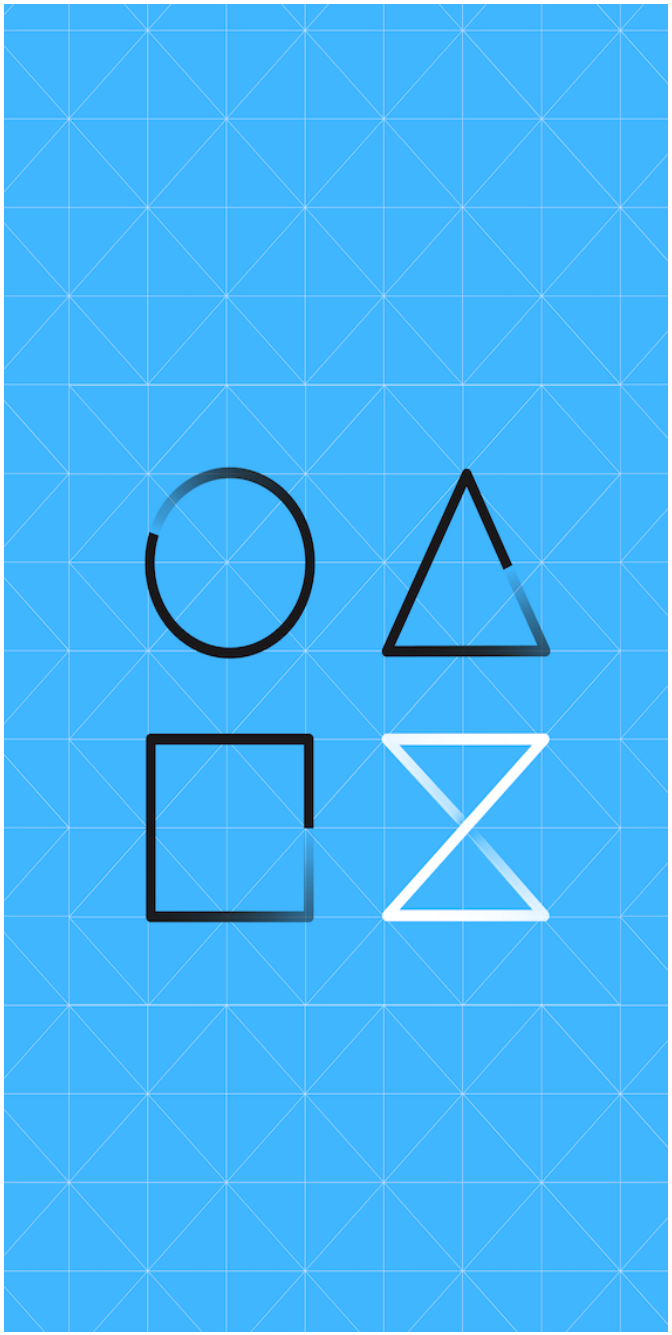


Fundamentos

240630-2221

CODE1

Faça a sua própria diversão

Tradução automatizada pelo Watson Language Translator

240630-2221

PYTHON É UMA ÓTIMA LINGUAGEM PARA SE TER EM Z

O Desafio

Você será apresentado aos conceitos básicos de codificação aqui. Ou, se você já estiver familiarizado com variáveis, loops e instruções 'if', você estará rapidamente recebendo uma atualização e aprendendo que não importa em qual plataforma você está, o código Python se parece com o código Python.

Você estará usando o Python A linguagem de programação, como é talvez a linguagem mais amplamente utilizada para o novo desenvolvimento agora, e torna muito fácil começar com um código simples que não irá sobrecarregá-lo se esta for a sua primeira codificação.

Antes De Começar

Tudo o que você precisa para começar é o acesso ao USS shell, por meio de um SSH ou uma janela do terminal.

