



STEM²ID

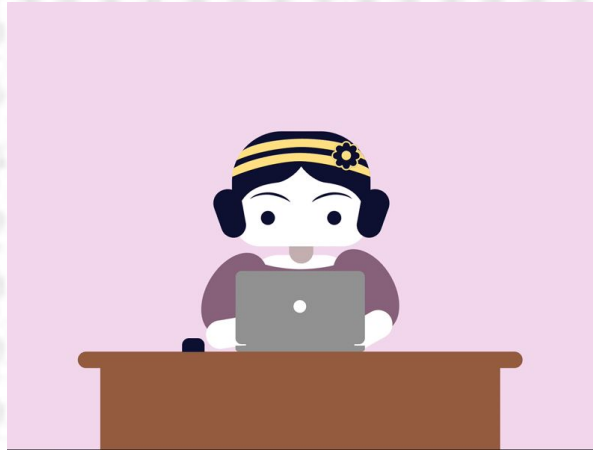
— Go girls! —



like a girl

CONNECTIVOS LÓGICOS

- Os **conectivos lógicos** devem ser entendidos como ferramenta de notação utilizada para unir duas ou mais expressões, e como resultado da união, forma-se uma nova expressão.



CONNECTIVOS LÓGICOS

- Eles são comumente utilizados para definir se certa função ou operação será realizada ou não, dependendo do valor lógico retornado pela expressão.

AND : conectivo de conjunção

OR : conectivo de disjunção

NOT : operador de negação

CONNECTIVOS LÓGICOS

- O resultado dessa nova expressão depende do resultado de cada sub-expressão que a compõe e do operador utilizado.
- **Tabela verdade:**

a	b	NOT a	a AND b	a OR b
V	V	F	V	V
V	F	F	F	V
F	V	V	F	V
F	F	V	F	F

CONECTIVOS LÓGICOS

```
x = 1  
y = 4  
nome = 'STEM'  
(x > 0) and (x < 100)  
True
```

```
(x != 0) and (y >= 4)  
True
```

```
(x + y < 10) and (nome == 'livro') and (y%3 ==  
1)  
False
```

```
(x > 0) or (x < 100)  
True
```

```
(x == 0) or (y > 4)  
False
```

```
(x + y < 10) or (nome == 'livro') or (y%3 == 1)  
True
```

CONECTIVOS LÓGICOS

```
not (x*y%2 == 1)
```

True

```
(x + y < 10) and not (nome == 'STEM') and (y%3 == 1)
```

False

```
(x > 0) and (x < 100) or (x + y < 100)
```

True

```
((nome == 'livro') or (x + y < 100)) and (y >= 4)
```

True

```
((nome == 'livro') and (x + y < 100)) or (y > 4)
```

False

```
((nome == 'STEM') or (y/x < 0)) and ((nome == 'livro') or ((y%3 == 1))
```

False

COMANDOS DE SELEÇÃO

- Na programação, muitas vezes fazemos perguntas sim ou não e decidimos fazer algo com base na resposta.
- Esses tipos de perguntas são chamadas de condições e combinamos essas condições e as respostas em declarações do tipo **if**.

COMANDOS DE SELEÇÃO

- A palavra **if** é um comando que nos permite **realizar um trecho** de código **se** dada **condição** for satisfeita. O uso do **if** dá-se pela seguinte sintaxe:

```
if condicao:  
    fazer alguma coisa
```

- Note que as linhas que seguem a condição apresentada dentro do **if** devem estar em um bloco em que suas linhas devem ser igualmente espaçadas.

COMANDOS DE SELEÇÃO

- A condição compara duas coisas, retorna se a condição é verdadeira ou falsa, baseando-se nisso executa o próximo passo.

```
idade = 15
if idade > 20:
    print('Voce eh mais velho que o meu irmao!')
```

```
idade = 25
if idade > 20:
    print('Voce eh mais velho que o meu irmao!')
```



Code like a girl!

