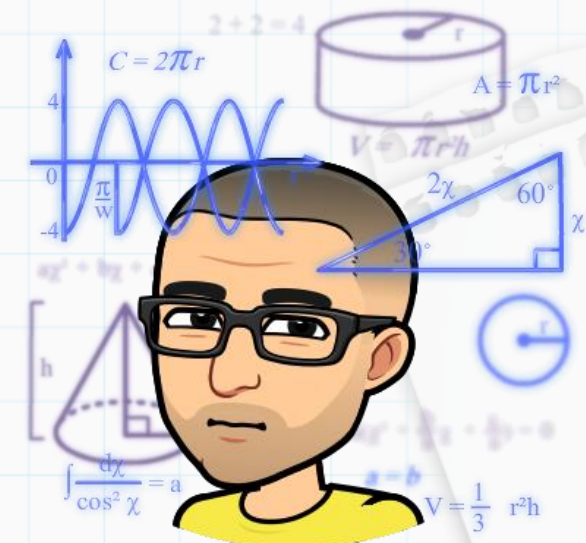
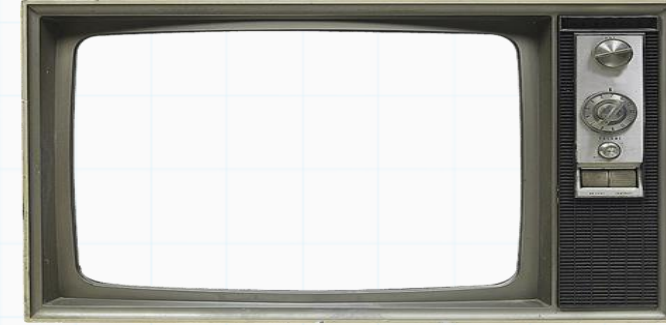


Programação De Computadores

Professor : Yuri Frota

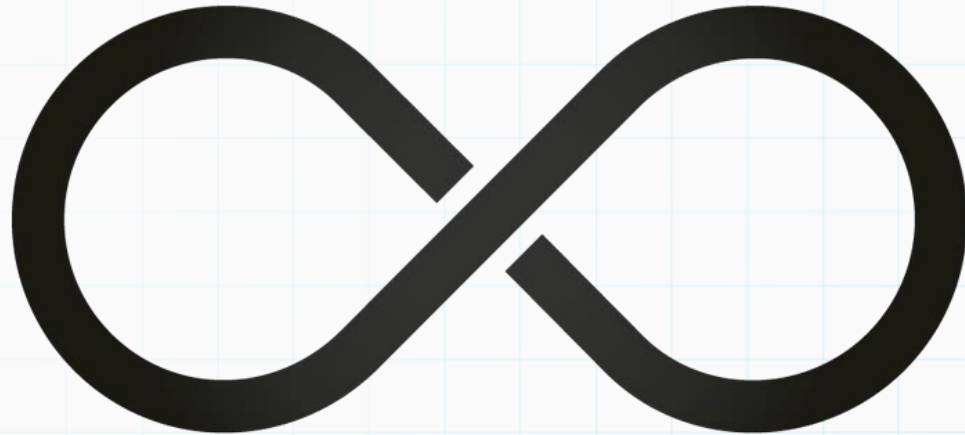
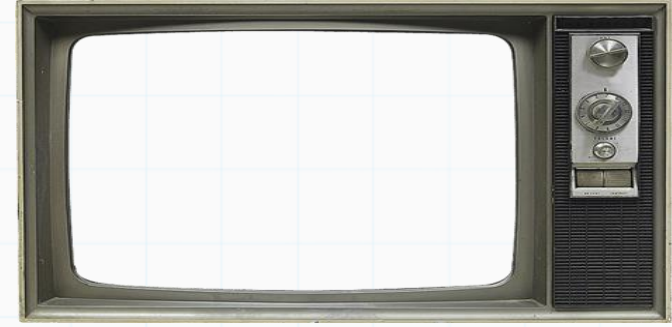
www.ic.uff.br/~yuri/prog.html

yuri@ic.uff.br

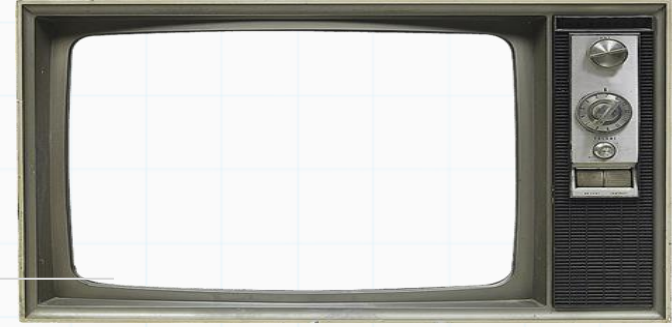
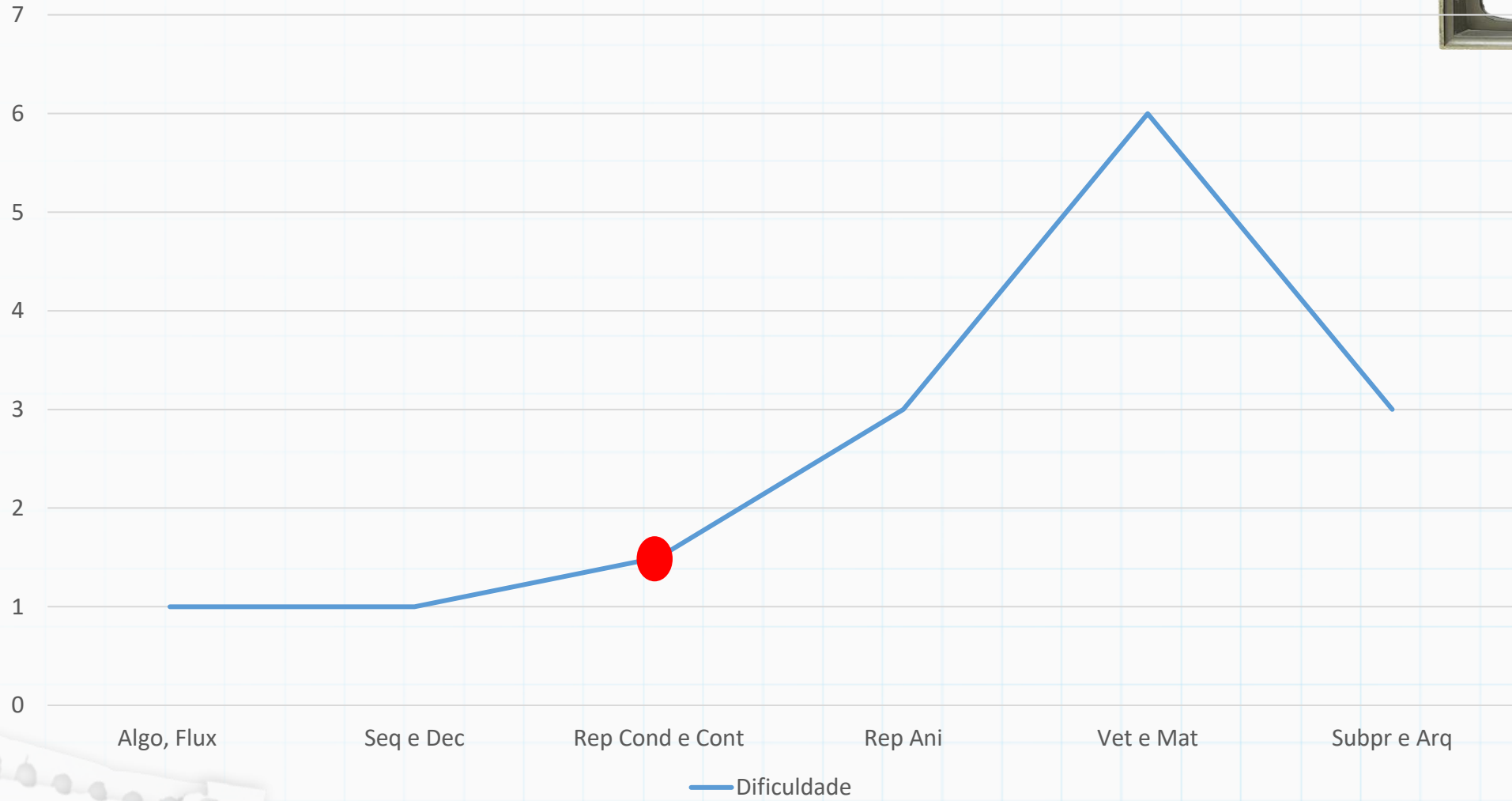


Repetição

Um comando de repetição é aquele que permite repetir um determinado bloco de comandos. Existem dois tipos de repetição: as condicionais e as contáveis.