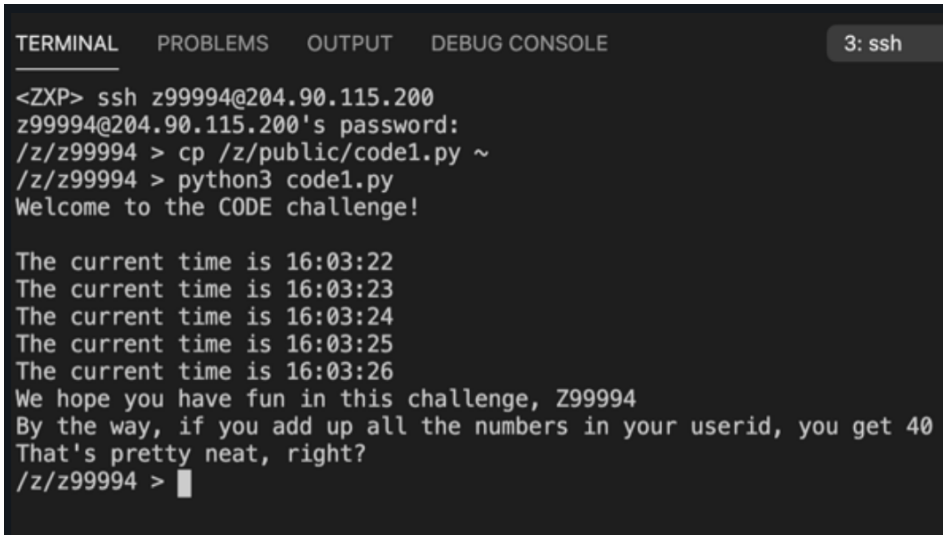
O único desafio técnico que você precisa ter concluído antes que este seja VSC1.

Investimento

Etapas	Duração
12	120 minutos

CODE1240630-2221

1 SE VOCÊ QUISER CODIFICAR, ENTÃO ...



Efetue login no IBM z/OS sistema usando ssh

Verifique seu diretório inicial para um arquivo chamado "code1.py" (usando o ls comando); se você não tiver um, copie o original **code1.py** programa a partir de **/z/público** , copiando e colando por meio de VSCode ou inserindo o comando a seguir:

cp /z/public/code1.py ~

(conforme mostrado na captura de tela acima)

Em seguida, execute o programa com

python3 code1.py

Isso diz " Use o Python interpretador para executar o programa code1.py neste diretório " e olhar para a saída.

Como você pode ver, há muitas coisas acontecendo aqui; no próximo passo, você vai dar uma olhada "sob o capô" e ver o que está acontecendo por dentro.

2 VAMOS VER O QUE ESTÁ DENTRO

Abra o código em uma janela do editor dentro VSCode.

Há uma pequena quantidade de código para ver aqui, e muitos comentários (essas são as linhas em verde começando com #)

```
1   #!/usr/bin/python3
2
3   # Here, we're importing a few modules.
4   # We're using datetime so we can tell what time it is.
5   # We're using time so we can use "sleep"
6   # We're using getpass to get your userid
7   from datetime import datetime
8   import time
9   import getpass
10
11  #This is a typical print message
12  #Wherever we put \n, there will be an extra newline, wh
13  #can be helpful for formatting and making things look r
14  print("Welcome to the CODE challenge!\n")
15
```

Mesmo se você nunca programou antes, você pode ver a parte onde

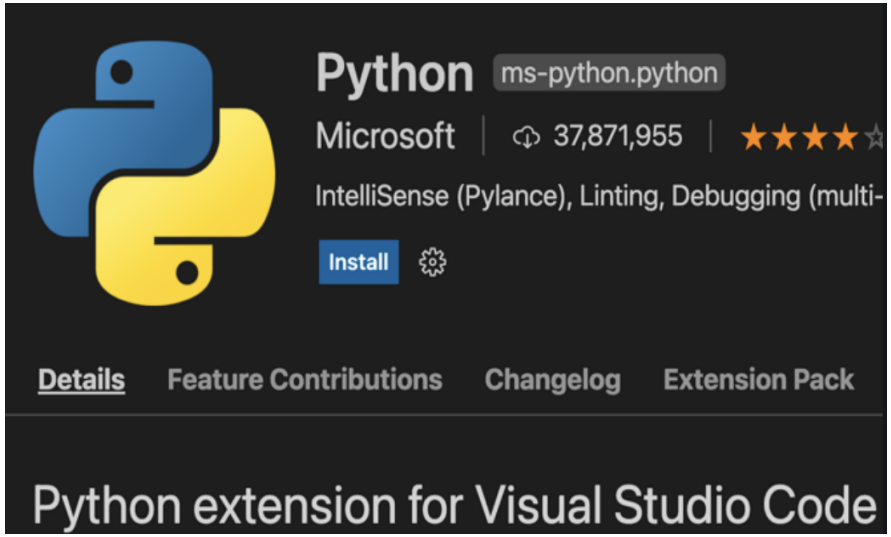
- conta para trás a partir de 5
- lê seu ID do usuário
- sabe que horas são
- e como ela pára por um segundo entre cada seção.

