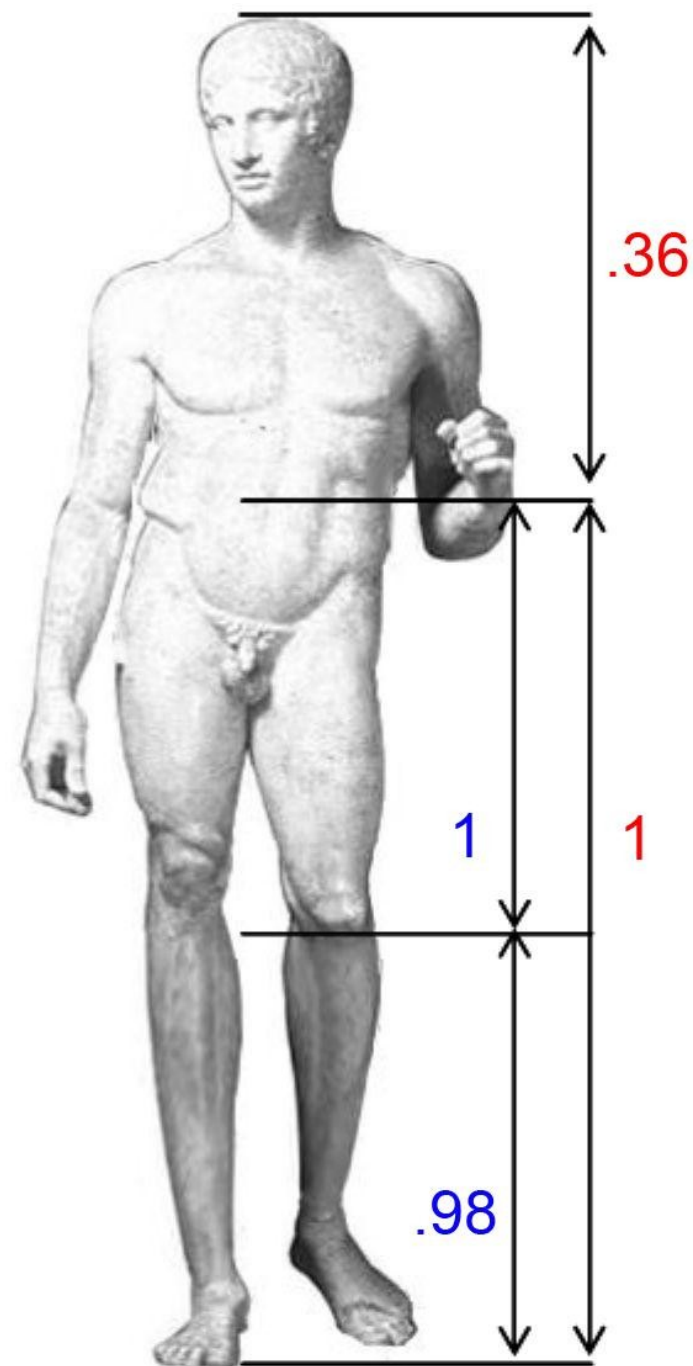
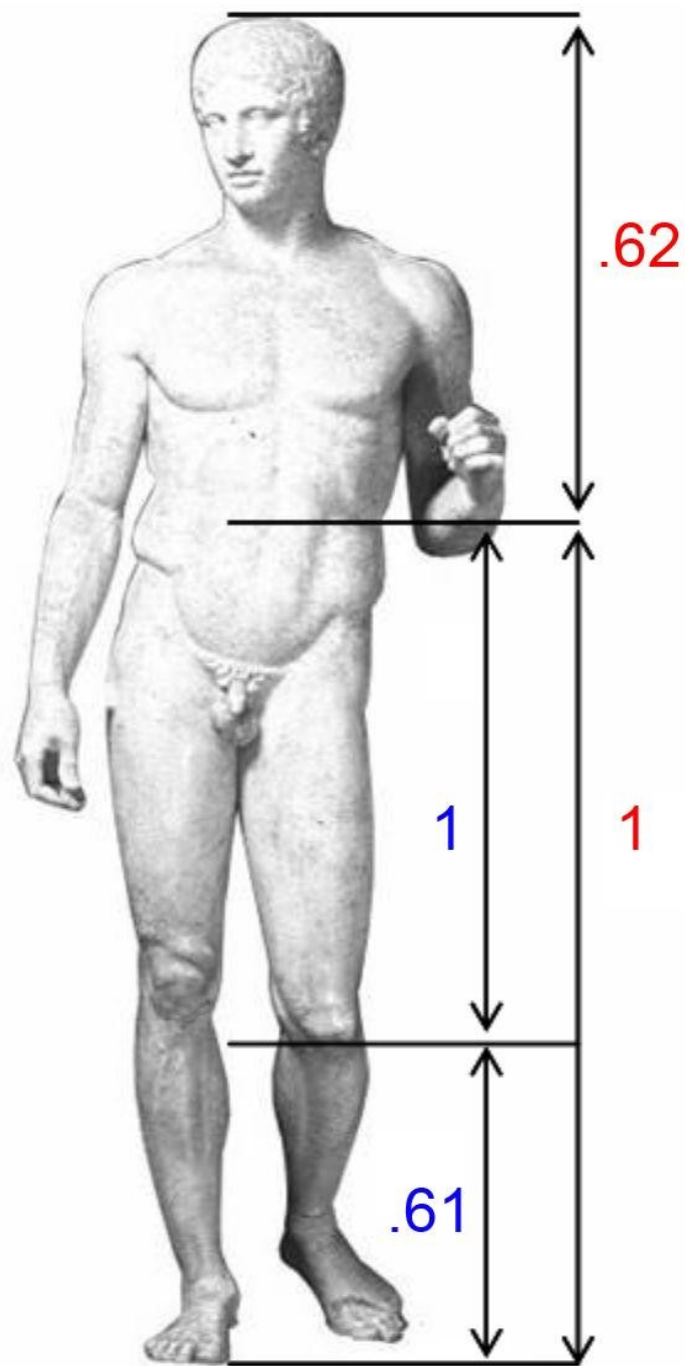
