Aula 10

Revisão

Módulo random.

É a biblioteca padrão do Python que fornece funções para gerar números aleatórios.

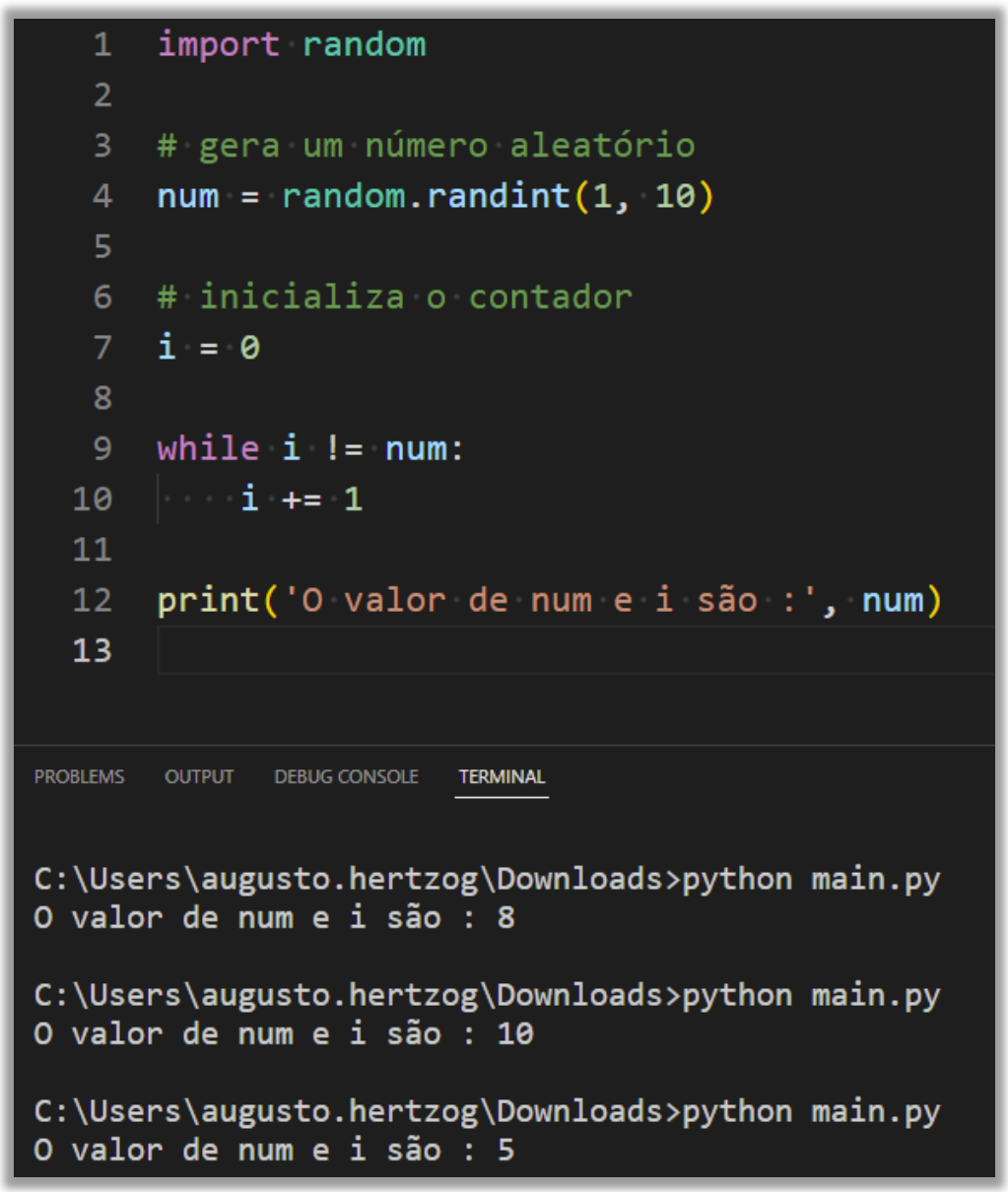
Abaixo tem algumas funções que iremos usar para gerar exemplos em aula mais rapidamente:

- random(): retorna um número de ponto flutuante entre 0 e 1.
- randint(a, b): retorna um número inteiro aleatório entre a e b, incluindo a e b.
- uniform(a, b): retorna um número de ponto flutuante aleatório entre a e b.
- choice(seq): retorna um elemento aleatório da sequência seq.
- shuffle(seq): embaralha os elementos da sequência seq de forma aleatória.

