

Semana de Nivelamento em LP





Lucas Migliorin

Instrutor

5º período de Engenharia de
Computação

Instagram: @lucas.migliorin

Email: lmdr.eng19@uea.edu.br



Letícia Magalhães

Monitora

7º período de Engenharia de
Computação

Instagram: @ledizia_

Email: Imm.eng18@uea.edu.br

Instruções para o decorrer do WorkShop

Perguntas durante a apresentação:

- ⬡ Escreva no chat sua dúvida que a monitora estará respondendo.
- ⬡ Caso a dúvida seja mais complexa de responder por texto, será dado uma pausa e então respondida pelo instrutor.
- ⬡ Mantenham os microfones e câmeras desligadas
- ⬡ Podem nos procurar para tirar dúvidas

Material do Workshop disponível online:

- ⬡ Link: <https://github.com/Migliorin/Semana-LP>

Siga as redes sociais do DAETEC e CAECOMP!



@daetec_uea



@caecomp.uea

AGENDA DO DIA

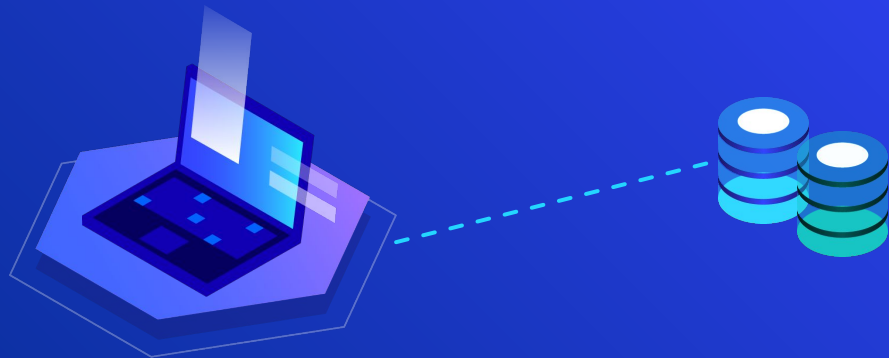
Listas

Laços de repetição



1. Laços de repetição

Vamos repetir



Laços de repetição

Os tipos que existem:



Laços de repetição

Os tipos que existem:

⬡ While



Laços de repetição

Os tipos que existem:

- ⬡ While
- ⬡ For



Laços de repetição

Os tipos que existem:

- ⬡ While
- ⬡ For

Vamos começar
por esse aqui



Laços de repetição

O While



Laços de repetição

O While

code	output
<pre>1 a = 1 2 while a < 10: 3 print (a) 4 a += 2</pre>	
variables	

Laços de repetição

O While

code	output
<pre>1 a = 1 2 while a < 10: 3 print (a) 4 a += 2</pre>	
variables	

$a < 10$

$a = 1$

Laços de repetição

O While

code	output
<pre>1 a = 1 2 while a < 10: 3 print (a) 4 a += 2</pre>	
variables	

$a < 10$

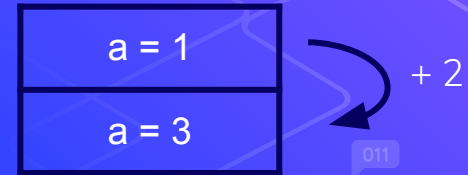


Laços de repetição

O While

code	output
<pre>1 a = 1 2 while a < 10: 3 print (a) 4 a += 2</pre>	
variables	

$a < 10$

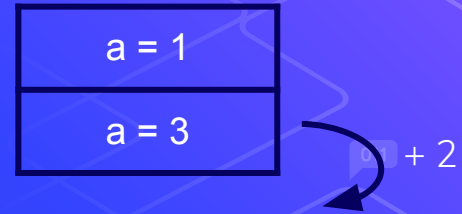


Laços de repetição

O While

code	output
<pre>1 a = 1 2 while a < 10: 3 print (a) 4 a += 2</pre>	
variables	

$a < 10$

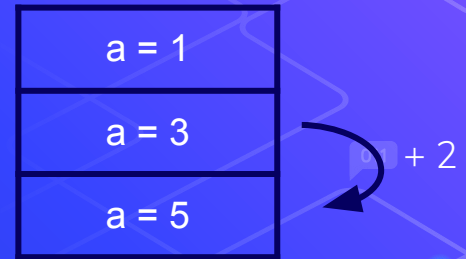


Laços de repetição

O While

code	output
<pre>1 a = 1 2 while a < 10: 3 print (a) 4 a += 2</pre>	
variables	

$a < 10$

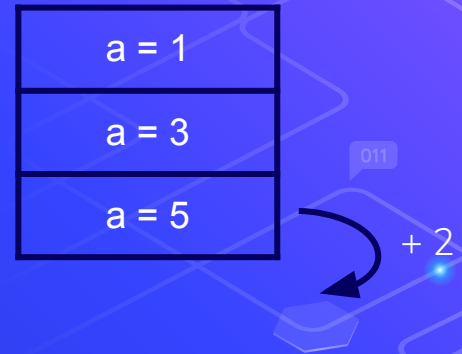


Laços de repetição

O While

code	output
<pre>1 a = 1 2 while a < 10: 3 print (a) 4 a += 2</pre>	
variables	

$a < 10$

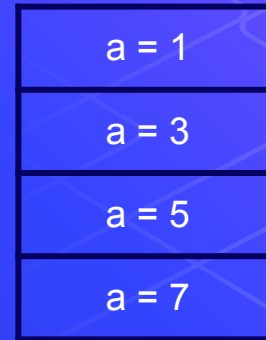


Laços de repetição

O While

code	output
<pre>1 a = 1 2 while a < 10: 3 print (a) 4 a += 2</pre>	
variables	

$a < 10$



Laços de repetição

O While

code	output
<pre>1 a = 1 2 while a < 10: 3 print (a) 4 a += 2</pre>	
variables	

$a < 10$

a = 1
a = 3
a = 5
a = 7

+ 2

Laços de repetição

O While

code	output
<pre>1 a = 1 2 while a < 10: 3 print (a) 4 a += 2</pre>	
variables	

$a < 10$

a = 1
a = 3
a = 5
a = 7
a = 9

+ 2

Laços de repetição

O While

code	output
<pre>1 a = 1 2 while a < 10: 3 print (a) 4 a += 2</pre>	
variables	

$a < 10$

a = 1
a = 3
a = 5
a = 7
a = 9

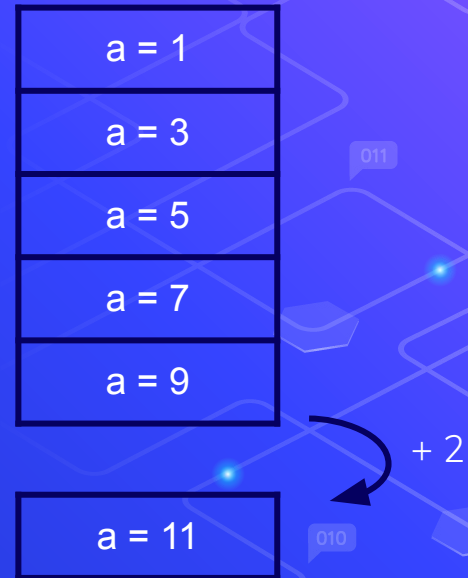
+ 2

Laços de repetição

O While

code	output
<pre>1 a = 1 2 while a < 10: 3 print (a) 4 a += 2</pre>	
variables	

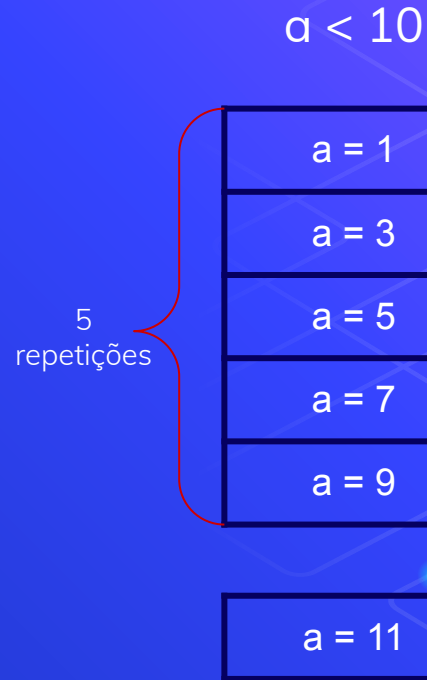
$a < 10$



Laços de repetição

O While

code	output
<pre>1 a = 1 2 while a < 10: 3 print (a) 4 a += 2</pre>	
variables	



Laços de repetição

O For



Laços de repetição

O For

```
1  for itr in range(1,10):  
2  |      print(itr)
```

Laços de repetição

O For

```
1  for itr in range(1,10):  
2  |      print(itr)
```

A saída:

Laços de repetição

O For

```
1  for itr in range(1,10):  
2      print(itr)
```

A saída:

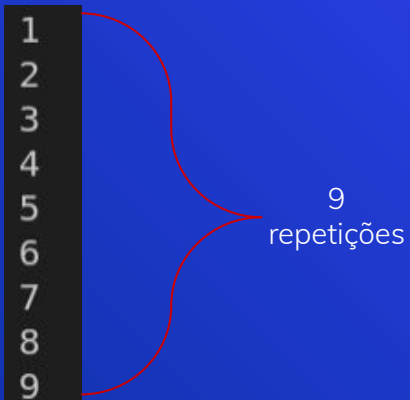
```
1  
2  
3  
4  
5  
6  
7  
8  
9
```

Laços de repetição

O For

```
1  for itr in range(1,10):  
2  |    print(itr)
```

A saída:



1
2
3
4
5
6
7
8
9

9
repetições

Laços de repetição

O For

```
1  for itr in range(1,10):  
2  |    print(itr)
```

A saída:

```
1  
2  
3  
4  
5  
6  
7  
8  
9
```

Nossa iteração
começa no 1 e vai
até o 9

Laços de repetição

O For

```
1  for itr in range(1,10):  
2  |    print(itr)
```

A saída:

```
1  
2  
3  
4  
5  
6  
7  
8  
9
```

Se declaramos
somente range(10) ele
printa começando do 0

Laços de repetição

O For

```
1  for itr in range(10):  
2      print(itr)
```

Laços de repetição

O For

```
1  for itr in range(10):  
2      print(itr)
```

A saída:

Laços de repetição

O For

```
1  for itr in range(10):  
2      print(itr)
```

A saída:

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9
```

Laços de repetição

O For

```
1  for itr in range(10):  
2      print(itr)
```

A saída:



0
1
2
3
4
5
6
7
8
9

10
repetições

Vamos fazer juntos o seguinte programa

1. Escreva um algoritmo que imprime os números ímpares de 1 até 1000

Dica: Os números primos tem resto de divisão por 2 igual a 1



Vamos fazer juntos o seguinte programa

1. Escreva um programa em Python que recebe um número inteiro n e que calcula e retorna o fatorial de n

Dica: Lembrete $6! = 6*5*4*3*2*1$





2. Listas

De que?



Nossa primeira lista



Nossa primeira lista

```
● ● ●  
nossa_lista = [0,1,2,3,4,5,6,7,8,9]
```



Nossa primeira lista

```
● ● ●  
nossa_lista = [0,1,2,3,4,5,6,7,8,9]
```

Podemos
armazenar
números



Nossa primeira lista



```
nossa_lista = [0,1,2,3,4,5,6,7,8,9]
```

Podemos
armazenar
números



```
nossa_lista = ['aa','b','cccc','dd','e','f','ggg']
```



Nossa primeira lista



```
nossa_lista = [0,1,2,3,4,5,6,7,8,9]
```

Podemos
armazenar
números



```
nossa_lista = ['aa','b','cccc','dd','e','f','ggg']
```

Podemos
armazenar textos



Nossa primeira lista



```
nossa_lista = [0,1,2,3,4,5,6,7,8,9]
```

Podemos
armazenar
números



```
nossa_lista = ['aa','b','cccc','dd','e','f','ggg']
```

Podemos
armazenar textos



```
nossa_lista = ['lucas miglierin',20,1.75,True]
```



Nossa primeira lista



```
nossa_lista = [0,1,2,3,4,5,6,7,8,9]
```

Podemos
armazenar
números



```
nossa_lista = ['aa','b','cccc','dd','e','f','ggg']
```

Podemos
armazenar textos



```
nossa_lista = ['lucas miglierin',20,1.75,True]
```

Podemos
armazenar todo
tipo de variavel



Acessando seus valores