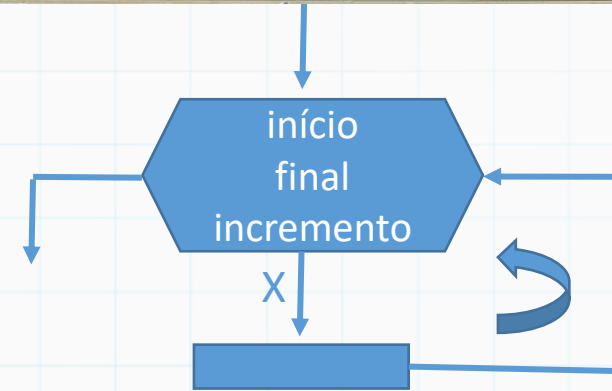
Percepção de Dificuldade dos Alunos



Repetição Contável

Comando For:

Executa o bloco de instruções um número fixo de vezes



X começa de <início>, e a cada iteração do laço, incrementa <incremento>, e executa enquanto X for estritamente menor que <final>

Repetição Contável

Comando For:

Executa o bloco de instruções um número fixo de vezes

Portugol

...

para VAR variando de VALOR
INICIAL a VALOR FINAL com
passo INCREMENTO

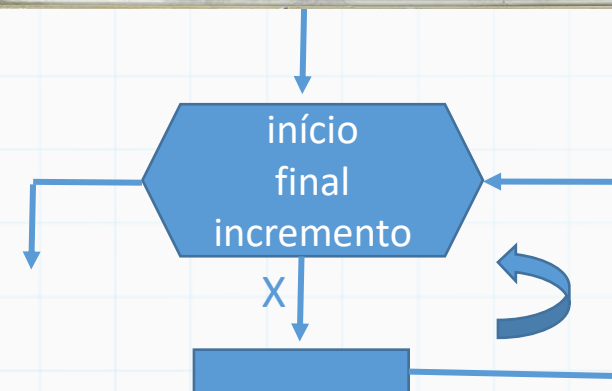
INSTRUÇÃO 1

INSTRUÇÃO 2

...

INSTRUÇÃO N

...



X começa de <início>, e a cada
iteração do laço, incrementa
<incremento>, e executa enquanto X
for estritamente menor que <final>

Repetição Contável

Comando For:

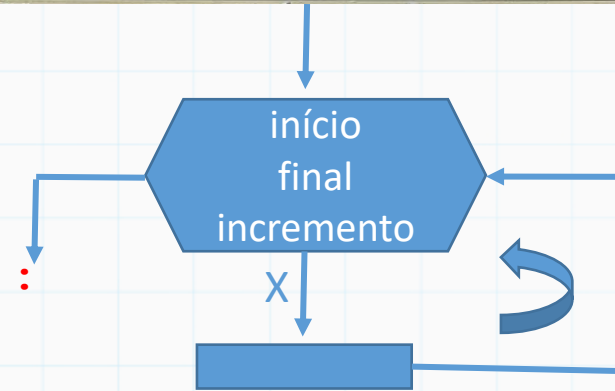
Executa o bloco de instruções um número fixo de vezes

Portugol

```
...  
para VAR variando de VALOR  
INICIAL a VALOR FINAL com  
passo INCREMENTO  
    INSTRUÇÃO 1  
    INSTRUÇÃO 2  
    ...  
    INSTRUÇÃO N  
...
```

Python

```
...  
for VAR in range(<ini>, <fim>, <inc>):  
    INSTRUÇÃO 1  
    INSTRUÇÃO 2  
    ...  
    INSTRUÇÃO N  
...
```



X começa de <início>, e a cada iteração do laço, incrementa <incremento>, e executa enquanto X for estritamente menor que <final>

Repetição Contável

formato

for X in range (<início>, <fim>, <passo>):

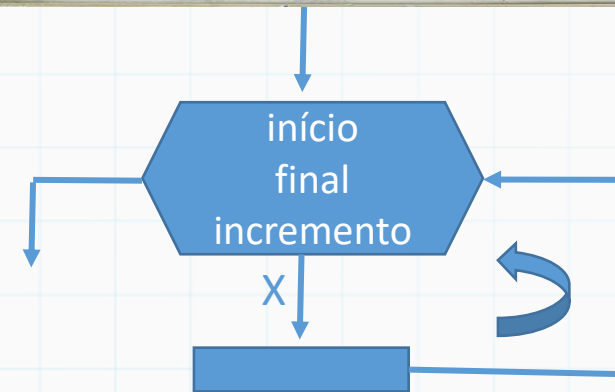
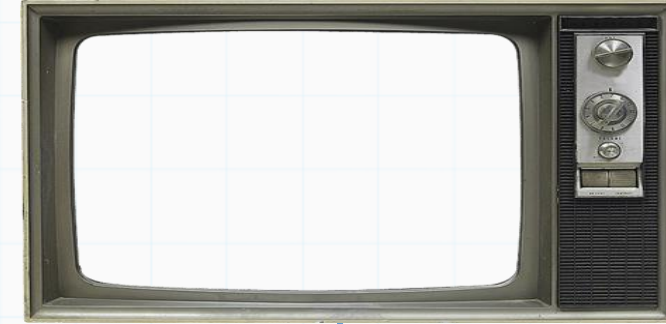
```
1 for x in range (1,5,2):  
2     print(x)  
3
```

Shell ×

```
>>> %Run teste.py
```

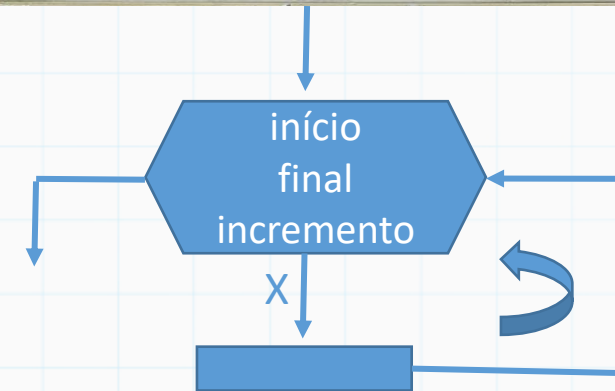
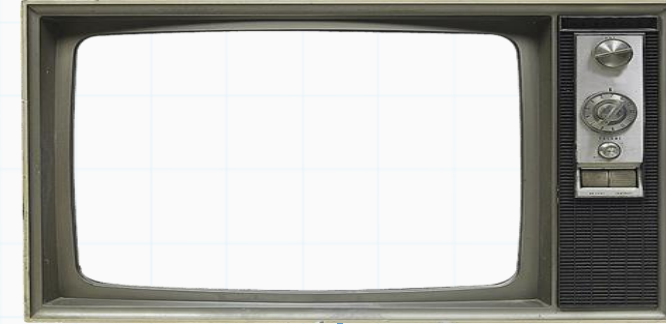
```
1  
3
```

código



X começa de <início>, e a cada iteração do laço, incrementa <incremento>, e executa enquanto X for estritamente menor que <final>