Comando while

Anteriormente, vimos como podemos usar o for loop. Ele é muito bom quando sabemos quantas vezes temos que repetir algum bloco de código, mas se tivermos que lidar com um número indeterminado de repetições, é melhor utilizarmos o while loop.

Abaixo, vamos gerar um número aleatório e incrementaremos o valor de i até que ele se iguale ao valor aleatório gerado.

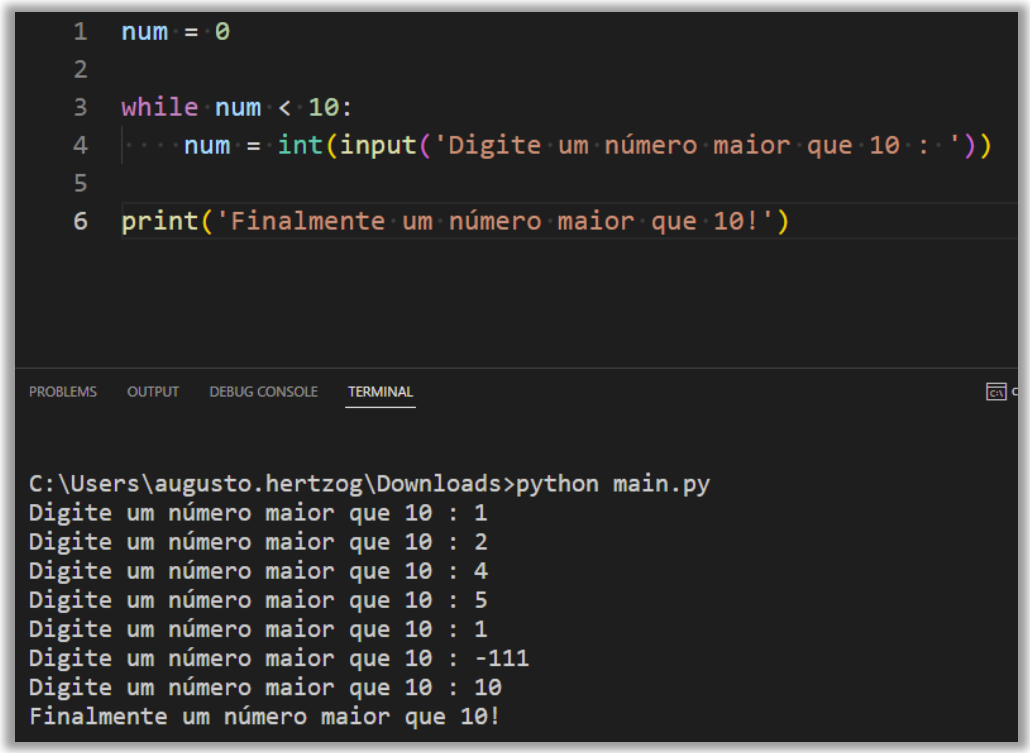


```python
import random

# gera um número aleatório
num = random.randint(1, 10)

# inicializa o contador
i = 0

while i != num:
    i += 1

print('O valor de num e i são :', num)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

C:\Users\augusto.hertzog\Downloads>python main.py
O valor de num e i são : 8

C:\Users\augusto.hertzog\Downloads>python main.py
O valor de num e i são : 10

C:\Users\augusto.hertzog\Downloads>python main.py
O valor de num e i são : 5
```

Para o while ser interrompido, é preciso que a condição seja falsa, ou seja, será interrompido apenas quando o valor de “i” for igual ao valor de “num”.