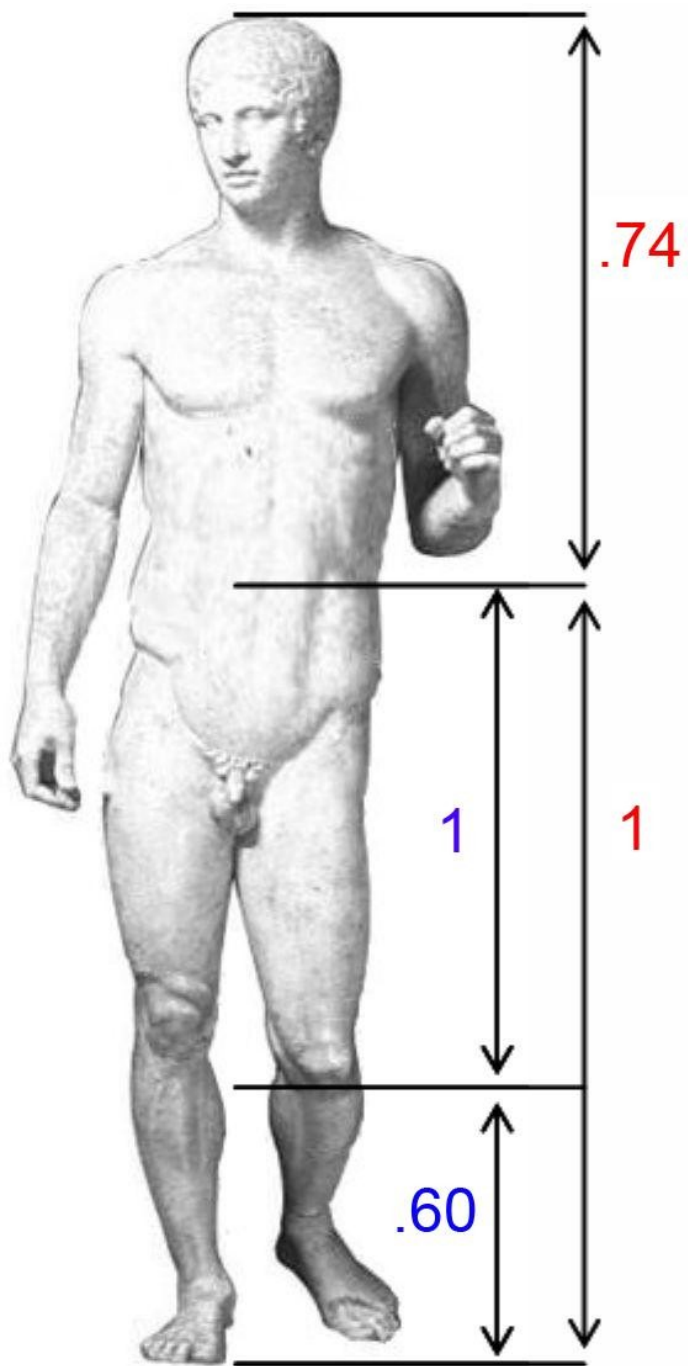
$$21 \div 13 = 1.615\dots$$