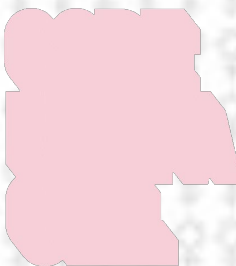


EXERCÍCIO

- Faça um programa que leia da tela três números e diga qual é o maior deles.

Dica: Lembre-se da sintaxe condicional em *Python*:

```
if condicao:  
    fazer alguma coisa
```



EXERCÍCIO

- Faça um programa que pergunte o preço de três produtos e informe qual produto você deve comprar, sabendo que a decisão é sempre pelo mais barato.

COMANDOS DE SELEÇÃO

- Assim como usamos o **if** para fazer algo quando a condição dada é verdadeira, podemos usar expressões condicionais para fazer algo quando a condição é falsa.
- O truque aqui é usar expressões com **if** seguidas de expressões com **else**. O uso do **if** com **else** dá-se pela seguinte sintaxe:

```
if condicao:  
    fazer alguma coisa  
else:  
    fazer outra coisa
```


COMANDOS DE SELEÇÃO

```
print('Voce quer ouvir uma piada?')  
idade = 8  
if idade >= 10:  
    print ('O porco caiu na lama.')  
else:  
    print ('Shh, eh um segredo!')
```

COMANDOS DE SELEÇÃO

- O **elif** é um modo de estender expressões condicionais ainda mais. O uso do **if** com **elif** e **else** dá-se pela seguinte sintaxe:

```
if condicao:  
    fazer alguma coisa  
elif condicao:  
    fazer outra coisa  
else:  
    fazer outra coisa
```

- Você pode utilizar quantos **elif** quiser!
- Cuidado com a indentação!

COMANDOS DE SELEÇÃO

```
idade = 12
if idade == 10:
    print("Porque o pinheiro nao se perde na floresta?")
    print("Porque ele tem uma pinha!")
elif idade == 11:
    print("Sabe como eh a piada do pintinho caipira?")
    print("Pir")
elif idade == 12:
    print("Por que o lapis esta triste?")
    print("Porque ele foi desapontado")
elif idade == 13:
    print("Qual a formula da agua benta?")
    print("H Deus 0")
else:
    print("Huh?")
```




Code like a girl!



EXERCÍCIO

- Faça um Programa que leia um número e exiba o dia correspondente da semana. (1-Domingo, 2- Segunda, etc.), se digitar outro valor deve aparecer valor inválido.

CODE LIKE A GIRL

EXERCÍCIO

- Escreva um programa que faça 5 perguntas para uma pessoa sobre um crime. As perguntas são: “**Telefonou para a vítima?**”, “**Esteve no local do crime?**”, “**Mora perto da vítima?**”, “**Devia para a vítima?**”, “**Já trabalhou com a vítima?**”. O programa deve no final emitir uma classificação sobre a participação da pessoa no crime. Se a pessoa responder positivamente a 2 questões ela deve ser classificada como “**Suspeita**”, entre 3 e 4 como “**Cúmplice**” e 5 como “**Assassino**”. Caso contrário, ele será classificado como “**Inocente**”.

COMANDOS DE SELEÇÃO

- Você pode combinar condições usando as palavras chave **and** (e) e **or** (ou), resultando em códigos menores e mais simples.

```
if idade == 10 or idade == 11 or idade == 12 or idade == 13:  
    print('Ola')  
else:  
    print('Tchau')
```



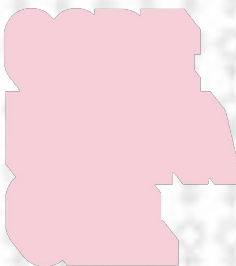
Code like a girl!



EXERCÍCIO

- Escreva um programa que peça um número correspondente a um determinado ano e em seguida informe se este ano é ou não bissexto.