Repetição Contável



formato

for X in range (<início>, <fim>, <passo>):

<início> e <passo> são opcionais, quando omitidos, <início>=0 e <passo>=1

```
1 for i in range (3):  
2     print(i)  
3
```

↔ ... in range (0,3,1)

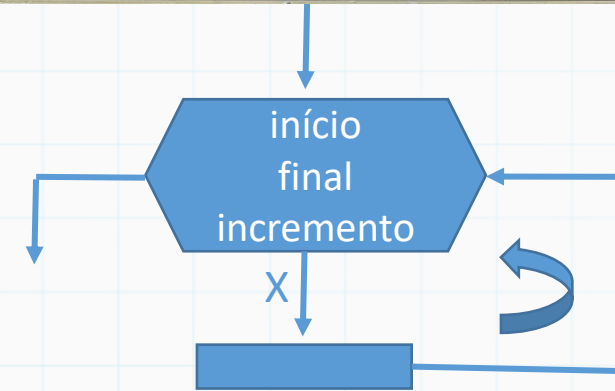
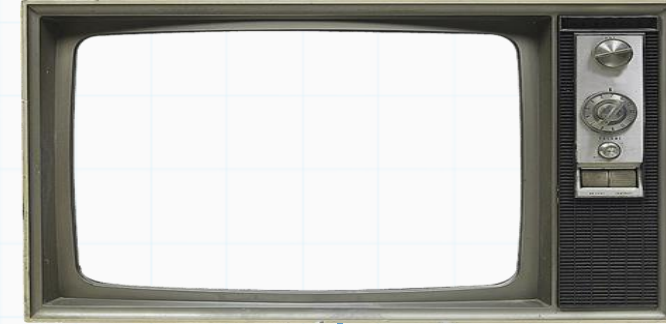
Shell x

```
>>> %Run teste.py
```

```
0  
1  
2
```

X começa de <início>, e a cada iteração do laço, incrementa <incremento>, e executa enquanto X for estritamente menor que <final>

Repetição Contável



formato

for X in range (<início>, <fim>, <passo>):

<início> e <passo> são opcionais, quando omitidos, <início>=0 e <passo>=1

```
1 for i in range (3):
2     print(i)
3
```

⇔ ... in range (0,3,1)

```
Shell x
>>> %Run teste.py
0
1
2
```

... in range (1,3,1) ⇔

```
1 for i in range (1,3):
2     print(i)
3
```

```
Shell x
>>> %Run teste.py
1
2
```

X começa de <início>, e a cada iteração do laço, incrementa <incremento>, e executa enquanto X for estritamente menor que <final>

Repetição Contável

formato

for X in range (<início>, <fim>, <passo>):

quando <passo> for negativo, a condição de parada se inverte, i.e., estritamente maior que <fim>.

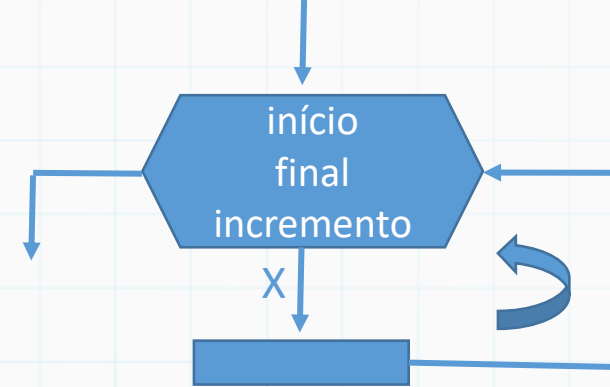
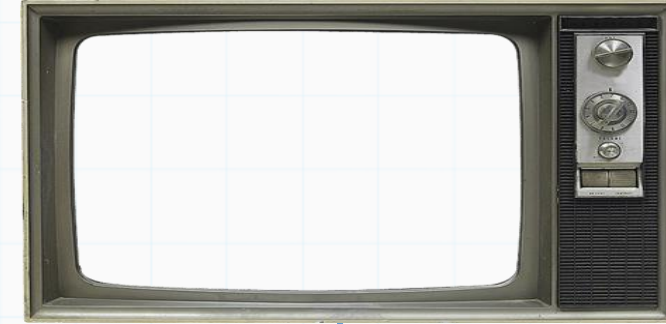
```
1 for a in range (5,0,-2):  
2     print(a)  
3
```

Shell x

```
>>> %Run teste.py
```

```
5  
3  
1
```

X começa de <início>, e a cada iteração do laço, incrementa <incremento> (que é negativo), e executa enquanto X for estritamente MAIOR que <final>



Repetição Contável

Podemos também usar o For sem o in range, apenas com um conjunto de valores

```
1 for x in (0,22,5,-13):  
2     print(x)
```

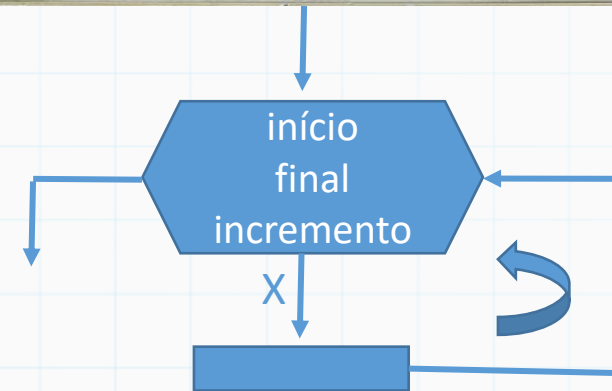
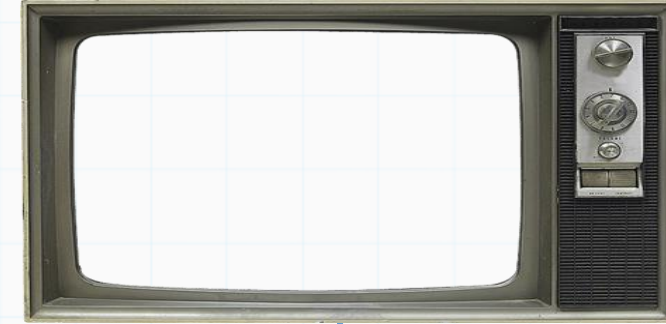
Shell x

Python 3.7.7 (bundled)

```
>>> %Run teste.py
```

```
0  
22  
5  
-13
```

x assume em cada repetição, os valores do conjunto, na ordem apresentada



Repetição Contável

Podemos também usar o For sem o in range, apenas com um conjunto de valores

```
1 for x in (0,5,"Vegeta",-3,5):  
2     print(x)  
3
```

Shell x

>>>

>>> %Run teste.py

0

5

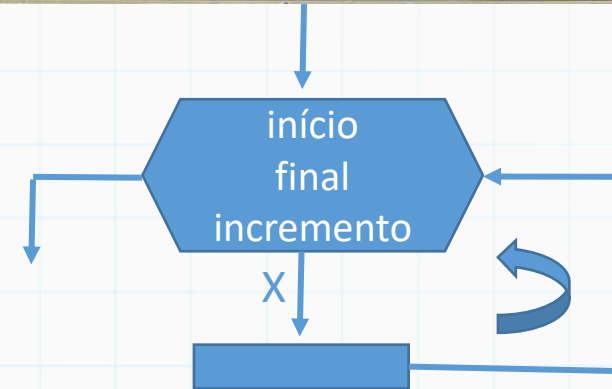
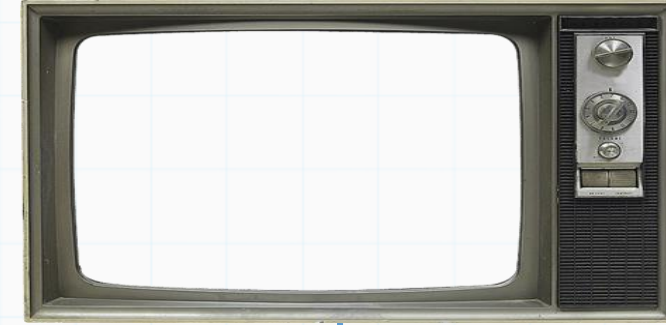
Vegeta

