

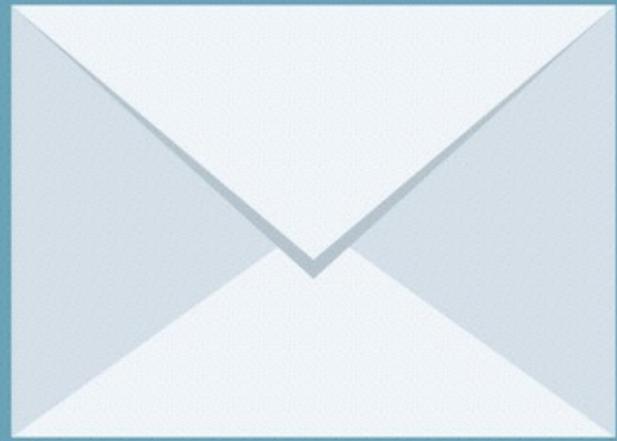
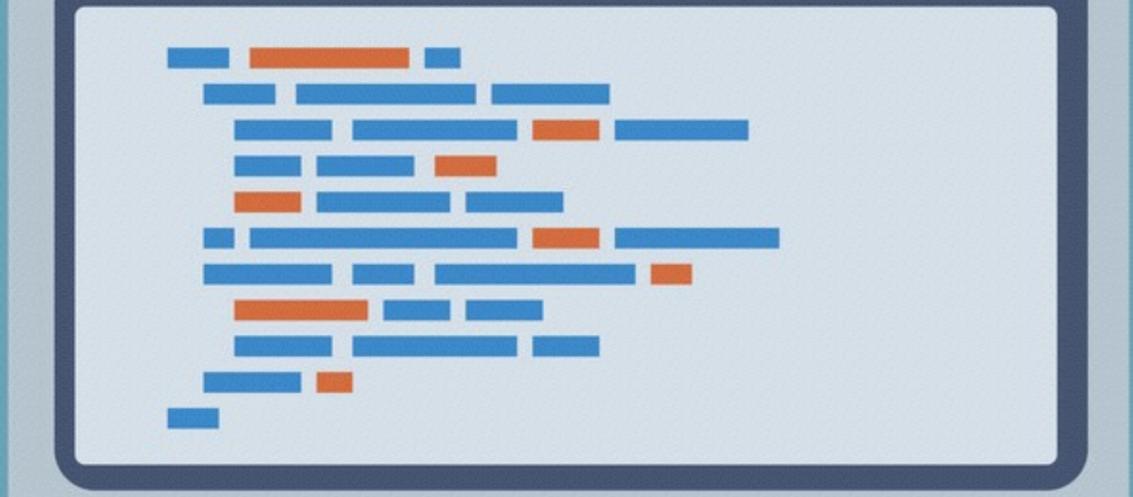
CK0211 - Fundamentos de Programação: Prática em Estrutura de Repetição

Emanuele Santos

Bibliografia: Ascencio, Cap. 5

Objetivos

- Praticar as estruturas de repetição em Python
- Os exercícios devem ser iniciados no laboratório e enviados ao final da aula pelo SIGAA.



EXERCÍCIOS

Por onde começar?

- No Linux:
 - Usando o navegador de arquivos, vá para a sua pasta fup (caso não tenha, crie uma pasta no seu diretório de usuário para guardar os arquivos da disciplina)
 - Abra o editor de texto Sublime
 - Abra uma janela do terminal e vá para sua pasta fup

```
cd fup
```

Relembrando...

- Qual o comando para mostrar saída para o usuário?

```
print(...)
```

- Qual o comando para pedir entrada de dados do usuário?

```
input(...)
```

- Como fazer comentários no código em Python?

```
#comentário
```

- Como converter a entrada de dados para inteiro?

```
int(input(...))
```

- Como converter a entrada de dados para real?

```
float(input(...))
```

Relembrando: Estrutura Condicional

- Usada para verificar se uma condição é atendida:

```
if condição :  
    comandos  
else:  
    outros comandos
```

```
if condição1 :  
    comandos1  
elif condição2 :  
    comandos2  
elif condição3 :  
    comandos3  
else:  
    comandos4
```

Relembrando: Estrutura de Repetição

- Usada para repetir comandos

```
while condição :  
    comandos
```

```
for variável in range(início=0, fim, passo=1):  
    comandos
```

Exercício 1

Ascencio, 3ed, p. 117

- Um funcionário de uma empresa recebe, anualmente, aumento salarial. Sabe-se que:
 - a) Esse funcionário foi contratado em 2005, com salário inicial de R\$ 1.000,00.
 - b) Em 2006 ele recebeu aumento de 1,5% sobre o seu salário inicial.
 - c) A partir de 2007 (inclusive), os aumentos salariais sempre corresponderam ao dobro do percentual do ano anterior
- Faça um programa que determine o salário atual desse funcionário. O algoritmo desse programa está no próximo slide.

```
$> python3 salario.py
```

