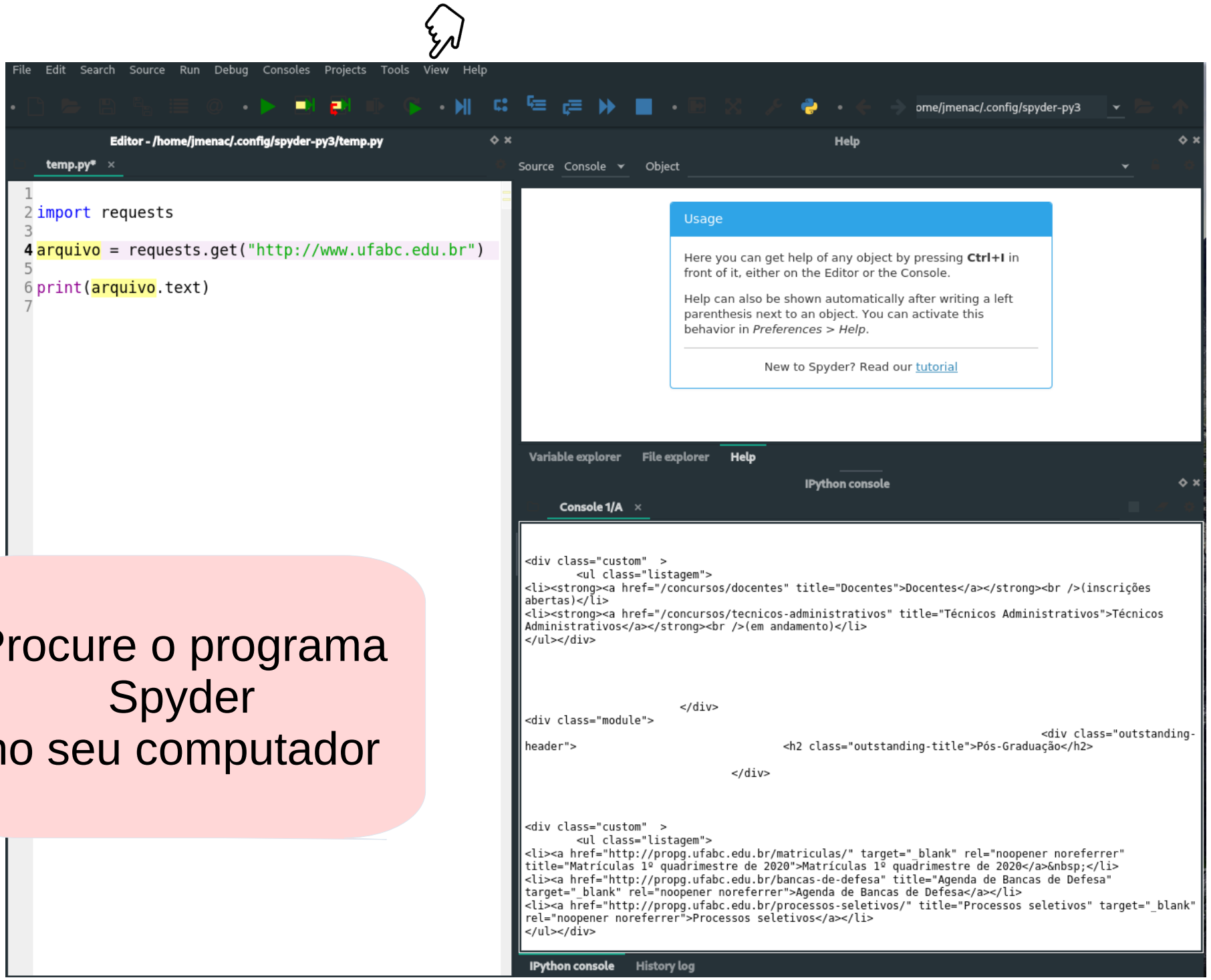
Processamento da Informação

Estruturas de seleção - Parte 2
Estruturas de repetição - Parte 1

Prof. Jesús P. Mena-Chalco
CMCC/UFABC

Q1/2020



Procure o programa Spyder
no seu computador

Exercício sobre seleção: Gerações

Tipos de gerações: baseadas no ano de nascimento

Crie um programa que faça a leitura de um ano (entre 1940 e 2018) e imprima na tela o tipo de geração ao qual o ano está relacionado.

