Aula 11

for vs while

O Python oferece duas opções de loops para executar tarefas repetitivas em seu código: o loop for e o loop while. Embora ambos os loops permitam executar tarefas repetitivas, existem diferenças importantes entre eles:

- Sintaxe: A sintaxe do loop for é mais simples do que a do loop while. No loop for, precisamos apenas especificar uma variável de iteração e uma sequência de valores que essa variável deve assumir. Já no loop while, precisamos especificar uma condição de término explícita, que pode ser mais complicada de escrever.

- Controle de iteração: No loop for, o Python cuida do controle de iteração automaticamente. Em outras palavras, a variável de iteração é atualizada automaticamente a cada iteração do loop. Já no loop while, precisamos atualizar manualmente a variável de controle de iteração em cada iteração do loop, caso contrário, o loop pode se tornar infinito.

- Flexibilidade: O loop while é mais flexível do que o loop for, pois permite que o programador escreva condições mais complexas para controlar a execução do loop. Além disso, o loop while pode ser usado para executar uma tarefa repetitiva até que uma condição específica seja atendida, enquanto o loop for é mais adequado para iterar sobre uma sequência fixa de valores.

- Eficiência: Em geral, o loop for é mais eficiente do que o loop while, especialmente quando precisamos iterar sobre uma sequência fixa de valores. Isso ocorre porque o loop for é executado pelo interpretador Python de maneira mais eficiente, já que sabe de antemão quantas vezes o loop precisa ser executado.

Exemplos:

- Abaixo temos uma lista de planetas. O loop for vai pegar cada elemento da lista individualmente, enquanto no loop while, acessamos o elemento da lista através do índice (definido com a variável i). O código está praticamente igual nos dois loops. Ambos criam uma variável “planeta” que receberá um elemento da lista e exibir.

```python
# usando uma lista fixa
planetas = ['Mercúrio','Vênus','Terra','Marte']

# realizando o loop com for
for planeta in planetas:
    print(planeta)

# realizando o mesmo loop com o while
i = 0
while i < len(planeta):
    planeta = planetas[i]
    print(planeta)
    i += 1
```

- Agora, vamos usar um índice para passar pela mesma lista.

```python
planetas = ['Mercúrio','Vênus','Terra','Marte']

for i in range(len(planetas)):
    print(planetas[i])

i = 0
while i < len(planetas):
    print(planetas[i])
    i += 1
```

- Agora vamos usar for e while para receber uma quantidade indeterminada de números do usuário.

```python
numeros = []
for i in range(1000):
    num = input('Digite um número ou "sair" para mostrar a lista : ')
    if num == 'sair':
        break
    num = int(num)
    numeros.append(num)
print(numeros)

numeros = []
while True:
    num = input('Digite um número ou "sair" para mostrar a lista : ')
    if num == 'sair':
        break
    num = int(num)
    numeros.append(num)
print(numeros)
```

Repare que no exemplo acima, temos que fazer uma gambiarra para simular um loop infinito.

Em resumo, a escolha entre usar um loop for ou um loop while depende da tarefa específica que estamos tentando realizar. O loop for é geralmente mais fácil de ler e escrever, especialmente quando iteramos sobre uma sequência fixa de valores. Já o loop while é mais flexível e permite condições de término mais complexas, o que o torna mais adequado para tarefas repetitivas com condições de término mais complicadas.

Prompt de Comando / Terminal

Para um uso básicos do Prompt de Comando, veja o arquivo básico-prompt-comando.pdf que está junto do projeto.

Git

Para usar o Git, primeiro temos que baixar em https://git-scm.com/downloads e baixar a versão para seu sistema operacional Ele pode ser instalado para seu usuário, então não se preocupe em precisar de permissões de acesso para instalação. Lembre de adicionar ao PATH o executável do Git.
Todos os comandos apresentados são feitos a partir do terminal ou do executável Git Bash de dentro da pasta clonada.
Para maiores explicações sobre o funcionamento, veja esse link https://git-scm.com/book/pt-br/v2/Come%C3%A7ando-O-B%C3%A1sico-do-Git e o livro Git Notes for Professionals (o link pode ser encontrado no arquivo link-livros.txt, no projeto).