Exercício 1 - algoritmo

Ascencio, 3ed, p. 117

ALGORITMO

```
DECLARE i, ano_atual, salario, percentual NUMÉRICO  
LEIA ano_atual  
salario ← 1000.00  
percentual ← 1.5/100  
salario ← salario + percentual * salario  
PARA i ← 2007 ATÉ ano_atual FAÇA  
INÍCIO  
    percentual ← 2 * percentual  
    salario ← salario + percentual * salario  
FIM  
ESCREVA salario  
FIM_ALGORITMO.
```

Exercício 1 (cont.)

Ascencio, 3ed, p. 117

- Implemente o algoritmo do slide anterior como `aumento_salario_for.py`
- Depois implemente o mesmo programa, usando o comando `while`. Salve o seu arquivo como `aumento_salario_while.py`
- Teste o seu programa entrando com o ano atual 2016. Os dois programas devem mostrar a mesma saída.

```
$> python3 aumento_salario_for.py
```

```
$> python3 aumento_salario_while.py
```

Exercício 2

- Uma pessoa gostaria de fazer uma viagem que custa uma quantia **V** em reais e precisa de ajuda com o planejamento. A pessoa está disposta a fazer uma poupança iniciando em **I** reais e irá depositar mensalmente uma quantia **M** na poupança. Sabendo que a poupança rende 0,5% ao mês, faça um programa que receba esses dados do usuário e estime em quantos meses ele conseguiria o dinheiro para viajar.
- Salve o seu programa como **financa.py**

Exercício 3-a

- Na última aula vimos o algoritmo para calcular o fatorial de um número inteiro positivo digitado pelo usuário.
- Certifique-se de que o número é mesmo maior que 0.
- Implemente esse algoritmo e salve o seu arquivo como `fatorial.py`

```
$> python3 fatorial.py  
Digite um número inteiro positivo: 4  
Fatorial de 4: 24
```

Exercício 3-b

- Modifique o seu programa, tal que o programa possa calcular o fatorial de vários números em uma mesma execução. O usuário deve digitar um número menor ou igual a zero para interromper a execução. Salve o seu programa como **fatorial2.py**
- Dica: você vai precisar de dois laços aninhados para implementar esse programa.
- Um exemplo de execução do programa é mostrado a seguir:

```
$> python3 fatorial2.py
Digite um número inteiro positivo: 4
Fatorial de 4: 24
Digite um número inteiro positivo: 5
Fatorial de 4: 120
Digite um número inteiro positivo: 0
Fim.
```

Exercício 4

- Faça um programa que receba dez números inteiros, calcule e mostre a soma dos números pares e a soma dos números primos. Um número é primo quando é divisível apenas por 1 e por ele mesmo. Salve o seu programa como **numeros.py**

Exercício 5

- Faça um programa que receba o número de alunos de uma turma, e para cada aluno receba duas notas, calcule e mostre:
 - A média das duas notas, sabendo-se que a segunda nota possui peso 2
 - O resultado do aluno: Aprovado (se $M \geq 7$), Reprovado (se $M < 4$) e AF ($4 \leq M < 7$).
- Ao final, mostre:
 - Os totais de alunos aprovados, reprovados e que ficaram de AF e a média da turma
- Observação: se você já fez o programa `media` na última aula de laboratório, pode aproveitar parte do código, mas salve o novo programa como **`media2.py`**

Exercício 6-a

- Jogo: adivinhe o número amador
- Faça um programa que escolha um número aleatório entre 1 e 10 e peça para o usuário adivinhá-lo. Permita tantas tentativas quanto forem necessárias para o usuário acertar o número. Mostre dicas ao usuário, quando ele errar: tente um número maior ou tente um número menor. Ao final diga em quantas tentativas o usuário acertou o número. Salve o programa como `adivinha_amador.py`
- Para a escolha do número aleatório entre x e y , use a função `randint(x,y)` da biblioteca `random`, como mostrado a seguir:

```
import random #biblioteca para números aleatórios  
#escolher um número inteiro de 1 a 10  
n = random.randint(1, 10)
```

Exercício 6-b

- Jogo: adivinhe o número profissional
- Modifique o programa do jogo anterior, tal que agora você mostre uma lista de níveis para o jogador escolher, como mostrado a seguir:

```
$> python3 adivinhe_prof.py
Adivinhe o número! Escolha o nível
1 - Beginner
2 - Hard
3 - Insane
Digite a sua opção:
```

- O nível Beginner é o mesmo programa anterior; o nível hard limita o número de tentativas em 3, ainda mostrando as dicas; e o nível insane também limita o número de tentativas em 3, mas sem dicas.