-3

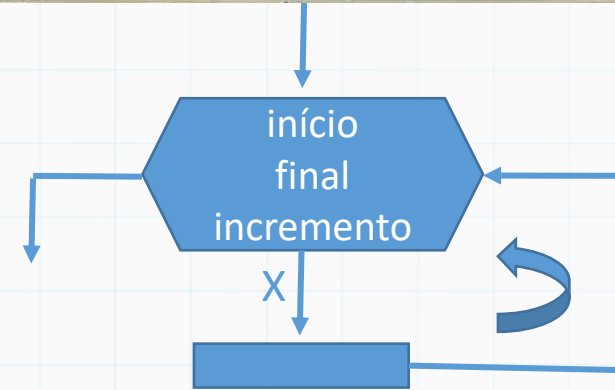
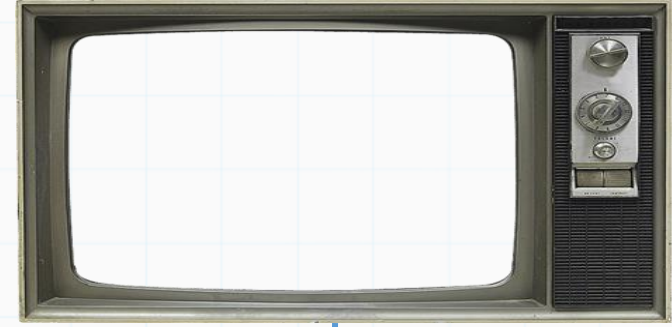
5

x assume em cada repetição, os valores do conjunto, na ordem apresentada

Neste caso, os valores não precisam ser numéricos



Repetição Contável



Comandos que alteram o fluxo da repetição:

break: encerra o laço imediatamente, mesmo se a condição de término não tiver sido alcançada.

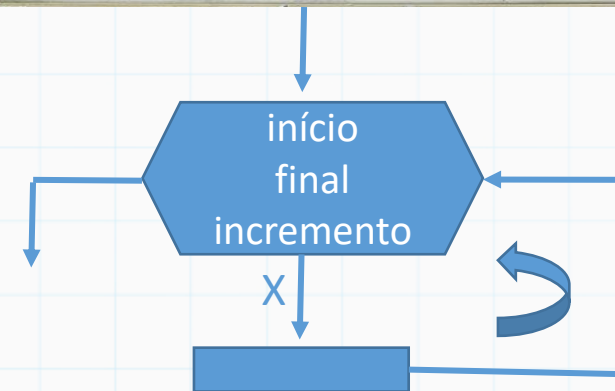
Repetição Contável

Comandos que alteram o fluxo da repetição:

break: encerra o laço imediatamente, mesmo se a condição de término não tiver sido alcançada.

```
1 for x in range (1,10,1):  
2     if (x==5):  
3         break  
4     print(x)  
5     print("fim")
```

O que será
impresso ?



Repetição Contável

Comandos que alteram o fluxo da repetição:

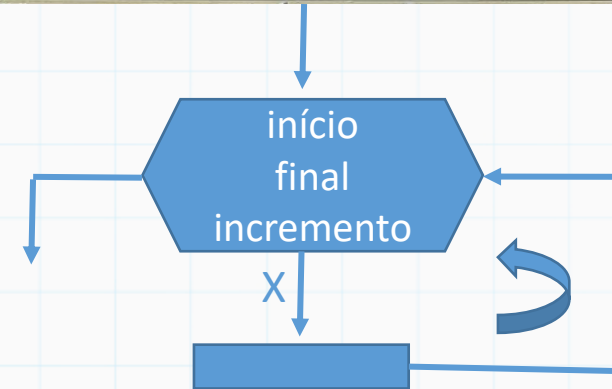
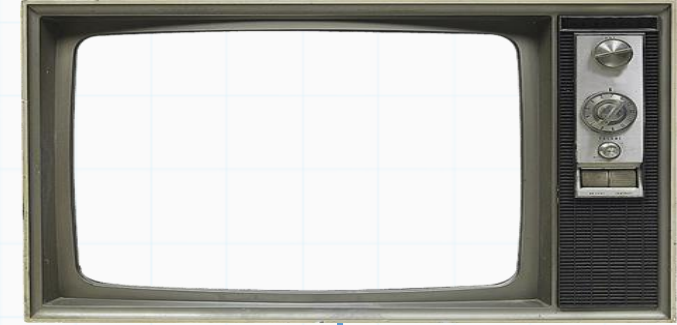
break: encerra o laço imediatamente, mesmo se a condição de término não tiver sido alcançada.

```
1 for x in range (1,10,1):  
2     if (x==5):  
3         break  
4     print(x)  
5     print("fim")
```

O que será
impresso ?

```
>>> %Run teste.py
```

```
1  
2  
3  
4  
fim
```



Repetição Contável

Comandos que alteram o fluxo da repetição:

break: encerra o laço imediatamente, mesmo se a condição de término não tiver sido alcançada.

```
1 for x in range (1,10,1):
2     if (x==5):
3         break
4     print(x)
5 print("fim")
```

```
>>> %Run teste.py
```

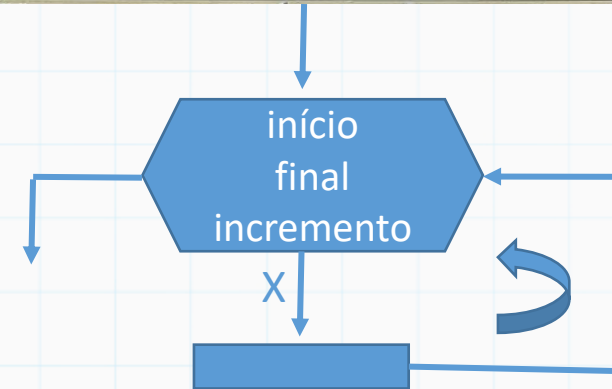
```
1
2
3
4
fim
```

```
1 x=1
2 while (x<10):
3     if (x==5):
4         break
5     print(x)
6     x=x+1
7 print("fim")
```

```
>>> %Run teste.py
```

```
1
2
3
4
fim
```

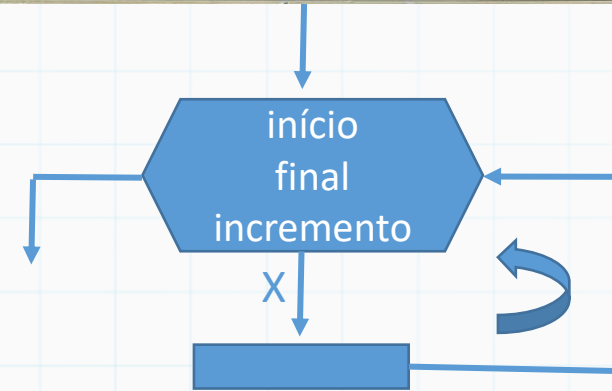
Vale também
para o While



Repetição Contável

Comandos que alteram o fluxo da repetição:

continue: passa imediatamente para a próxima iteração do laço.