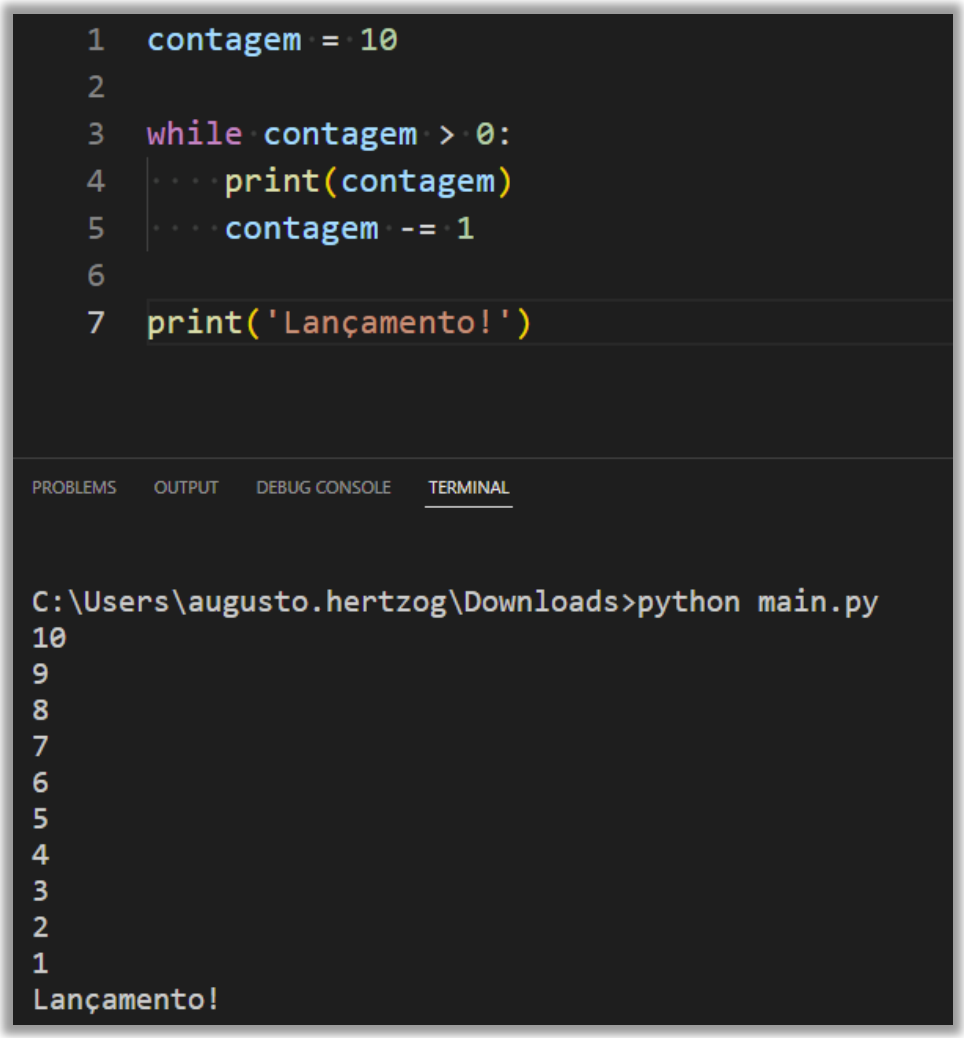
Também podemos usar para pegar um registro do usuário, como um número, por exemplo:



```
num = 0

while num < 10:
    num = int(input('Digite um número maior que 10 : '))

print('Finalmente um número maior que 10!')
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

C:\Users\augusto.hertzog\Downloads>python main.py
Digite um número maior que 10 : 1
Digite um número maior que 10 : 2
Digite um número maior que 10 : 4
Digite um número maior que 10 : 5
Digite um número maior que 10 : 1
Digite um número maior que 10 : -111
Digite um número maior que 10 : 10
Finalmente um número maior que 10!
```

Usando para contagem regressiva:



```
contagem = 10

while contagem > 0:
    print(contagem)
    contagem -= 1

print('Lançamento!')
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

C:\Users\augusto.hertzog\Downloads>python main.py
10
9
8
7
6
5
4
3
2
1
Lançamento!
```

Comando break

O comando “break” é usado em loops (while e for) para interromper o loop antes que a condição de parada seja atingida. Isso significa que, quando o comando break é executado dentro de um loop, o loop é imediatamente encerrado e o código subsequente é executado.