Dica: chama-se ano bissexto o ano ao qual é acrescentado um dia extra, ficando com 366 dias, um dia a mais do que os anos normais de 365 dias, ocorrendo a cada quatro anos, exceto anos múltiplos de 100 que não são múltiplos de 400.



COMANDOS DE REPETIÇÃO

- Imagine que você deseja escrever na tela 5 vezes a frase 'hello', você provavelmente faria algo do tipo:

```
print('hello')  
print('hello')  
print('hello')  
print('hello')  
print('hello')
```

COMANDOS DE REPETIÇÃO

- Para que não seja necessário fazer a mesma coisa repetidamente, usamos uma ferramenta chamada laço.
- O primeiro tipo de laço que aprenderemos a utilizar é o **for**.
O uso do **for** dá-se pela seguinte sintaxe:

```
for variavel in range(<intervalo>):  
    fazer alguma coisa
```

- Cuidado com a indentação!

COMANDOS DE REPETIÇÃO

```
for x in range(0, 5):  
    print('01a')
```

- O grande poder dos comandos de repetição como o **for** é repetir instruções por quantas vezes for necessário.



Code like a girl!

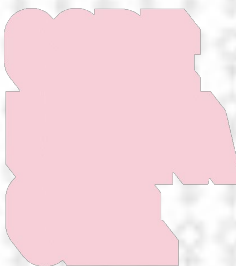


EXERCÍCIO

- Faça um programa que imprima na tela os números de 1 a 20, um abaixo do outro.

Dica: lembre-se da sintaxe do **for**.

```
for variavel in range(<intervalo>):  
    fazer alguma coisa
```



EXERCÍCIO

- Faça um programa que imprima na tela os números múltiplos de 5 presentes entre 1 a 20, um abaixo do outro.

CODE LIKE A GIRL

COMANDOS DE REPETIÇÃO

- Não é obrigatório o uso da função **range** em um **for**, você pode usar algo que você já criou, como em:

```
lista = ['batata', 'carne', 'macarrao']  
for i in lista:  
    print(i)
```

COMANDOS DE REPETIÇÃO

- Um laço de repetição também pode ser formado por outro comando: o **while**. Basicamente esse comando executa os seguintes passos:
 1. Checar a condição;
 2. Se verdadeira, executar o código no bloco;
 3. Repetir até que a condição seja falsa.
- O uso do **while** dá-se pela seguinte sintaxe:

```
while condicao:  
    fazer alguma coisa
```

COMANDOS DE REPETIÇÃO

```
nome = input('Digite seu nome: ')\nwhile nome != 'Maria':\n    print('Voce nao eh a Maria!')\n    nome = input('Digite seu nome: ')
```

- **Importante:** lembrar de mudar a variável da condição para que o laço de repetição não seja executado infinitamente.

COMANDOS DE REPETIÇÃO

- Qual é a diferença entre o **for** e o **while**?
Geralmente um laço com for é executado uma quantidade específica de vezes, enquanto um laço de while é utilizado quando não se sabe a quantidade de repetições necessárias.





Code like a girl!

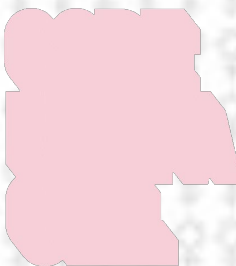


EXERCÍCIO

- Numa eleição existem três candidatos. Faça um programa que requisiite o número total de eleitores. Então, peça para cada eleitor votar e ao final mostre o número de votos de cada candidato.

Dica: lembre-se da sintaxe do laço de repetição **while** em Python:

```
while condicao:  
    fazer alguma coisa
```



EXERCÍCIO

- O Departamento Estadual de Meteorologia lhe contratou para desenvolver um programa que leia as um conjunto indeterminado de temperaturas, e informe ao final a menor e a maior temperaturas informadas, bem como a média das temperaturas.