$$34 \div 21 = 1.619\dots$$

$$55 \div 34 = 1.6176\dots$$

$$89 \div 55 = 1.61818\dots$$

Golden ratio





Estruturas de seleção – parte 1

Execução condicional

Para escrever programas úteis, quase sempre precisamos da possibilidade de verificar condições e mudar o comportamento do programa.

Instruções/estruturas condicionais nos dão essa habilidade. A forma mais simples é o `if`:

```
if (x>0) :  
    Instrucao1  
    Instrucao2  
    . . .
```

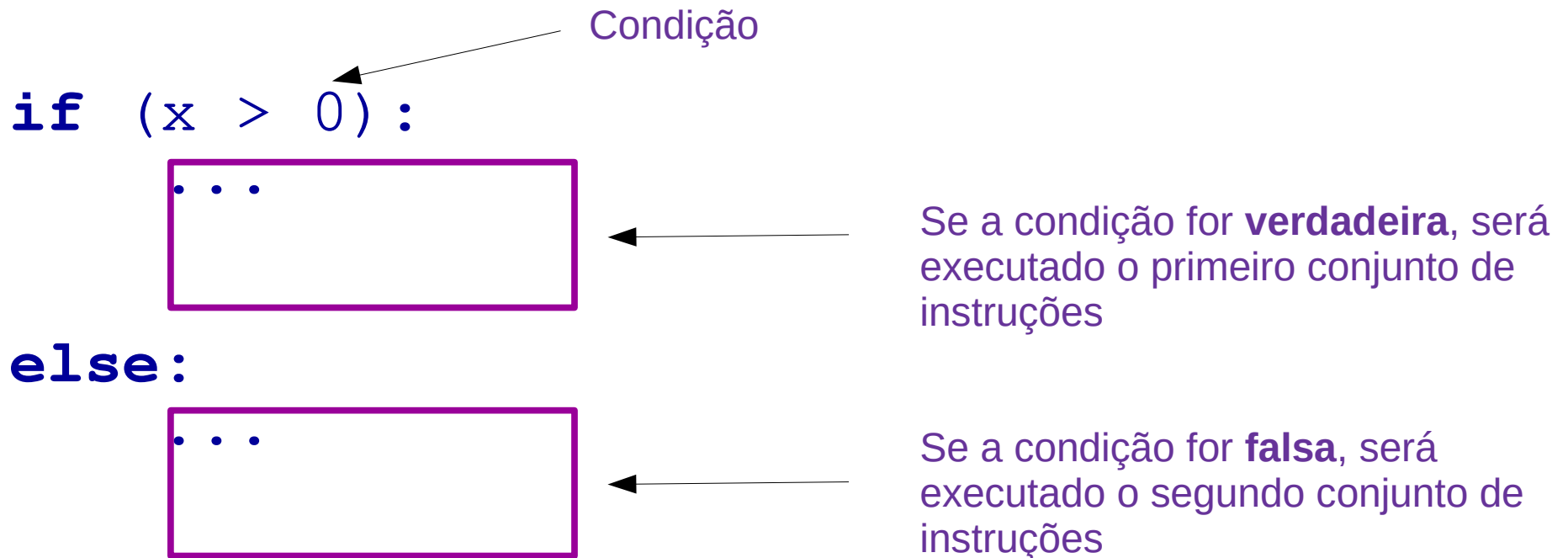
Condição

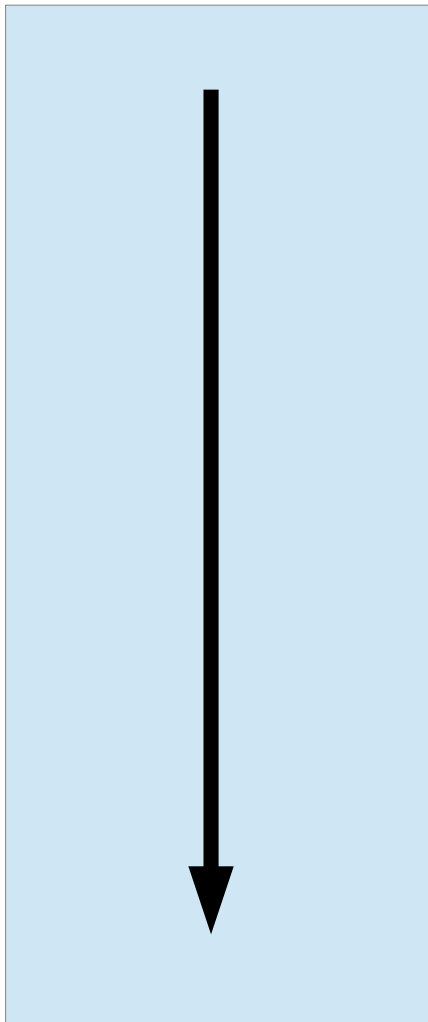
Se a condição for **verdade**, então as instruções no bloco de instruções são executadas.

O bloco deve conter a mesma indentação.

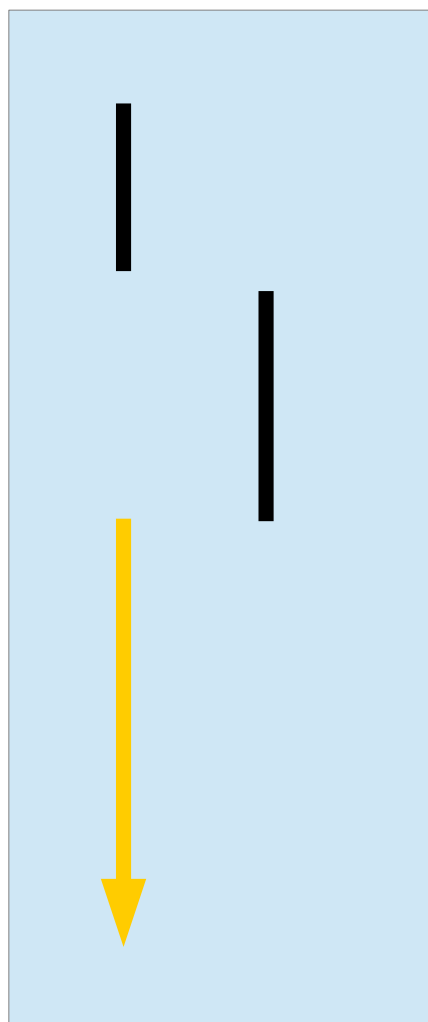
Execução condicional

A segunda forma da instrução `if` é a execução alternativa, na qual existem duas possibilidades e a condição determina qual delas será executada.