```
nossa_lista = ['lucas migliorin',20,1.75,True]
```

```
# Acessando o primeiro valor
```

```
nossa_lista[0]
```



Acessando seus valores



```
nossa_lista = ['lucas migliorin',20,1.75,True]
```

```
# Acessando o primeiro valor
```

```
nossa_lista[0]
```



Acessando seus valores

```
● ● ●  
nossa_lista = ['lucas migliorin',20,1.75,True]  
  
# Acessando o primeiro valor  
nossa_lista[0]
```

Posição na lista:
posição observada - 1



Acessando seus valores



```
nossa_lista = ['lucas miglierin',20,1.75,True]
```

```
# Acessando o primeiro valor
```

```
nossa_lista[0]
```

```
# Acessando o segundo valor
```

```
nossa_lista[1]
```



Acessando seus valores



```
nossa_lista = ['lucas migliorin',20,1.75,True]
```

```
# Acessando o primeiro valor
```

```
nossa_lista[0]
```

```
# Acessando o segundo valor
```

```
nossa_lista[1]
```



Acessando seus valores

```
● ● ●  
nossa_lista = ['lucas migliorin',20,1.75,True]  
  
# Acessando o primeiro valor  
nossa_lista[0]  
  
# Acessando o segundo valor  
nossa_lista[1]
```

Posição na lista:
posição observada - 1



Acessando seus valores

OU



Acessando seus valores



```
nossa_lista = ['lucas migliorin',20,1.75,True]

for elemento in nossa_lista:
    print(elemento)
```



Adicionando valores



```
nossa_lista = ['lucas migliorin',20,1.75,True]  
  
nossa_lista.append('lmdr.eng19@uea.edu.br')  
  
# ['lucas migliorin',20,1.75,True,'lmdr.eng19@uea.edu.br']
```



Adicionando valores

```
● ● ●  
nossa_lista = ['lucas migliorin',20,1.75,True]  
nossa_lista.append('lmdr.eng19@uea.edu.br')  
# ['lucas migliorin',20,1.75,True,'lmdr.eng19@uea.edu.br']
```



Adicionando valores

```
● ● ●  
nossa_lista = ['lucas miglriorin',20,1.75,True]  
nossa_lista.append('lmdr.eng19@uea.edu.br')  
# ['lucas miglriorin', 20,1.75,True,'lmdr.eng19@uea.edu.br']
```

Função para
adicionarmos no fim da
lista



Adicionando valores



```
nossa_lista = ['lucas migliorin',20,1.75,True]  
  
nossa_lista.append('lmdr.eng19@uea.edu.br')  
  
# ['lucas migliorin',20,1.75,True,'lmdr.eng19@uea.edu.br']
```



Adicionando valores

```
● ● ●  
nossa_lista = ['lucas miglorin',20,1.75,True]  
nossa_lista.append('lmdr.eng19@uea.edu.br')  
# ['lucas miglorin',20,1.75,True,'lmdr.eng19@uea.edu.br']
```



Deletando valores

```
● ● ●  
nossa_lista = ['lucas miglorin',20,1.75,True]  
nossa_lista.append('lmdr.eng19@uea.edu.br')  
# ['lucas miglorin',20,1.75,True,'lmdr.eng19@uea.edu.br']  
del nossa_lista[3]
```



Deletando valores

```
● ● ●  
nossa_lista = ['lucas miglierin',20,1.75,True]  
nossa_lista.append('lmdr.eng19@uea.edu.br')  
# ['lucas miglierin',20,1.75,True,'lmdr.eng19@uea.edu.br']  
del nossa_lista[3]  
# ['lucas miglierin',20,1.75,'lmdr.eng19@uea.edu.br']
```



Deletando valores

```
● ● ●  
nossa_lista = ['lucas miglierin',20,1.75,True]  
nossa_lista.append('lmdr.eng19@uea.edu.br')  
# ['lucas miglierin',20,1.75,True,'lmdr.eng19@uea.edu.br']  
del nossa_lista[3]  
# ['lucas miglierin',20,1.75,'lmdr.eng19@uea.edu.br']
```



Vamos fazer juntos o seguinte programa

1. Escreva um programa em Python que leia vários números positivos e armazene em uma lista, quando for passado ele para de armazenar e printa nossa lista