Em condições normais, o loop abaixo seria interrompido apenas quando o fosse igual ou maior que 5.

```
1  i = 0
2
3  while i < 5:
4      print('i =', i)
5      i += 1
6  print('fim do loop')
```

```
PROBLEMS 1K+  OUTPUT  DEBUG CONSOLE  TERMINAL

C:\Users\augusto.hertzog\Downloads>python main.py
i = 0
i = 1
i = 2
i = 3
i = 4
fim do loop
```

Mas podemos usar o comando “break” para sair de um loop while quando um determinado valor é encontrado.

```
1  i = 0
2
3  while i < 5:
4      if i == 3:
5          break
6      print('i =', i)
7      i += 1
8  print('fim do loop')
```

```
PROBLEMS 1K+  OUTPUT  DEBUG CONSOLE  TERMINAL

C:\Users\augusto.hertzog\Downloads>python main.py
i = 0
i = 1
i = 2
fim do loop
```

Este script imprime os números de 0 a 2 e, em seguida, exibe a mensagem “fim do loop”`. A cada iteração, verifica-se se o valor do contador é igual a 3, se for verdadeiro, o comando “break” é executado e o loop é interrompido, sem imprimir o número 3 e sem continuar as iterações, caso contrário ele imprime o número e continua.

O “break” costuma ser usado em conjunto com “if...elif...else” para sair de um loop quando uma determinada condição é atendida. O exemplo acima ilustra isso, pois o loop é interrompido quando o número encontrado é igual a 3.

É importante lembrar que, quando você usa o `break` para sair de um loop, você perde a possibilidade de continuar processando os dados restantes dentro do loop, então é importante utilizar este comando com cuidado e ter certeza de que é a melhor opção para a tarefa específica.

Comando continue

O comando “continue” é usado em loops (“while” e “for”) para pular uma iteração específica e continuar com a próxima. Isso significa que, quando o comando “continue”` é executado dentro de um loop, a iteração atual é interrompida e o código subsequente dentro do loop não é executado, mas o loop continua com a próxima iteração.

Por exemplo, você pode usar o comando “continue” para pular números pares em um loop:

```python
numeros = list(range(10))
print(numeros)

indice = 0

while indice < len(numeros):
    if numeros[indice] % 2 == 0:
        indice += 1
        continue
    print(numeros[indice])
    indice += 1

print('Loop while finalizado!')
```

```
PROBLEMS 1K+  OUTPUT  DEBUG CONSOLE  TERMINAL

C:\Users\augusto.hertzog\Downloads>python main.py
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
1
3
5
7
9
Loop while finalizado!
```

Este script imprime somente os números ímpares de 1 a 9 e, em seguida, exibe a mensagem `Loop while finalizado!`. A cada iteração, verifica-se se o valor do contador é par, se for verdadeiro, o comando `continue` é executado e o código pula para a próxima iteração, sem imprimir o número, caso contrário ele imprime o número e continua.

Assim como o break, o “continue” costuma ser usado em conjunto com “if...elif...else” para pular uma iteração quando uma determinada condição é atendida. O exemplo acima ilustra isso, pois o código pula as iterações quando o número encontrado é par.

É importante lembrar que, quando você usa o “continue” para pular uma iteração, você perde a possibilidade de processar os dados dentro daquela iteração específica, então é importante utilizar este comando com cuidado e ter certeza de que é a melhor opção para a tarefa específica.

break e continue

Abaixo há um exemplo de como usar os comandos break e continue em apenas um while loop.

```python
i = 0

while i < 20:
    if i % 2 == 0:
        i += 1
        continue

    if i == 15:
        break

    print('ímpar :', i)
    i += 1
print('Loop while finalizado!')
```

