

# **PYTHON - SELEÇÃO E REPETIÇÃO**

# Objetivos

- Aprender como construir e utilizar as estruturas de seleção e repetição do Python

# Comando de seleção simples em Python

- Analisam e executam uma única vez um bloco de instruções de acordo com o valor de uma condição.
- Podem ser SIMPLES E COMPOSTO
- SINTAXE DO COMANDO DE SELEÇÃO SIMPLES

if CONDIÇÃO:

COMANDOS1

Exemplo:

*if a > b:*

*print a*

# Programa

- Fazer um programa para ler um número inteiro e imprimir a palavra ÍMPAR se o número for impar

```
n = input('N: ')\nif n % 2 == 1:\n    print 'ÍMPAR ', n
```

# Comando de seleção composto em Python

- SINTAXE DO COMANDO DE SELEÇÃO COMPOSTO

if CONDIÇÃO:

COMANDOS1

else:

COMANDOS2

Ex:

*if a > b:*

*print a*

*else:*

*print b*

# Programa

- Fazer um programa para ler duas notas e imprimir a sua situação de acordo com os critérios da UERJ

```
n1 = input('Nota 1: ')
n2 = input('Nota 2: ')
media = (n1+n2)/float(2)
if media >= 7:
    print 'APROVADO ', media
else:
    if media >= 4:
        print 'PROVA FINAL ', media
    else:
        print 'REPROVADO ', media
```

# Exercício

- Fazer um algoritmo para ler a idade de uma pessoa e imprimir sua situação de acordo com os critérios abaixo:
  - Idade  $\leq 0$  , ERRO
  - $1 \leq \text{idade} \leq 3$  , BEBE
  - $4 \leq \text{idade} \leq 11$  , CRIANÇA
  - $12 \leq \text{idade} \leq 17$  , TEEN
  - $18 \leq \text{idade} \leq 30$  , JOVEM
  - $31 \leq \text{idade} \leq 64$  , ADULTO
  - idade  $\geq 65$  , SENIOR

# Programa

```
idade = input('Idade: ')
if idade <= 0:
    print 'ERRO ', idade
else:
    if idade <= 3:
        print 'BEBE ', idade
    else:
        if idade <= 11:
            print 'CRIANÇA ', idade
        else:
            if idade <= 17:
                print 'TEEN ', idade
            else:
                if idade <= 30:
                    print 'JOVEM ', idade
                else:
                    if idade <= 64:
                        print 'ADULTO ', idade
                    else:
                        print 'SENIOR ', idade
```



# Sequência de decisões

- Se houver a necessidade de uma sequência de decisões como as do algoritmo anterior podemos utilizar a estrutura composta abaixo:

*if condição :*

*Comandos1*

*elif condição:*

*Comandos2*

*...*

*elif condição:*

*Comandos...*

*else:*

*ComandosN*

- Este tipo de estrutura entra em apenas uma das opções e ignora o restante

# Programa

```
idade = input('Idade: ')
if idade <= 0:
    print 'ERRO ', idade
elif idade <= 3:
    print 'BEBE ', idade
elif idade <= 11:
    print 'CRIANÇA ', idade
elif idade <= 17:
    print 'TEEN ', idade
elif idade <= 30:
    print 'JOVEM ', idade
elif idade <= 64:
    print 'ADULTO ', idade
else:
    print 'SENIOR ', idade
```

# Programa

- Fazer um algoritmo para ler o peso de 8 bolas. Assumindo que todas tem o peso igual e apenas uma de peso DIFERENTE, imprimir a bola de peso DIFERENTE.

```
b1 = input('b1: ')
b2 = input('b2: ')
b3 = input('b3: ')
b4 = input('b4: ')
b5 = input('b5: ')
b6 = input('b6: ')
b7 = input('b7: ')
b8 = input('b8: ')

if b1+b2+b3 == b4+b5+b6:
    if b7 != b1:
        print 'b7 ', b7
    else:
        print 'b8 ', b8
elif b1+b2 == b3+b4:
    if b5 != b1:
        print 'b5 ', b5
    else:
        print 'b6 ', b6
elif b1 == b2:
    if b3 != b1:
        print 'b3 ', b3
    else:
        print 'b4 ', b4
else:
    if b1 != b8:
        print 'b1 ', b1
    else:
        print 'b2 ', b2
```

# Estrutura de repetição

- São usadas para repetir um conjunto de instruções
- O principal comando de repetição do PYTHON é o COMANDO while
  - Repete um conjunto de instruções enquanto uma condição for verdadeira
  - A condição é testada antes de executar os comandos
  - É responsabilidade do programador garantir que a condição irá se tornar falsa durante a execução da repetição

# while

- SINTAXE:  
***while*** **CONDIÇÃO:**  
**COMANDOS**

# Programa

- Fazer um programa em PYTHON para ler o nome, sexo e duas notas dos alunos de uma turma até que seja digitado o nome FIM. Imprimir a média pessoal APENAS das alunas. Imprimir também a média aritmética dos homens.

```
cont = 0
soma = 0
nome = raw_input('nome: ')
while nome != 'fim':
    sexo = raw_input('sexo: ')
    n1 = input('n1: ')
    n2 = input('n2: ')
    mediaPessoal = (n1+n2)/float(2)
    if sexo == 'f':
        print 'media f:', mediaPessoal
    else:
        cont += 1
        soma = soma + mediaPessoal
    nome = raw_input('nome: ')

print 'media homens: ', soma/float(cont)
```

# Programa

- Faça um programa que mostre os números ENTRE 100 e 200 que divididos por 7 dão resto 4

```
num = 101  
while num < 200:  
    if num % 7 == 4:  
        print ' ', num  
    num += 1
```

# Comando *for*

- Quando queremos percorrer os ITENS ou ELEMENTOS de uma lista OU quando sabemos a quantidade de repetições a ser executada podemos utilizar o for

SINTAXE:

***for*** ***variavel*** ***in*** ***objeto\_sequencial***:

***Comando1***

***...***

***Comando n***



# *for* utilizando *range()*

- O comando **range** gera uma lista sequencial de elementos com início, fim e uma razão
- SINTAXE:  
**range(início, fim, passo)**
- É importante ressaltar que o intervalo é **FECHADO NO INÍCIO** e **ABERTO NO FIM**
- O passo é opcional
  - Caso não seja colocado a sequência irá de 1 em 1

# *range()* - exemplos

- Exemplos:

```
>>> range(1, 5)  
[1, 2, 3, 4]
```

```
>>> range(1,1)  
[]
```

```
>>> range(10, 20, 2)  
[10, 12, 14, 16, 18]
```

# Programa

- Faça um programa para somar os números pares entre 1000 e 2000, inclusive.

```
soma = 0  
for i in range(1000, 2001, 2):  
    soma += i  
print 'soma:', soma
```

# Programa

- Faça um programa para somar o FATORIAL dos 5 primeiros números

```
soma = 0  
for i in range(1,6):  
    fat = 1  
    for j in range(1, i+1):  
        fat = fat*j  
    print 'fat: ', fat  
    soma += fat  
print 'soma:', soma
```