Você pode achar algumas dessas informações úteis ao lidar com etapas posteriores. **(Isso é uma dica, bem ali!)**

3 UMA PALAVRA SOBRE EXTENSÕES

Extensões em VSCode são uma boa maneira de ganhar funções adicionais, mas também podem fazer com que as coisas não funcionem como esperado.

Você pode receber uma mensagem de VSCode perguntando se deseja instalar a extensão Python.



Se você tiver um Python instalação, você pode tentar continuar usando-o, mas esteja ciente de que você provavelmente receberá mensagens de aviso em mais tarde Python Desafios relacionados.

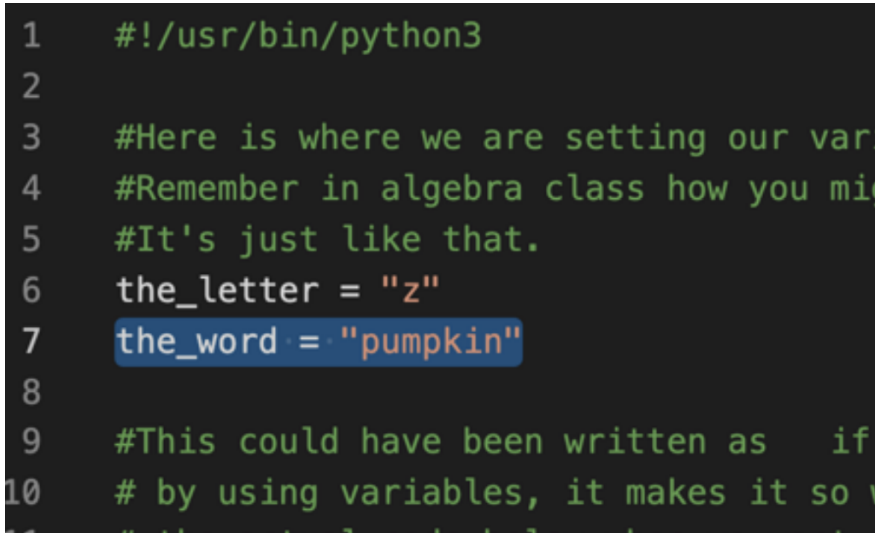
Um a ser observado em particular é uma mensagem de erro informando que o módulo 'zoautil_py' não pode ser encontrado-isso é *NÃO* um problema para você trabalhando nesses desafios.

Se você quiser que seu editor se pareça com as capturas de tela aqui, não instale nenhuma extensão além das descritas nos desafios.

Tudo bem? Vamos seguir em frente.

4 UMA ROSA POR QUALQUER OUTRO NOME

Certifique-se de ter uma cópia de over code2.py em seu diretório inicial.



Este programa Python leva uma palavra e descobre se a letra "z" está nele.

Observe como 'the_word' para definir como "pumpkin" na linha 7.

Isso é criar o que é conhecido como uma variável.

Uma variável é um marcador para um valor em um programa e pode ser letras, números ou realmente qualquer tipo de dados.

Neste caso, ele é usado para representar a palavra que você deseja "investigar"-para verificar se ele contém um "z".