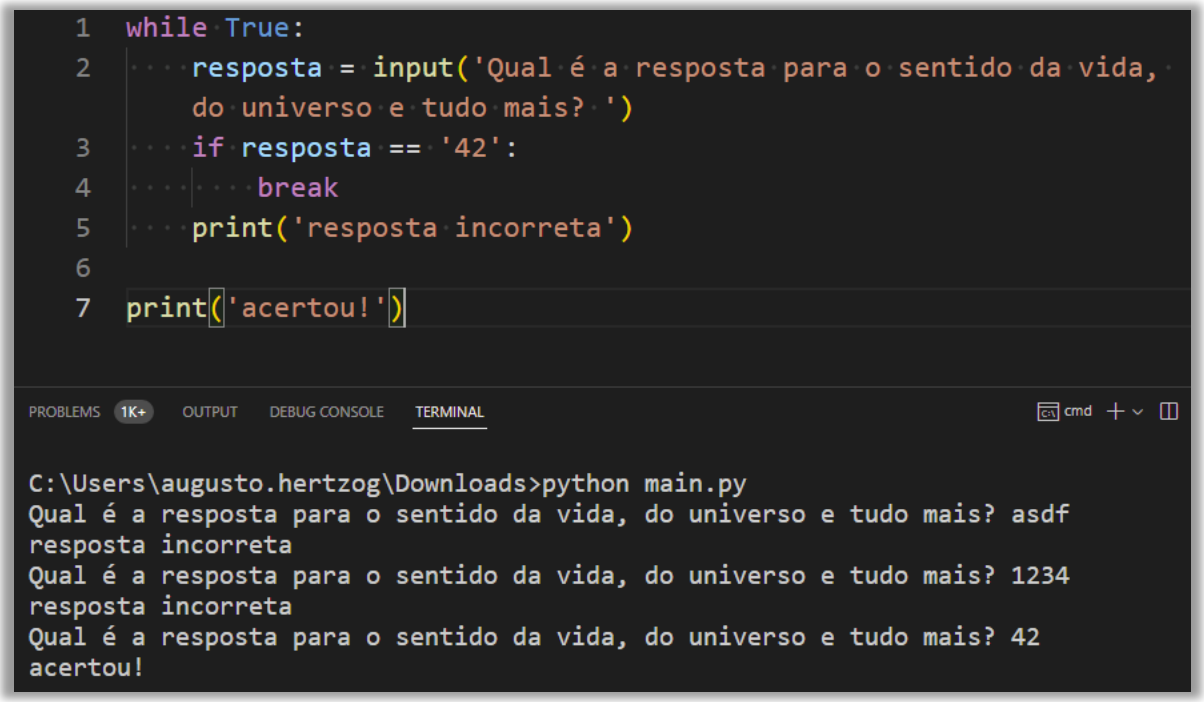
```
PROBLEMS 1K+  OUTPUT  DEBUG CONSOLE  TERMINAL

C:\Users\augusto.hertzog\Downloads>python main.py
ímpar : 1
ímpar : 3
ímpar : 5
ímpar : 7
ímpar : 9
ímpar : 11
ímpar : 13
Loop while finalizado!
```

Loops Infinitos

Um loop infinito é caracterizado por nunca ter fim. Ele será executado até o Dia do Julgamento ao Som das Trombetas.... ou até que o interpretador Python seja interrompido.

Há algumas formas de realizar isso com o while, a mais elegante é tornar a condição de parada sempre verdadeira usando “True”.

Mas podemos nos aproveitar disso. Podemos usar para que um determinado bloco de código seja repetido até que uma determinada condição seja satisfeita.



Editores de Código

Para o desenvolvimento dos códigos em aula, estamos usando o VS Code, conforme tem no material.
Mas também há outros editores de texto que podem ser usados. Abaixo há uma lista dos editores alternativos:

- Bloco de Notas do Windows
  - O próprio editor de texto do Windows pode ser usado para editar código, mas você não terá acesso a verificações de código, sugestões de códigos;
- Notepad++
  - Editor muito usado, principalmente por ser compatível com edições antigas do Windows;
  - https://notepad-plus-plus.org/
- Sublime Text
  - Outro editor que já foi muito usado, mas hoje em dia não é mais tanto;
  - https://www.sublimetext.com/
- Atom
  - O Atom já foi um grande editor de código do GitHub, mas desde a compra do GitHub pela Microsoft, esse editor foi substituído pelo VS Code, mas ele ainda pode ser usado com algumas limitações;
  - https://github.com/atom/atom
- Vim
  - É um editor altamente configurável, mas pode ser muito complicado para quem está aprendendo;
  - https://www.vim.org/download.php

Exercícios para Entregar

1. Crie um programa que peça ao usuário para digitar uma palavra e imprima a palavra em formato de escada crescente usando um loop while.