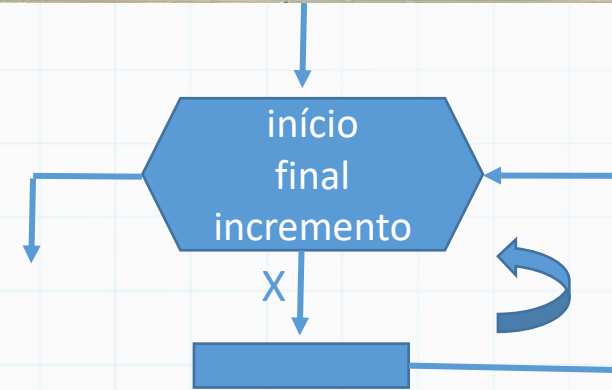
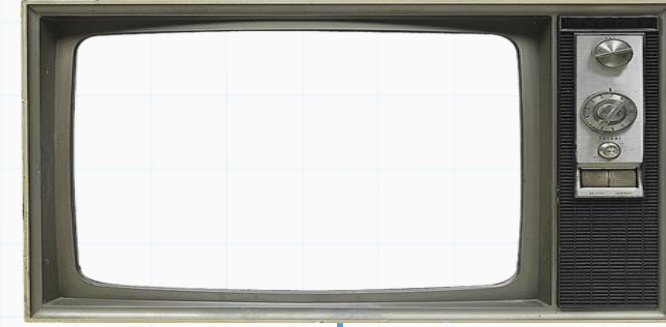
Repetição Contável

Comandos que alteram o fluxo da repetição:

continue: passa imediatamente para a próxima iteração do laço.

```
1 for x in range (1,10,1):  
2     if (x==5):  
3         continue  
4     print(x)  
5 print("fim")
```

O que será
impresso ?



Repetição Contável

Comandos que alteram o fluxo da repetição:

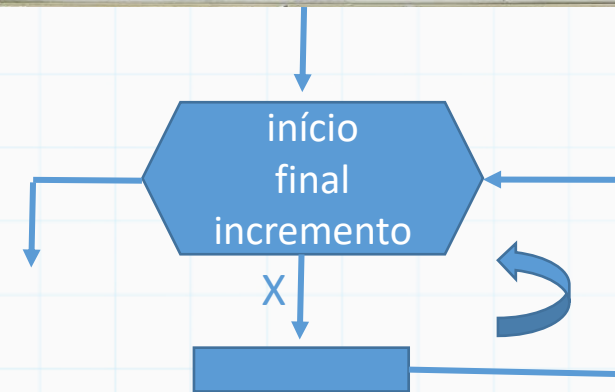
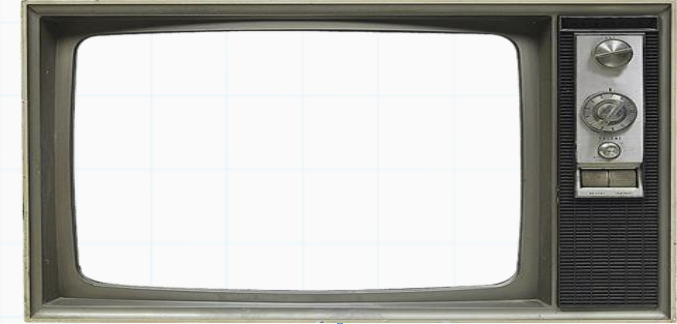
continue: passa imediatamente para a próxima iteração do laço.

```
1 for x in range (1,10,1):
2     if (x==5):
3         continue
4     print(x)
5 print("fim")
```

O que será
impresso ?

```
>>> %Run teste.py
```

```
1
2
3
4
6
7
8
9
fim
```



Repetição Contável

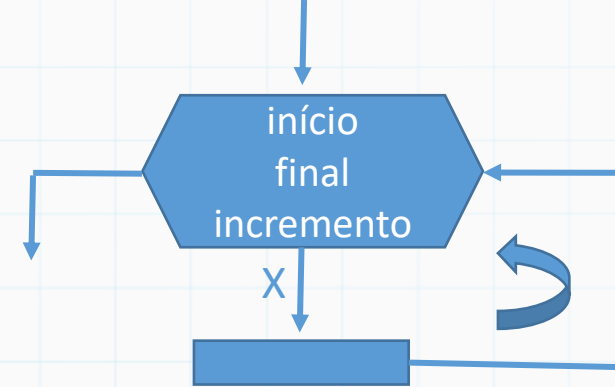
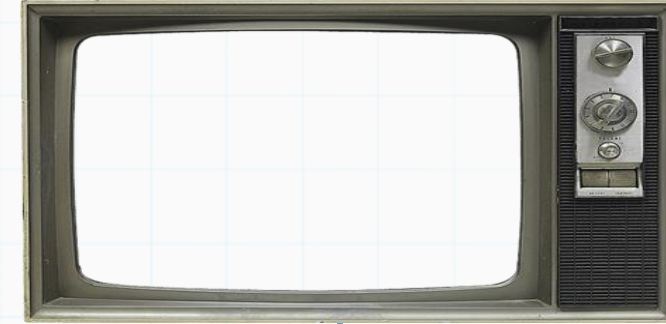
Comandos que alteram o fluxo da repetição:

continue: passa imediatamente para a próxima iteração do laço.

```
1 for x in range (1,10,1):
2     if (x==5):
3         continue
4     print(x)
5 print("fim")
```

```
1 x=1
2 while (x<10):
3     if (x==5):
4         continue
5     print(x)
6     x=x+1
7 print("fim")
```

E com While ?



```
>>> %Run teste.py
```

```
1
2
3
4
6
7
8
9
fim
```

Repetição Contável

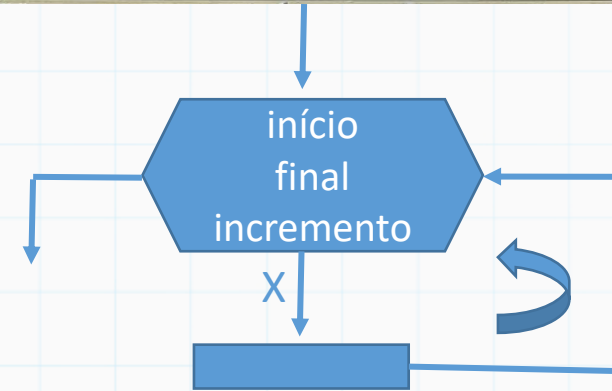
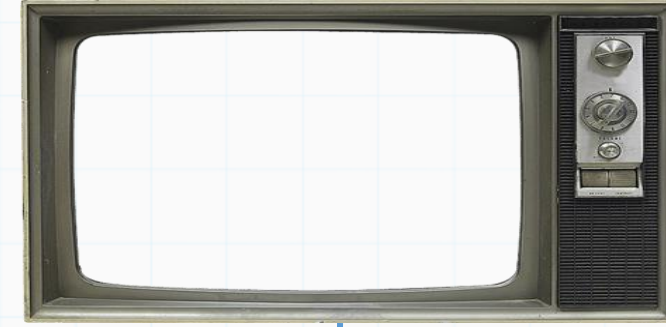
Comandos que alteram o fluxo da repetição:

continue: passa imediatamente para a próxima iteração do laço.

```
1 for x in range (1,10,1):
2     if (x==5):
3         continue
4     print(x)
5 print("fim")
```

```
1 x=1
2 while (x<10):
3     if (x==5):
4         continue
5     print(x)
6     x=x+1
7 print("fim")
```

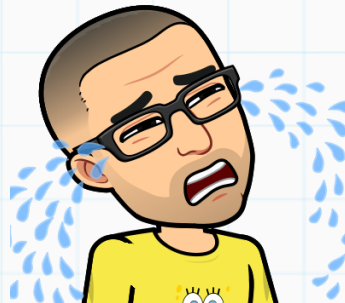
E com While ?