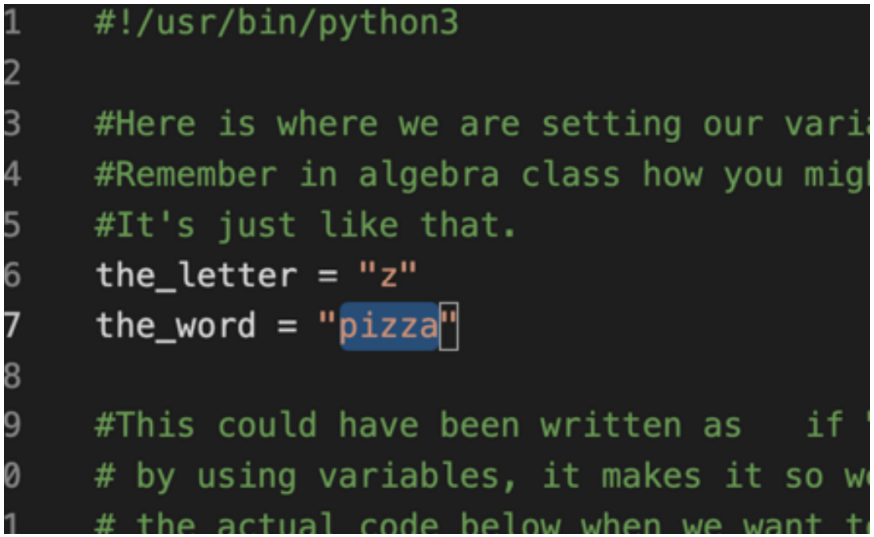
O programa também possui uma variável para manter o valor da letra específica usada para a investigação.

Leia o programa para entender o que ele deve fazer; em seguida, execute o programa e veja o que acontece (use `python3 code2.py`).

5 UM POUCO DE ANTICLIMAX?

Nada acontece, certo?

Vamos consertar isso!



Edite o código para que, em vez de "pumpkin", a palavra variável seja configurada como "pizza", salve o arquivo e * execute o programa novamente ** em seu USS sessão de terminal *

Você deve ver uma mensagem confirmando que " pizza " De fato tem a letra " z " dentro dele.

Sinta-se livre para experimentar outras palavras e letras e, em seguida, olhar para a parte do código que faz a verificação.

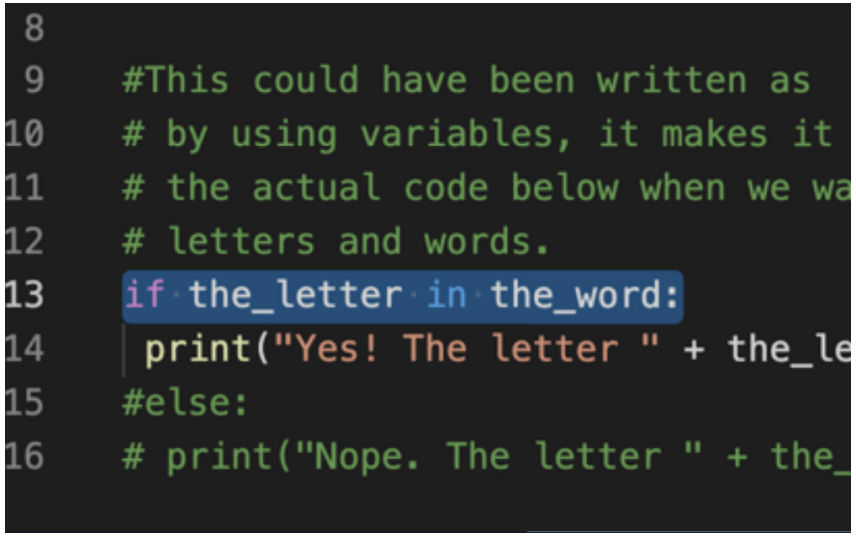
Nota : Certifique-se de salvar seu arquivo entre as edições (usando a opção de menu do editor Arquivo-> Salvar ou usando o atalho do teclado) para confirmar quaisquer mudanças que você fizer e lembre-se de executar o código em sua sessão de terminal USS.

6 BLOCOS PARA MELHOR

Uma grande parte da escrita e compreensão do código é descobrir a lógica de um programa.

Em outras palavras, o modo como ela flui.

O código do programa geralmente pode ser organizado em blocos de código que podem ser executados em ordem, ou se uma condição for atendida



Este programa usa um ' if ' instrução na linha 13.