Considere apenas os seguintes 5 tipos:

- 1940-1960: Geracao Baby Boomers
- 1961-1980: Geracao X
- 1981-1997: Geracao Y
- 1998-2009: Geracao Z
- 2010-hoje: Geracao Alfa

Obs: Se não estiver em nenhuma dessa geração deve de imprimir “Não classificado”

Tipos de gerações: baseadas no ano de nascimento

Vamos usar funções, veja a parte inicial do programa

```
def geracao(ano: int) -> str:
    TO-DO

ano = int(input())
print( geracao(ano) )
```

```python
def geracao(ano: int) -> str:
    if ano>=2010:
        return "Geração Alfa"
    elif ano>=1998:
        return "Geração Z"
    elif ano>=1981:
        return "Geração Y"
    elif ano>=1961:
        return "Geração X"
    elif ano>=1940:
        return "Geração Baby boomers"
    else:
        return "Não classificado"

ano = int(input())
print( geracao(ano) )
```

Exercício sobre repetição: Fatorial

Fatorial de um número

Crie uma função que permite calcular o fatorial de um número inteiro:

0! = 1
1! = 1
2! = 2
3! = 6
4! = 24

Assinatura:

```
def fatorial(n: int) -> int:
```

Fatorial de um número

```
def fatorial(n: int) -> int:
    produto = 1
    while n>0:
        produto = produto*n
        n = n-1
    return produto

print (fatorial(5))
```

Teste de mesa: https://donkirkby.github.io/live-py-plugin/demo/

```python
def fatorial(n: int) -> int:
    produto = 1
    while n>0:
        produto = produto*n
        n = n-1
    return produto

print (fatorial(5))
```

Teste com valores Diferentes de 5

```
n = 5
produto = 1
                |                 |                  |                  |
produto = 5 | produto = 20 | produto = 60 | produto = 120 | produto = 120
n = 4       | n = 3        | n = 2        | n = 1         | n = 0
return 120

print('120')
```

Note como a variável **n** atinge o valor **zero**.
Note como o fatorial é **‘construído’**.

Teste de mesa: https://donkirkby.github.io/live-py-plugin/demo/

```python
def fatorial(n: int) -> int:
    produto = 1
    while n>0:
        produto = produto*n
        n = n-1
    return produto

print (fatorial(15))
```

```
n = 15
produto = 1
                |                 |                 |
produto = 15 | produto = 210 | produto = 2730 | produto = 32760
n = 14       | n = 13        | n = 12         | n = 11              ...
return 1307674368000

print('1307674368000')
```

Exercício sobre repetição: Aproximação de *e*

Aproximação de *e*

O número de Euler $e = \sum_{n=0}^{\infty} \frac{1}{n!} = 2.71828182845904\ldots$

$$e = \underbrace{\frac{1}{0!}}_{\text{Termo 1}} + \underbrace{\frac{1}{1!}}_{\text{Termo 2}} + \underbrace{\frac{1}{2!}}_{\text{Termo 3}} + \underbrace{\frac{1}{3!}}_{\text{Termo 4}} + \underbrace{\frac{1}{4!}}_{\text{Termo 5}} + \cdots$$

Implemente essa somatoria mas considerando os **n** primeiros termos, para n>=1.

Assinatura:

```
def euler(termos: int) -> float:
```

```python
def fatorial(n: int) -> int:
    produto = 1
    while n>0:
        produto = produto*n
        n = n-1
    return produto

def euler(termos: int) -> float:
    soma = 0
    i = 0

    while i<termos:
        soma = soma + 1/fatorial(i)
        i = i+1

    return soma

print( euler(5) )
```

Teste de mesa: https://donkirkby.github.io/live-py-plugin/demo/

```
1  def fatorial(n: int) -> int:
2      produto = 1
3      while n>0:
4          produto = produto*n
5          n = n-1
6      return produto
7
8  def euler(termos: int) -> float:
9      soma = 0
10     i = 0
11
12     while i<termos:
13         soma = soma + 1/fatorial(i)
14         i = i+1
15
16     return soma
17
18 print( euler(3) )
```

```
1  n = 0          | n = 1          | n = 2
2  produto = 1    | produto = 1    | produto = 1
3                 |                |                  |
4                 | produto = 1    | produto = 2      | produto = 2
5                 | n = 0          | n = 1            | n = 0
6  return 1       | return 1       | return 2
7
8  termos = 3
9  soma = 0
10 i = 0
11
12                |                |
13 soma = 1.0     | soma = 2.0     | soma = 2.5
14 i = 1          | i = 2          | i = 3
15
16 return 2.5
17
18 print('2.5')
```