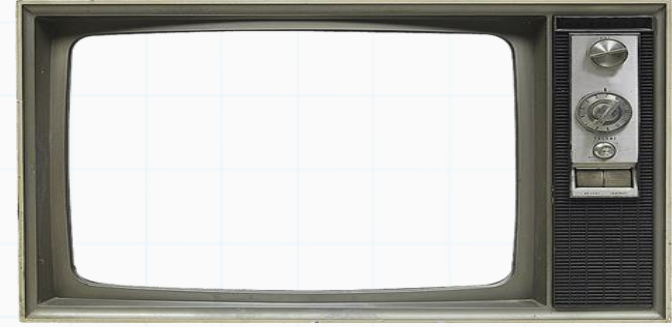
```
>>> %Run teste.py
```

```
1
2
3
4
6
7
8
9
fim
```

INFINITO

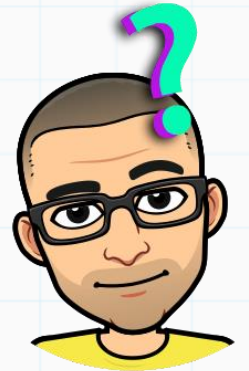


Repetição Contável

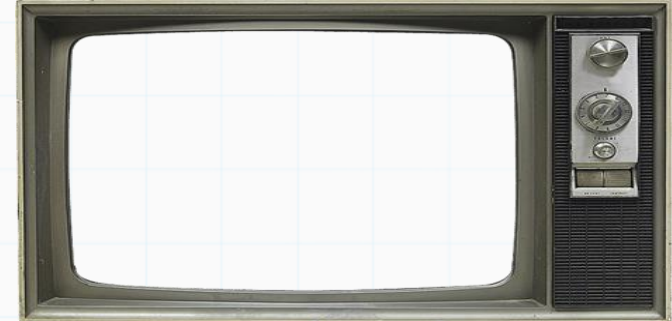


Então, While ou For ?

- Use um laço for, se você souber, antes de iniciar o laço, o número de vezes que você precisará executar o corpo do laço
- Use um laço while se você precisa repetir alguma computação até que alguma condição seja atendida, e você não pode calcular antecipadamente quando isso acontecerá



Repetição Contável



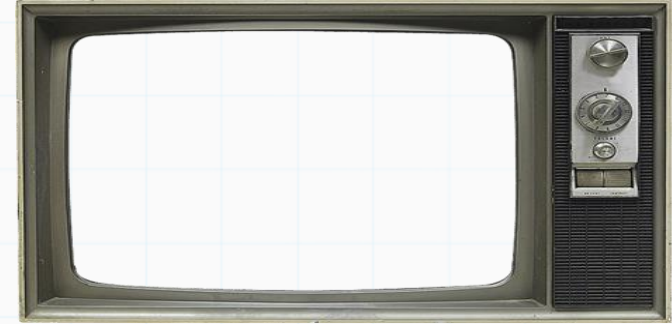
Vamos revisitar o exercício que fizemos antes:

Faça um programa que soma X números gerados aleatoriamente no intervalo de 1 a 10, onde X é informado pelo usuário

```
1 import random
2
3 x      = int(input('Digite um numero: '))
4 soma   = 0
5 contador = 0
6 while contador < x:
7     numero_sorteado = random.randint(1,10)
8     print(numero_sorteado)
9
10    soma      = soma + numero_sorteado
11    contador  = contador + 1
12 print('A soma eh', soma)
```

Vamos usar o For !

Repetição Contável



Vamos revisitar o exercício que fizemos antes:

Faça um programa que soma X números gerados aleatoriamente no intervalo de 1 a 10, onde X é informado pelo usuário

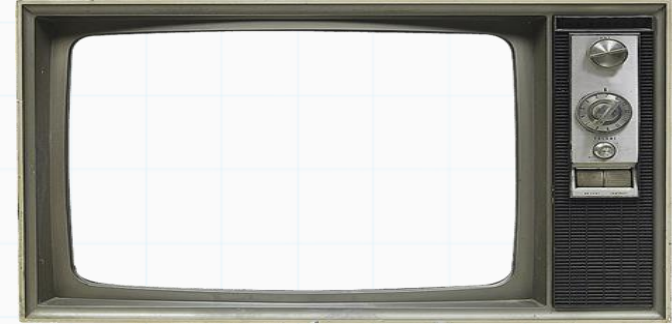
```
1 import random
2
3 x      = int(input('Digite um numero: '))
4 soma  = 0
5 contador = 0
6 while contador < x:
7     numero_sorteado = random.randint(1,10)
8     print(numero_sorteado)
9
10    soma      = soma + numero_sorteado
11    contador = contador + 1
12 print('A soma eh', soma)
```

```
1 from random import randint
2
3 x      = int(input('Digite um numero: '))
4 soma  = 0
5 for contador in range (x):
6     numero_sorteado = random.randint(1,10)
7     print(numero_sorteado)
8
9     soma      = soma + numero_sorteado
10 print('A soma eh', soma)
```

Vamos usar o For !

⇔ ... in range (0,x,1)

Repetição Contável



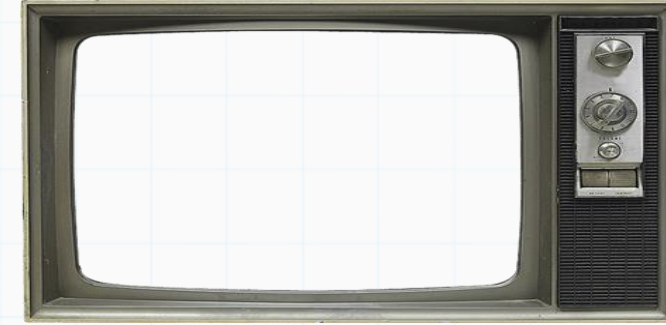
Vamos revisitar o exercício que fizemos antes:

Faça um programa que lê um número n e diga se ele é primo ou não. Lembrando que um número é primo se ele só tem 2 divisores, 1 e ele mesmo (i.e., não existe nenhum divisor de n entre 2 e $n-1$)

```
1 n      = int(input("n="))
2 numero = 2
3 primo  = True
4
5 while numero < n:
6     if (n % numero == 0):
7         primo = False
8         numero = numero + 1
9
10 if (primo):
11     print('É primo')
12 else:
13     print('Não é primo')
```

Vamos usar o For !

Repetição Contável



Vamos revisitar o exercício que fizemos antes:

Faça um programa que lê um número n e diga se ele é primo ou não. Lembrando que um número é primo se ele só tem 2 divisores, 1 e ele mesmo (i.e., não existe nenhum divisor de n entre 2 e $n-1$)

```
1 n = int(input("n="))
2 numero = 2
3 primo = True
4
5 while numero < n:
6     if (n % numero == 0):
7         primo = False
8         numero = numero + 1
9
10 if (primo):
11     print('É primo')
12 else:
13     print('Não é primo')
```

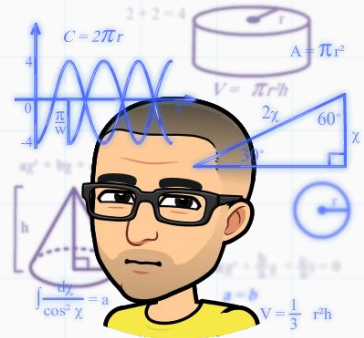
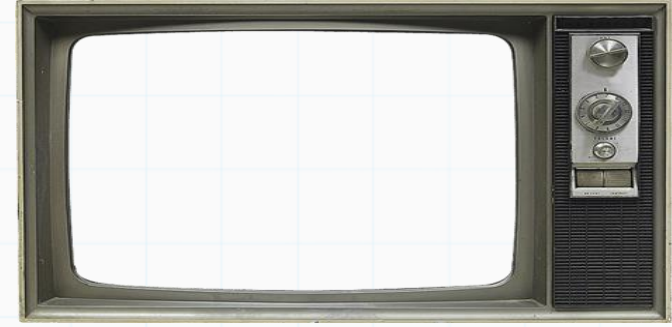
```
1 n = int(input("n="))
2 primo = True
3
4 for numero in range(2,n,1):
5     if (n % numero == 0):
6         primo = False
7
8 if (primo):
9     print('É primo')
10 else:
11     print('Não é primo')
```

Vamos usar o For !