Ele está testando para ver se 'the_letter' está em 'the_word '.

Um ' if ' a instrução tem a condição sendo testada, seguida por dois pontos (:) e o código que será executado, quando essa condição for satisfeita, está em um "bloco" abaixo dele.

Um bloco é um grupo de instruções de programa que todos pertencem juntos

O bloco de código é "recuado" (movido para a direita), então você sabe que qualquer coisa nesse bloco é o que vai acontecer se a condição for atendida.

POR QUE PRECISAMOS DE MAIS DE UMA LÍNGUA? NÃO POSSO APRENDER UM?

Línguas diferentes são melhores em coisas diferentes.

COBOL é uma linguagem criada para transações de alta precisão e baixa latência

C e C++ são bons para utilitários do sistema e aplicativos de alto desempenho que se beneficiam de muito controle sobre memória e outros recursos do sistema.

Python é uma linguagem extensível que existe há muito tempo e é usada para muitas tarefas diferentes. Você pode chamá-lo de uma linguagem de programação de propósito geral, como você pode usá-lo para

- IA (Inteligência Artificial)
- simulações matemáticas
- aplicativos da web
- Jogos
- automação
- apenas sobre qualquer coisa.

Se você vai aprender apenas uma linguagem de programação, faz sentido que ela seja Python.

A partir daí, uma vez que você tenha os fundamentos compreendidos, é apenas uma questão de aprender quaisquer outros pontos específicos da linguagem de programação.

Código em!

7 OU ENTÃO!

```
7   the_word = "pumpkin"
8
9   #This could have been written as
10  # by using variables, it makes it s
11  # the actual code below when we war
12  # letters and words.
13  if the_letter in the_word:
14      print("Yes! The letter " + the_let
15  else:
16      print("Nope. The letter " + the_le
```

As linhas 15 e 16 no code2.py adicionam outro passo ao ' if ' instrução-um ' else ' cláusula.

Se a condição que está sendo testada na linha 13 não for atendida (não há 'z' na palavra), o programa pode dizer ou fazer algo sobre isso.

Neste momento, estas linhas de código são comentadas, por isso não fazem nada.

Excluir as marcas de hash (#) no início das linhas no ' else ' , salve o arquivo e, em seguida, execute-o novamente.

Anote a indentação-o ' else ' deve alinhar com o 'if', e a próxima linha (ou bloco ou linhas) deve ser indentada para mostrar que ela pertence ao ' else ' acima dele.

Teste-o e certifique-se de que funciona para palavras 'z' e não 'z'.

8 NÃO PERCA SEUS BOLINHAS DE GUDE

Em sua sessão de terminal USS , tente executar o programa 'marbles.py' do seu diretório inicial.

Espero que isso seja bem simples.

Abra o arquivo no editor para efetuar check-out do código.

```
#!/usr/bin/python3

marbles = 10  #You start out with 10 marbles
marble_dots = "**********" #Pretend these are ten mar

while (marbles > 0):

 #This line of code prints out the "marble_dots" vari
 # but will only include the number of characters in
 # for how many marbles are left.
 print(marble_dots[:marbles])

 #This prints out how many marbles you have left.
 # We have to say str(marbles) because marbles is a n
```

Uma coisa importante a notar é como a mensagem sobre quantos mármores são deixados aparece no código apenas uma vez, mas é impresso 10 vezes.

Isto é porque há um ' while loop ' começando na linha 6.
A ' while loop ' diz, "Continue fazendo esse conjunto de ações, contanto que a condição a seguir seja verdadeira".

Neste caso, a condição é verdadeira, desde que haja mais de 0 mármores restantes.