Comandos Básicos

git clone: clona um repositório existente. Esse é o único comando que não é usado dentro da pasta do repositório. Onde você o executar, vai ser criado uma pasta com o nome do projeto e todos os seus arquivos dentro. Depois de clonado, entre na pasta usando o comando "cd <pasta_clonada>" para os demais comandos.

Exemplo: git clone https://github.com/gutohertzog/python-senac

git pull: chamado dentro da pasta do seu repositório clonado, vai atualizar todos os arquivos internos com a última versão do repositório remoto.

Exemplo: git pull

Para mais comandos, veja o site http://comandosgit.github.io/

Exercícios para Praticar

1. Escreva um loop for que imprima os números de 1 a 10.
2. Escreva um loop while que imprima os números de 1 a 10.
3. Escreva um loop for que imprima os números pares de 1 a 20.
4. Escreva um loop while que imprima os números pares de 1 a 20.
5. Escreva um loop for que imprima os números ímpares de 1 a 20.
6. Escreva um loop while que imprima os números ímpares de 1 a 20.
7. Escreva um loop for que calcule a soma dos números de 1 a 10.
8. Escreva um loop while que calcule a soma dos números de 1 a 10.
9. Escreva um loop for que calcule o produto dos números de 1 a 5.
10. Escreva um loop while que calcule o produto dos números de 1 a 5.
11. Escreva um loop for que imprima os caracteres de uma string.
12. Escreva um loop while que imprima os caracteres de uma string.
13. Escreva um loop for que encontre o maior elemento de uma lista.
14. Escreva um loop while que encontre o maior elemento de uma lista.
15. Escreva um loop for que encontre o menor elemento de uma lista.
16. Escreva um loop while que encontre o menor elemento de uma lista.
17. Escreva um loop for que conte o número de elementos em uma lista.
18. Escreva um loop while que conte o número de elementos em uma lista.
19. Escreva um loop for que inverta uma string.
20. Escreva um loop while que inverta uma string.
21. Escreva um loop for para criar uma lista de números de 1 a 50. Use um loop while interno para verificar se cada número é primo.
22. Escreva um loop while que lê uma lista de números do usuário e depois use um loop for para iterar sobre a lista e calcular a soma dos valores.
23. Escreva um loop for que imprime uma pirâmide de asteriscos de tamanho inserido pelo usuário. Use um loop while interno para controlar o número de asteriscos em cada linha.
24. Escreva um loop while que encontre o segundo maior elemento de uma lista. Depois use um loop for para iterar sobre a lista e encontrar o segundo menor elemento.
25. Escreva um loop for que imprime as tabuadas de multiplicação de 1 a 10. Use um loop while interno para iterar sobre os números de 1 a 10.
26. Escreva um loop while que lê uma lista de números do usuário e depois use um loop for para remover todos os valores duplicados. Depois, mostre uma lista contendo apenas os valores duplicados.
27. Escreva um loop for que calcula a soma dos quadrados dos números de 1 a 10 e salve em uma lista. Depois, use um loop while para iterar sobre a lista mostrando apenas os números pares.
28. Escreva um loop while que lê uma lista de nomes do usuário. Depois, use um for loop para verificar se um nome específico está na lista.
29. Escreva um loop while que lê uma lista de números do usuário. Depois use um loop for para iterar sobre a lista e calcular a soma dos valores.
30. Escreva um loop for que imprime as tabuadas de adição de 1 a 10. Use um loop while interno para iterar sobre os números de 1 a 10.
31. Escreva um loop for que imprime as tabuadas de subtração de 1 a 10. Use um loop while interno para iterar sobre os números de 1 a 10.
32. Escreva um loop for que imprime as tabuadas de multiplicação de 1 a 10. Use um loop while interno para iterar sobre os números de 1 a 10.
33. Escreva um loop for que imprime as tabuadas de divisão de 1 a 10. Use um loop while interno para iterar sobre os números de 1 a 10.
34. Escreva um loop for que imprime as tabuadas de potenciação de 1 a 10. Use um loop while interno para iterar sobre os números de 1 a 10.

35. Escreva um loop for que imprime uma pirâmide de números. Use um loop while interno para mostrar cada linha com seu respectivo número.
    Exemplo: uma pirâmide de tamanho 5 deve ser mostrado como