Repetição Condicional

Como calcular séries usando laços: Qual o valor de S para a soma dos 50 primeiros termos da série.

$$S = \frac{3}{2^2} + \frac{6}{4^3} + \frac{9}{8^4} + \frac{12}{16^5} + \dots$$



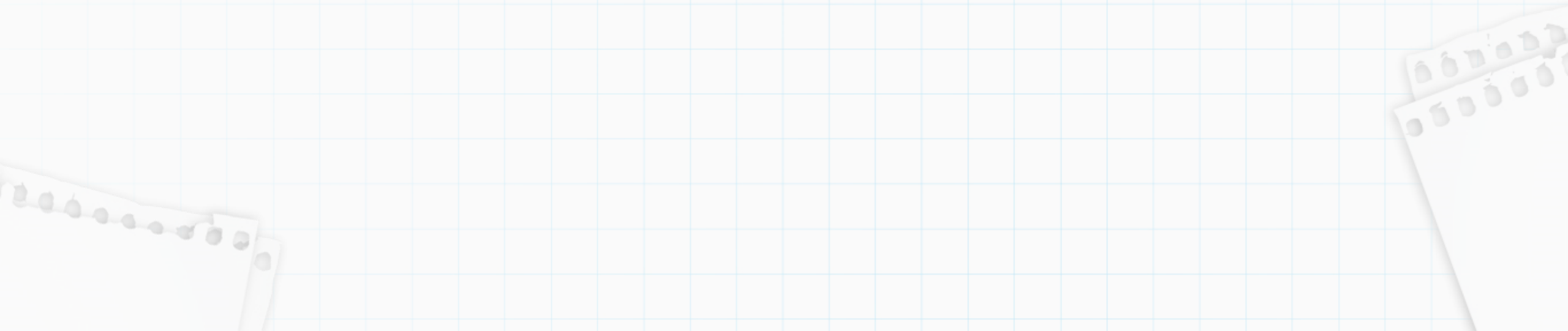
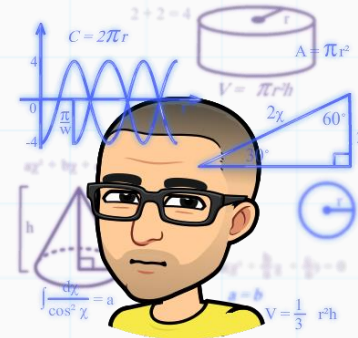
Repetição Condicional

Como calcular séries usando laços: Qual o valor de S para a soma dos 50 primeiros termos da série.

$$S = \frac{3}{2^2} + \frac{6}{4^3} + \frac{9}{8^4} + \frac{12}{16^5} + \dots$$

termo genérico da série: $\frac{x}{y^z}$

3 passos



Repetição Condicional

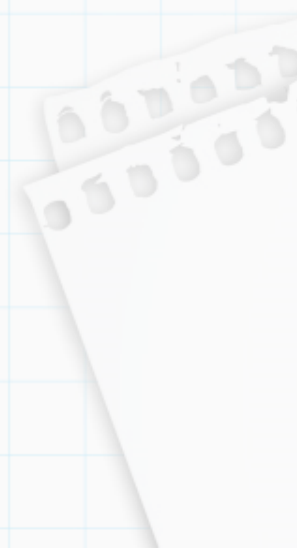
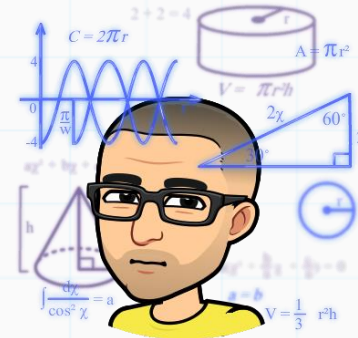
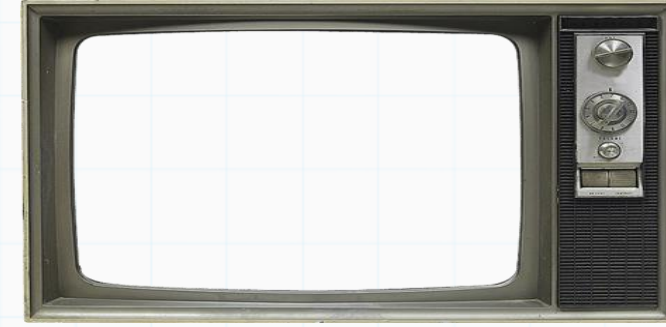
Como calcular séries usando laços: Qual o valor de S para a soma dos 50 primeiros termos da série.

$$S = \frac{3}{2^2} + \frac{6}{4^3} + \frac{9}{8^4} + \frac{12}{16^5} + \dots$$

termo genérico da série: $\frac{x}{y^z}$

3 passos

inicialização (termo inicial) : $x = 3, y = 2, z = 2$



Repetição Condicional

Como calcular séries usando laços: Qual o valor de S para a soma dos 50 primeiros termos da série.

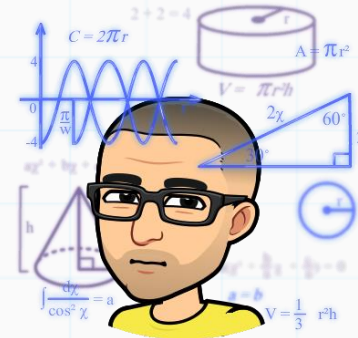
$$S = \frac{3}{2^2} + \frac{6}{4^3} + \frac{9}{8^4} + \frac{12}{16^5} + \dots$$

termo genérico da série: $\frac{x}{y^z}$

3 passos

inicialização (termo inicial) : $x = 3, y = 2, z = 2$

lei de formação (incremento) : $x \rightarrow$



Repetição Condicional

Como calcular séries usando laços: Qual o valor de S para a soma dos 50 primeiros termos da série.

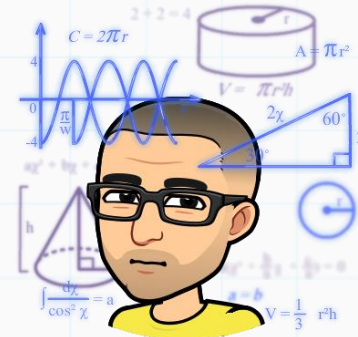
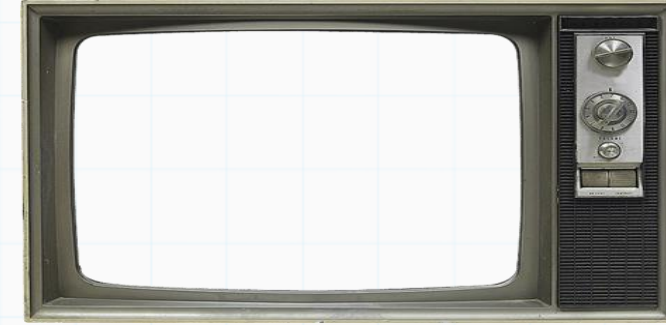
$$S = \frac{3}{2^2} + \frac{6}{4^3} + \frac{9}{8^4} + \frac{12}{16^5} + \dots$$

termo genérico da série: $\frac{x}{y^z}$

3 passos

inicialização (termo inicial) : $x = 3, y = 2, z = 2$

lei de formação (incremento) : $x \rightarrow x + 3$
 $y \rightarrow$



Repetição Condicional

Como calcular séries usando laços: Qual o valor de S para a soma dos 50 primeiros termos da série.