Exemplo: se o usuário digitar "Gandalf", deve ter a saída:
G
Ga
Gan
Gand
Ganda
Gandal
Gandalf

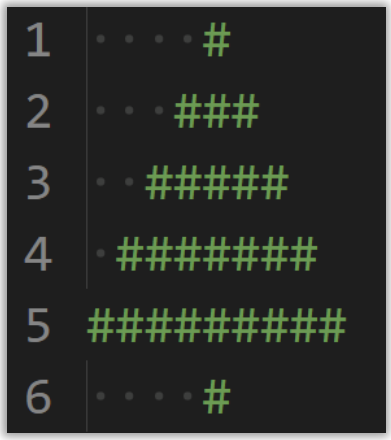
2. Crie um programa que peça um texto ao usuário e converta cada letra em seu respectivo número.

Exemplo: o usuário digitou "a nao", vai ser convertido para "0100140115", onde o "01" representa o "a", o "14" representa o "n", o "15" representa o "o" e o “00” representa o espaço. Desconsidere palavras acentuadas, números e pontuações.

3. Árvore de Natal !

Crie um algoritmo que construa uma árvore de Natal.
Seu programa deve perguntar ao usuário que altura ele gostaria a árvore. Depois de responder, seu programa deve desenhar a árvore com a altura desejada pelo usuário.

Abaixo há um exemplo de uma árvore de altura “5”:



Dicas :

- para o desenvolvimento, fixe o valor 5 para sua árvore, deixe para pedir o tamanho apenas no final;
- você terá que usar 1 while e 3 for loops;
- abaixo há uma lista do número de espaço e do número de “#” para uma árvore de tamanho 5 :
    - 4 - 1
    - 3 - 3
    - 2 - 5
    - 1 - 7
    - 0 - 9
    - 4 - 1 (tronco)
- passos do seu programa :
    - decrementar os espaços em 1 a cada repetição;
    - incrementar 1 “#” em 2 a cada repetição;
    - reservar 1 linha para o tronco da árvore;
    - decrementar a altura da árvore até chegar em 0;
    - imprimir os espaços e então a “#” para cada linha;
    - imprimir os espaços e depois 1 “#” para o tronco;

