

04

## Dividindo pontos

Vimos no vídeo como calcular a pontuação do jogador. O cálculo foi simples e usamos a função `abs()`:

```
pontos_perdidos = abs(chute - numero_secreto) #pontos perdidos da rodada  
Pontos = pontos - pontos_perdidos #subtraindo os pontos perdidos da pont
```

Vamos mudar um pouco esse cálculo dos pontos perdidos. Agora não basta apenas calcular o valor absoluto, devemos também dividir esse valor por 3.

```
pontos_perdidos = abs(chute - numero_secreto) / 3 #dividindo por três
```

Nada complicado aqui, mas pode surgir uma outra questão. Imagine que o chute é 21 e o número secreto é 32. Isso daria a seguinte conta:

```
pontos_perdidos = abs(21 - 32) / 3 #dividindo por três
```

Que é:

```
pontos_perdidos = 11 / 3
```

Será que o Python consegue dividir 11 por 3? Se sim, qual será o tipo da variável `pontos_perdidos`? Faça o teste!

A Não consegue e dá erro.



B Consegue sim e o tipo será `int`.



C Consegue sim e o tipo será `number`.



Consegue sim e o tipo será `float`.



Correto!

Você pode testar isso facilmente no console do Python3:

```
>>> pontos_perdidos = 11 / 3  
>>> type(pontos_perdidos)  
<class 'float'>
```

Agora temos um outro problema. Ao imprimir esse cálculo, percebemos muitas casas decimais:

```
>>> print(pontos_perditos)
3.6666666666666665
```

Bom, já aprendemos a formatar números decimais (lembre-se do `{:7.2f}`), mas nesse caso não é uma questão de formatação! Gostaríamos de arredondar o número, ou seja o valor decimal **3.66666** deve ser transformado para o valor **4**.

Será que você se lembra da função *built-in* para o arredondamento?

[PRÓXIMA ATIVIDADE](#)