Exercício sobre repetição: Aproximação de *e* (versão 2)

Aproximação de *e*

O número de Euler

$$e = \underbrace{\frac{1}{0!}}_{\text{Termo 1}} + \underbrace{\frac{1}{1!}}_{\text{Termo 2}} + \underbrace{\frac{1}{2!}}_{\text{Termo 3}} + \underbrace{\frac{1}{3!}}_{\text{Termo 4}} + \underbrace{\frac{1}{4!}}_{\text{Termo 5}} + \dots$$

Implemente essa nova somatoria mas considerando todos os termos **maiores ou iguais** a 0.0000001

Assinatura:

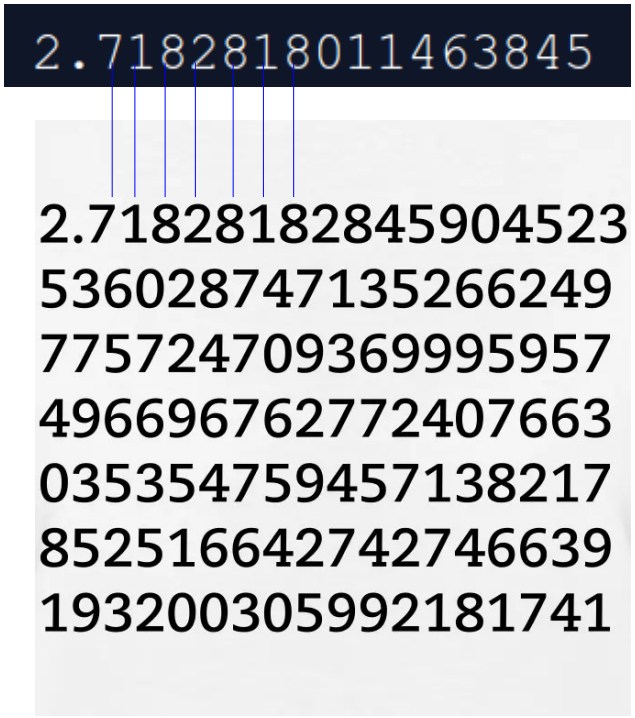
```
def euler() -> float:
```

Desafio:
Quantos termos são **maiores ou iguais** a 0.0000001?

```python
def fatorial(n: int) -> int:
    produto = 1
    while n>0:
        produto = produto*n
        n = n-1
    return produto

def euler() -> float:
    soma = 0
    i = 0

    while 1/fatorial(i)>0.0000001:
        soma = soma + 1/fatorial(i)
        i = i+1

    return soma

print( euler() )
```

Aproximação de *e*

11 termos?



Resposta do programa

Uma boa aproximação de *e*.

Leonhard Paul Euler (/ˈɔɪlər/; pronúncia em alemão: [ˈɔʏlɐ] (🔊 escutar (ajuda·info)), pronúncia local: [ˈɔɪlɐ̯] (🔊 escutar (ajuda·info)); Basileia, 15 de abril de 1707 – São Petersburgo, 18 de setembro de 1783) foi um matemático e físico suíço de língua alemã que passou a maior parte de sua vida na Rússia e na Alemanha.[3] Fez importantes descobertas em várias áreas da matemática como o cálculo e a teoria dos grafos. Também introduziu muitas das terminologias da matemática moderna e da notação matemática, particularmente na análise matemática, como também no conceito de função matemática.[4] É também reconhecido por seus trabalhos na mecânica, dinâmica de fluidos, óptica, astronomia e teoria da música.[5]

Euler é considerado um dos mais proeminentes matemáticos do século XVIII e também é considerado como um dos grandes matemáticos de todos os tempos, assim como Isaac Newton, Arquimedes e Carl Friedrich Gauss.[6] Foi um dos mais prolíficos matemáticos, calcula-se que toda a sua obra reunida teria entre 60 e 80 volumes de quartos.[7] Viveu a maior parte da vida em São Petersburgo, na Rússia, e em Berlim, que na época era capital da Prússia.