Exercícios para Praticar

1. Imprima números de 1 a 10 usando um loop while.
2. Peça ao usuário para digitar um número e imprima todos os números de 0 até o número digitado.
3. Peça ao usuário para digitar uma palavra e imprima cada letra em uma linha usando um loop while.
4. Peça ao usuário para digitar uma palavra e imprima a palavra ao contrário usando um loop while.
5. Peça ao usuário para digitar um número e imprima a soma de todos os números de 1 até o número digitado.
6. Peça ao usuário para digitar um número e imprima todos os números pares de 0 até o número digitado.
7. Peça ao usuário para digitar um número e imprima todos os números ímpares de 0 até o número digitado.
8. Peça ao usuário para digitar uma palavra e imprima o número de letras da palavra usando um loop while.
9. Imprima a tabuada do número 5 usando um loop while.
10. Peça ao usuário para digitar um número e imprima se ele é primo ou não usando um loop while.
11. Peça ao usuário para digitar uma palavra e imprima apenas as vogais da palavra usando um loop while.
12. Peça ao usuário para digitar uma palavra e imprima apenas as consoantes da palavra usando um loop while.
13. Imprima todos os números de 100 até 0 em ordem decrescente usando um loop while.
14. Peça ao usuário para digitar um número e imprima todos os números pares de 1 até o número digitado usando um loop while.
15. Peça ao usuário para digitar um número e imprima a soma de todos os números pares de 1 até o número digitado usando um loop while.
16. Peça ao usuário para digitar um número e imprima todos os números ímpares de 1 até o número digitado usando um loop while.
17. Peça ao usuário para digitar um número e imprima a soma de todos os números ímpares de 1 até o número digitado usando um loop while.
18. Peça ao usuário para digitar um texto e imprima a primeira letra de cada palavra usando um loop while.
19. Peça ao usuário para digitar um número e imprima a sequência de Fibonacci até o número digitado usando um loop while.
20. Peça ao usuário para digitar uma palavra e imprima a palavra sem as vogais usando um loop while.
21. Peça ao usuário para digitar uma palavra e imprima a palavra sem as consoantes usando um loop while.
22. Peça ao usuário para digitar uma palavra e imprima cada letra duas vezes usando um loop while. Exemplo: se o usuário digitar "casa", deve mostrar "ccaassaa".
23. Peça ao usuário para digitar uma palavra e imprima a palavra com cada letra alternando entre maiúscula e minúscula usando um loop while. Exemplo: se o usuário digitar "gandalf", a saída deve ser "GaNdAlF".
24. Gere uma lista de números aleatória (módulo random) e imprima a lista em ordem crescente usando um loop while. Não use os métodos de ordenação built-in do Python.
25. Gere uma lista de números aleatória (módulo random) e imprima a lista em ordem decrescente usando um loop while. Não use os métodos de ordenação built-in do Python.
26. Peça ao usuário para digitar uma palavra e imprima a palavra ao contrário usando um loop while e uma lista.
27. Gere uma lista de números aleatória (módulo random) e imprima apenas os números pares da lista usando um loop while.
28. Gere uma lista de números aleatória (módulo random) e imprima apenas os números ímpares da lista usando um loop while.
29. Gere uma lista de números aleatória (módulo random) e imprima a média dos números usando um loop while.
30. Gere uma lista de números aleatória (módulo random) e imprima a soma dos números usando um loop while.
31. Gere uma lista de números aleatória (módulo random) e imprima o maior número da lista usando um loop while.
32. Gere uma lista de números aleatória (módulo random) e imprima o menor número da lista usando um loop while.
33. Gere uma lista de números aleatória (módulo random) e imprima apenas os números que são múltiplos de 3 usando um loop while.

34. Gere uma lista de números aleatória (módulo random) e imprima apenas os números que são múltiplos de 5 usando um loop while.
35. Peça ao usuário para digitar uma palavra e imprima apenas as letras que aparecem mais de uma vez na palavra usando um loop while.
36. Gere uma lista de números aleatória (módulo random) e imprima a lista sem números duplicados usando um loop while.
37. Escreva um programa que imprima os números de 1 a 10, pulando o número 7 usando o continue.
38. Escreva um programa que solicita ao usuário que digite um número e verifica se o número é par ou ímpar. Se for ímpar, o programa deve continuar solicitando ao usuário que digite um número até que um número par seja inserido.
39. Escreva um programa que solicita ao usuário que digite uma senha. O programa deve continuar solicitando ao usuário que digite a senha até que a senha correta seja inserida. Use o break para sair do loop quando a senha correta for inserida.
40. Escreva um programa que solicita ao usuário que digite um número. O programa deve continuar solicitando ao usuário que digite um número até que um número divisível por 7 seja inserido.
41. Escreva um programa que solicita ao usuário que digite uma palavra. O programa deve continuar solicitando ao usuário que digite uma palavra até que a palavra "fim" seja inserida. Use o break para sair do loop quando a palavra "fim" for inserida.
42. Escreva um programa que imprima os números de 1 a 20. Se o número for divisível por 3, o programa deve imprimir "Fizz" em vez do número. Se o número for divisível por 5, o programa deve imprimir "Buzz" em vez do número. Se o número for divisível por 3 e 5, o programa deve imprimir "FizzBuzz" em vez do número. Use o continue para evitar a impressão de números que são divisíveis por 3 ou 5.
43. Escreva um programa que solicita ao usuário que digite uma palavra. O programa deve imprimir a palavra invertida, mas parar de imprimir assim que chegar na letra "a". Use o break para sair do loop quando a letra "a" for encontrada.
44. Escreva um programa que solicita ao usuário que digite um número. O programa deve continuar solicitando ao usuário que digite um número até que um número maior que 100 seja inserido. Use o break para sair do loop quando um número maior que 100 for inserido.
45. Escreva um programa que crie uma lista de números aleatórios (módulo random). O programa deve imprimir os números, mas parar de imprimir assim que encontrar um número negativo. Use o break para sair do loop quando um número negativo for encontrado.
46. Escreva um programa que imprima os números pares de 1 a 10 usando o continue.
47. Escreva um programa que solicita ao usuário que digite um número. O programa deve continuar solicitando ao usuário que digite um número até que um número entre 1 e 10 seja inserido. Use o continue para evitar a entrada de números que estão fora desse intervalo.
48. Escreva um programa que crie uma lista de números aleatórios (módulo random). O programa deve imprimir apenas os números ímpares da lista. Use o continue para evitar a impressão de números pares.
49. Escreva um programa que solicita ao usuário que digite uma palavra. O programa deve imprimir a palavra em ordem inversa, mas pular as vogais usando o continue.
50. Escreva um programa que solicita ao usuário que digite um número. O programa deve continuar solicitando ao usuário que digite um número até que um número entre 1 e 100 seja inserido. Use o continue para evitar a entrada de números que estão fora desse intervalo.
51. Escreva um programa que solicita ao usuário que digite uma lista de palavras. O programa deve imprimir cada palavra em uma linha separada, mas pular as palavras que contêm a letra "a" usando o continue.
52. Escreva um programa que crie uma lista de números aleatórios (módulo random). O programa deve imprimir apenas os números pares da lista. Use o continue para evitar a impressão de números ímpares.
53. Escreva um programa que solicita ao usuário que digite um número. O programa deve continuar solicitando ao usuário que digite um número até que um número entre 1 e 10 seja inserido. Quando um número válido for inserido, o programa deve imprimir "Número válido" e sair do loop usando o break.
54. Escreva um programa que crie uma lista de números aleatórios (módulo random). O programa deve imprimir cada número multiplicado por 2, mas pular os números ímpares usando o continue.