$$S = \frac{3}{2^2} + \frac{6}{4^3} + \frac{9}{8^4} + \frac{12}{16^5} + \dots$$

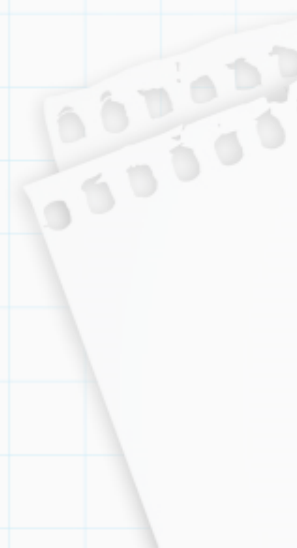
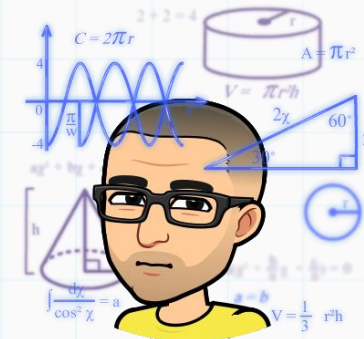
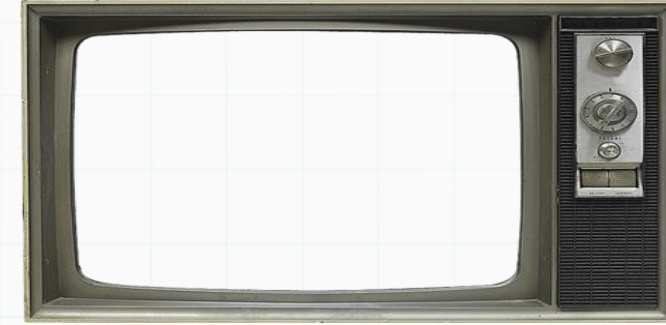
termo genérico da série: $\frac{x}{y^z}$

3 passos

inicialização (termo inicial) : $x = 3, y = 2, z = 2$

lei de formação (incremento) :

$x \rightarrow x + 3$
$y \rightarrow y * 2$
$z \rightarrow$



Repetição Condicional

Como calcular séries usando laços: Qual o valor de S para a soma dos 50 primeiros termos da série.

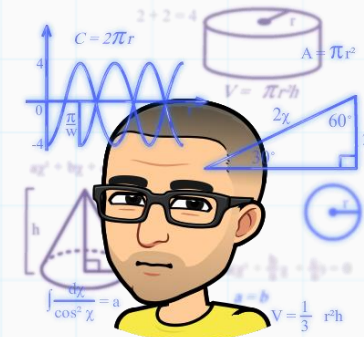
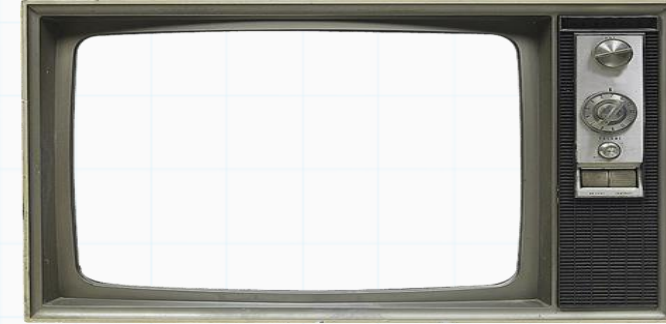
$$S = \frac{3}{2^2} + \frac{6}{4^3} + \frac{9}{8^4} + \frac{12}{16^5} + \dots$$

termo genérico da série: $\frac{x}{y^z}$

3 passos

inicialização (termo inicial) : $x = 3, y = 2, z = 2$

lei de formação (incremento) : $x \rightarrow x + 3$
 $y \rightarrow y * 2$
 $z \rightarrow z + 1$



Repetição Condicional

Como calcular séries usando laços: Qual o valor de S para a soma dos 50 primeiros termos da série.

$$S = \frac{3}{2^2} + \frac{6}{4^3} + \frac{9}{8^4} + \frac{12}{16^5} + \dots$$

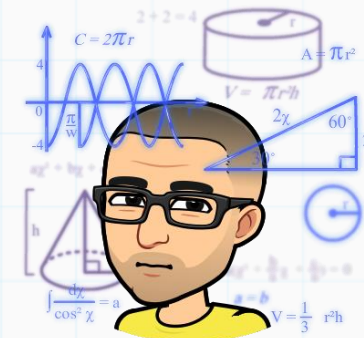
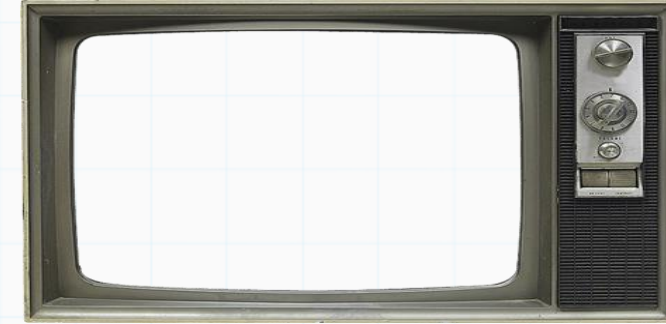
inicialização

```
1 x = 3
2 y = 2
3 z = 2
4 S = 0
```

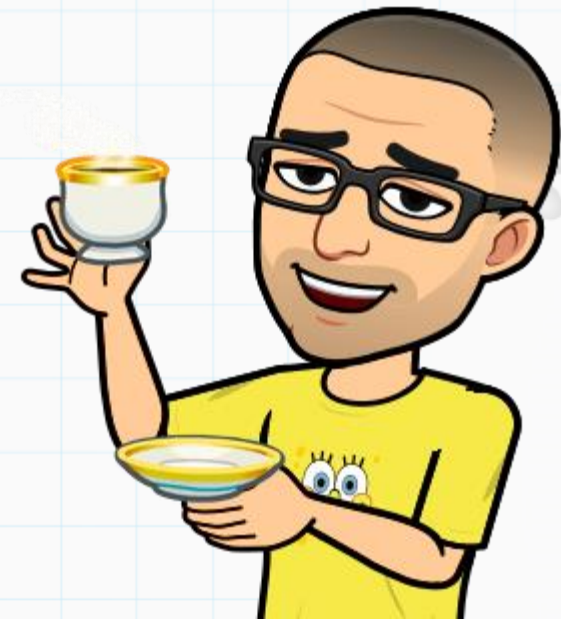
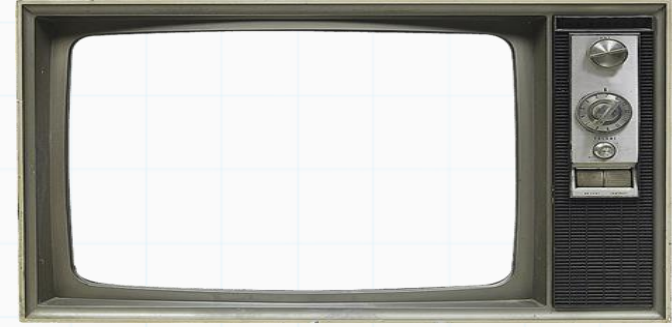
atualização

```
6 for i in range (50):
7     S = S + (x/(y**z))
8
9     x = x + 3
10    y = y * 2
11    z = z + 1
12
13 print ("S=",S)
```

[código](#)



Até a próxima



Slides baseados no curso de Vanessa Braganholo