9 A CONTAGEM REGRESSIVA FINAL

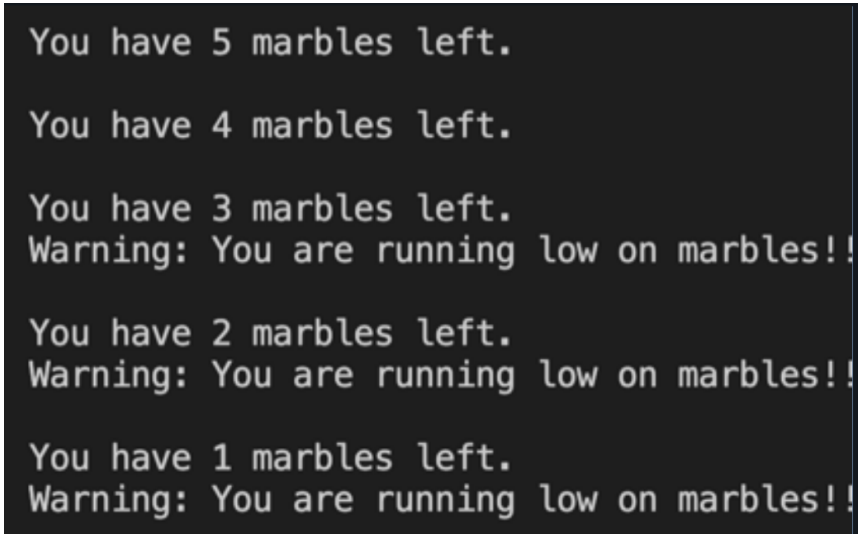
Depois de ter rastreado o seu caminho através do código 'marbles.py', dê uma olhada nas linhas 13 e 14.

Isso parece um recurso especial que você pode ativar removendo o comentário.

Apague as marcas de comentário para que o código esteja ativo e execute o programa novamente

O código foi *era suposto* para emitir uma mensagem de aviso quando há três ou menos bolinhas de gude, mas algo está errado com a lógica em que ' if ' instrução.

OH NO !!!



Veja se você pode corrigi-lo para que a mensagem seja impressa como a captura de tela acima-apenas quando restarem 3 ou menos bolinhas de gude.

Algumas dicas para você começar:

- > = significa "maior ou igual a"
- <significa "menor que"

Considere onde isso " if " é em relação à parte do código que subtrai um mármore.

Hmmm ...

Há muitas maneiras diferentes de corrigir esse código.

"DE QUE OUTRA FORMA POSSO CONTROLAR O CÓDIGO?"

Você tem realmente apenas skimmed a superfície do Python programação neste desafio, o suficiente para dar a alguém sem experiência de programação uma chance de verificar o que a programação parece com muitos exemplos.

Se você está interessado e quer aprender mais, certifique-se de ler mais sobre o que você pode fazer com Python em Desenvolvedor IBM .

Há vídeos, tutorias, podcasts, padrões de código e muitos projetos que você pode começar, não importa qual seja o seu nível de habilidade atual.

Mesmo se você não planeja ser um codificador em tempo integral, é importante ter alguma familiaridade com o código para que você possa corrigir problemas quando eles aparecerem, e talvez até mesmo adicionar recursos ao código que alguém escreveu.

10 EU SOU UM APRENDIZ VISUAL

Os números são bons, mas às vezes é melhor visualizar as coisas.

Sua próxima tarefa é fazer com que o programa imprima o número de bolinhas de gude deixadas como asteriscos. Então, quando restou 5 bolinhas de gude, você imprime *****

Como você vai fazer isso?

Uma das variáveis no programa ' marble_dots ' é apenas dez asteriscos em uma fileira.

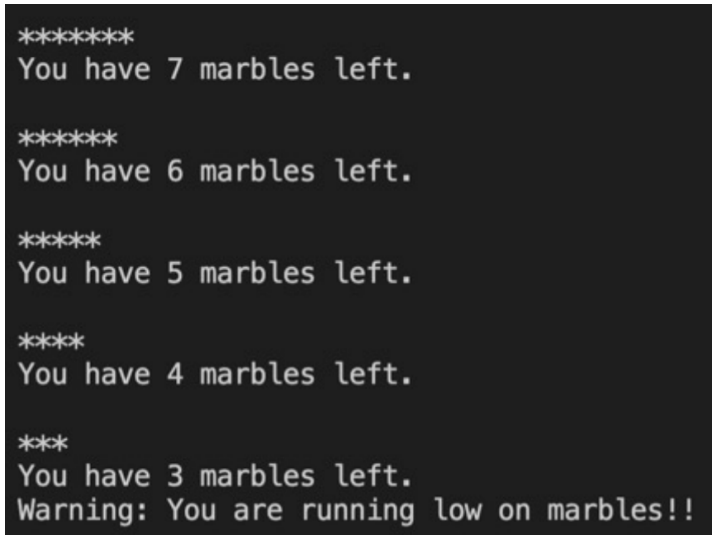
Talvez você estivesse curioso para o que era isso.