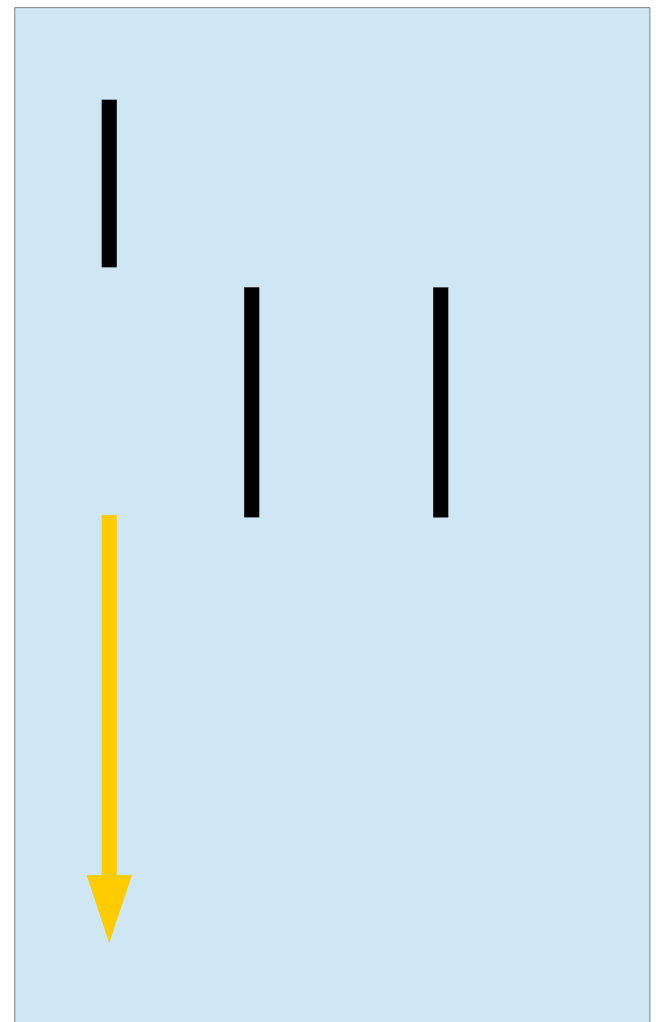




Processamento
sequencial



Seleção simples



Seleção composta

Maior valor entre 2 números

```
1  def maior(x: float, y: float) -> float:
2      |   |   |
3      |   |   |   |   |   |
4      |   |   |   |   |   |
5      |   |   |   |   |   |
6      |   |   |   |   |   |
7      |   |   |   |   |   |
8      |   |   |   |   |   |
9  print(maior(10, 20))
10 print(maior(3.14, 2.81))
```

```
20
3.14
```

As duas funções realizam a mesma operação

```
def maior(x: float, y: float) -> float:  
    if x>y:  
        maior = x  
    else:  
        maior = y  
  
    return maior
```

```
def maior(x: float, y: float) -> float:  
    if x>y:  
        return x  
    else:  
        return y
```

Maior valor entre 3 números

Como usar a função anterior para obter o maior valor entre **3 números**?

Maior valor entre 3 números

Diferentes possibilidades

```
maior( maior(a, b), c)
```

```
maior( a, maior(b, c) )
```



Atividade em aula

Maior valor entre 4 números

Crie um método que calcule o maior valor dados 4 números inteiros.

Assinatura:

```
def maior4(a: float, b: float, c:float, d:float) -> float:
```

Maior valor entre 4 números

```
def maior(a: float, b: float, c:float, d:float) -> float:  
    maior = a  
    if maior<b:  
        maior = b  
    if maior<c:  
        maior = c  
    if maior<d:  
        maior = d  
    return maior
```

Menor valor entre 4 números

```
def f1(a: float, b: float, c:float, d:float) -> float:
    m1=a
    m2=c;
    if m1>b:
        m1 = b
    if m2>d:
        m2 = d
    if m1>m2:
        return m2
    else:
        return m1
```

```
print(f1(4, -50, 6, -34))
```

-50

Fatorial de um número inteiro

```
def fff(n: int) -> int:  
    if n==0:  
        return 1  
    else:  
        return fff(n-1)*n
```

```
print(fff(5))
```