Uma declaração atribuída a Pierre-Simon Laplace manifestada sobre Euler na sua influência sobre a matemática: "Leiam Euler, leiam Euler, ele é o mestre de todos nós".[8][9]

Leonhard Euler

Leonhard Euler, quadro a óleo por Johann Georg Brucker.

Conhecido(a) por	Fórmula de Euler, Número de Euler, Característica de Euler, Identidade de Euler, Reta de Euler, Constante de Euler-Mascheroni, Produto de Euler, Diagrama de Euler, Ângulos de Euler, Soma de Euler, Conjetura de Euler, Equação de Euler, Equações de Euler (fluidos), 2002 Euler
Nascimento	15 de abril de 1707 Basileia, Suíça
Morte	18 de setembro de 1783 (76 anos) São Petersburgo, Rússia
Nacionalidade	Suíço
Alma mater	Universidade de Basileia

```
Lista 1:  18/fev   Entrega: 03/mar
Lista 2:  03/mar   Entrega: 09/mar
Lista 3:  10/mar   Entrega: 16/mar
Lista 4:  24/mar   Entrega: 30/mar
Lista 5:  31/mar   Entrega: 06/abr
Lista 6:  07/abr   Entrega: 13/abr
Lista 7:  14/abr   Entrega: 27/mar
Lista 8:  28/abr   Entrega: 12/mai
```

Lista 2:

- **8** exercícios obrigatórios + **1** opcional
- Entrega: 09/março (23h59)

Lista2_1 - Bart Simpson

Entrada:

O seu programa deve receber um número natural **N** e uma frase **F**.

Saída:

Seu programa deve imprimir **N** vezes a frase **F**.

Exemplos:

Entrada	Saída
3 Nao desperdicarei giz	Nao desperdicarei giz Nao desperdicarei giz Nao desperdicarei giz
4 Nao estimularei uma revolucao	Nao estimularei uma revolucao Nao estimularei uma revolucao Nao estimularei uma revolucao Nao estimularei uma revolucao
5 Peixe nao gosta de cafe	Peixe nao gosta de cafe Peixe nao gosta de cafe Peixe nao gosta de cafe Peixe nao gosta de cafe Peixe nao gosta de cafe

Lista2_2 - Tabuada

Entrada:

O seu programa receberá um inteiro positivo **N** e um inteiro positivo **M**.

Saída:

O seu programa deve imprimir a tabela de multiplicação do número **N**, começando de 0 e terminando em **M**.

Exemplos:

Entrada	Saída
3 5	3 x 0 = 0 3 x 1 = 3 3 x 2 = 6 3 x 3 = 9 3 x 4 = 12 3 x 5 = 15
7 12	7 x 0 = 0 7 x 1 = 7 7 x 2 = 14 7 x 3 = 21 7 x 4 = 28 7 x 5 = 35 7 x 6 = 42 7 x 7 = 49 7 x 8 = 56 7 x 9 = 63 7 x 10 = 70 7 x 11 = 77 7 x 12 = 84

Lista2_3 – Máquina de somar

Entrada:

O seu programa receberá um número inteiro não negativo **N** que denota a quantidade de números que seu programa receberá para computar o valor total da soma. Na sequência seu programa receberá **N** números reais.

Saída:

O seu programa deve imprimir a frase "**Total:** " seguida do valor da soma dos **N** números reais (com duas casas decimais de precisão).

Exemplos:

Entrada	Saída
3 7.00 1.50 2.25	Total: 10.75
5 1.25 5.70 2.56 9.99 3.00	Total: 22.50

Lista2_4 –Contagem regressiva maluca

Entrada:

O seu programa deve receber um número inteiro positivo **N**, que é o tempo inicial do temporizador.

Saída:

Seu deve escrever a saída conforme os exemplos abaixo.

Exemplos:

Entrada	Saída
10	Faltam 10 segundos Faltam 8 segundos Faltam 4 segundos Acabou
50	Faltam 50 segundos Faltam 48 segundos Faltam 44 segundos Faltam 38 segundos Faltam 30 segundos Faltam 20 segundos Faltam 8 segundos Acabou