A seguinte linha de código fará com que o programa imprima seis pontos de mármore.

```
print(marble_dots[:6])
```

Se você substituir o '6' por um nome de variável, ele imprimirá o número de asteriscos para qualquer que seja a variável definida. (Supondo que a variável tenha sido configurada para um número!)

As rodas da sua mente já estão girando?



Descubra onde você deseja colocar essa linha de código e como fazê-la funcionar para que o resultado se pareça com a saída na captura de tela acima.

11 TUDO FORA DE MÁRMORES

```
***
You have 3 marbles left.
Warning: You are running low on marbles!!

**
You have 2 marbles left.
Warning: You are running low on marbles!!

*
You have 1 marbles left.
Warning: You are running low on marbles!!

You are all out of marbles
```

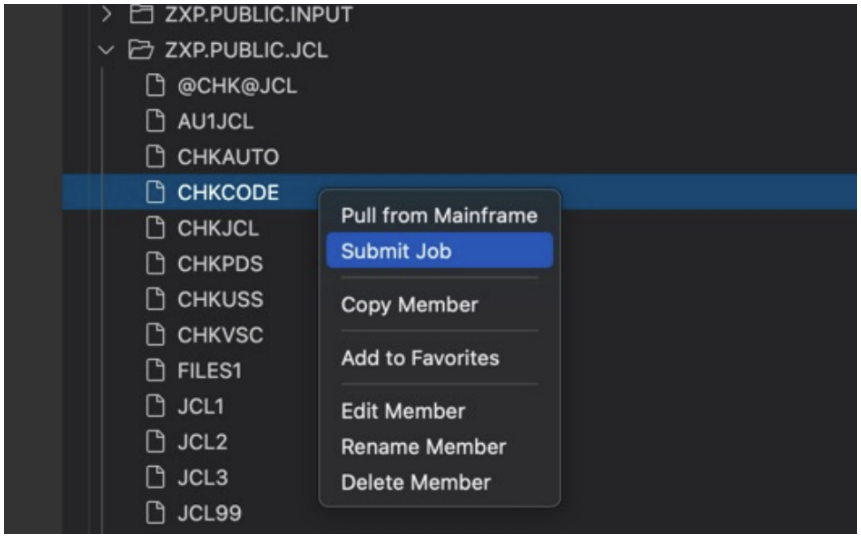

O usuário precisa saber quando o programa é feito, e que não há mais mármores.

Faça a mensagem dizer "You are all out of marbles", certifique-se de que ele apareça por último, e apenas uma vez em seu programa

12 VERIFICAÇÃO DUPLA; ENVIAR

Agora é hora de executar o CHKCODE tarefa de ZXP.PUBLIC.JCL e veja se você tem um programa que produz a mesma saída que a captura de tela anterior.

Certifique-se de ter o número certo de "mármores" exibido, que o espaçamento parece certo, e que sua mensagem final "todos fora de mármores" parece exatamente como faz na captura de tela.



A tarefa de validação executará seu programa 'marbles.py', portanto, certifique-se de que tudo esteja funcionando conforme descrito.

Isso é apenas o começo; você vai fazer ainda mais com Python em desafios posteriores, então espero que você tenha gostado disso.

Bom trabalho-vamos recapitular	Em seguida ...
Na maioria das coisas, a parte mais difícil é começar. Se essa foi a primeira vez que você hackear o código, pode ter se sentido difícil, frustrante e até um pouco estranho. Escolhemos um exemplo que ilustraria alguns dos conceitos mais importantes: + variáveis + loops + instruções if/then/else + modificando variáveis + usando funções de biblioteca. Concluir esse desafio significa que você é capaz de visualizar um problema, resolvê-lo mentalmente e, em seguida, implementar a solução em código.	Agrupando Fundamentos