55. Escreva um programa que solicita ao usuário que digite um número. O programa deve continuar solicitando ao usuário que digite um número até que um número divisível por 5 seja inserido. Quando um número válido for inserido, o programa deve imprimir "Número válido" e sair do loop usando o break.
56. Escreva um programa que crie uma lista de números aleatórios (módulo random). O programa deve imprimir apenas os números maiores que 10. Use o continue para evitar a impressão de números menores ou iguais a 10.
57. Escreva um programa que solicita ao usuário que digite uma sequência de números positivos. O programa deve imprimir a soma dos números inseridos até que o usuário insira um número negativo, momento em que o programa deve parar usando o break.
58. Escreva um programa que solicita ao usuário que digite uma palavra ou frase. O programa deve imprimir o número de vezes que cada letra aparece na palavra ou frase, mas pular a letra "e" usando o continue.
59. Escreva um programa que solicita ao usuário que digite uma lista de números. O programa deve imprimir o maior número da lista usando o break assim que encontrar o primeiro número maior que 10.
60. Escreva um programa que solicita ao usuário que digite uma sequência de números positivos. O programa deve imprimir a média dos números inseridos até que o usuário insira um número negativo, momento em que o programa deve parar usando o break.
61. Escreva um programa que solicita ao usuário que digite uma lista de palavras. O programa deve imprimir cada palavra em uma linha separada, mas pular as palavras que contêm a letra "o" usando o continue.
62. Escreva um programa que solicita ao usuário que digite uma lista de números. O programa deve imprimir apenas os números que são divisíveis por 3 e 5 usando o continue.
63. Escreva um programa que solicita ao usuário que digite uma sequência de números positivos. O programa deve imprimir a soma dos números inseridos até que o usuário insira um número maior que 100, momento em que o programa deve parar usando o break.
64. Escreva um programa que solicita ao usuário que digite uma lista de números. O programa deve imprimir a média dos números inseridos, mas pular os números negativos usando o continue. Se o usuário inserir um número maior que 100, o programa deve sair do loop e imprimir uma mensagem de erro usando o break.
65. Escreva um programa que solicita ao usuário que adivinhe um número entre 1 e 10. O programa deve permitir que o usuário faça até 3 tentativas. Se o usuário adivinhar corretamente em qualquer tentativa, o programa deve imprimir "Parabéns, você acertou!". Caso contrário, o programa deve imprimir "Suas tentativas acabaram. O número correto era X", onde X é o número correto.