

Para saber mais: arredondar no Python 2 e no Python 3

Nos exercícios, perguntamos como arredondar valores. Olhando nas funções *built-in* do Python, encontramos a função `round`, por exemplo:

```
>>> round(3.5)
4
```

O interessante é que a função `round` também mudou entre Python 2 e Python 3!

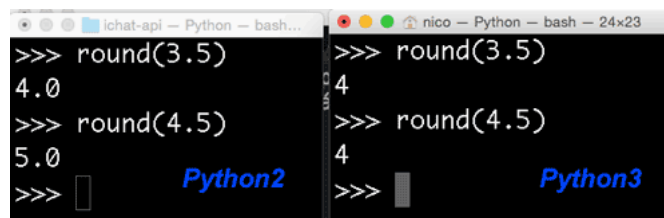
Por exemplo, executando `round(3.5)` no Python 3 dará o valor `4` (tipo inteiro), e com Python 2 dará o valor `4.0` (tipo decimal).



The image shows two side-by-side terminal windows. The left window is titled 'nico - Python - bash - 20x5' and shows the command `>>> round(3.5)` resulting in `4.0`. The right window is titled 'FN-23 - Python - bash - ...' and shows the same command resulting in `4`. Both windows have a blue 'Python' logo at the bottom.

Ou seja, o Python 3 sempre devolve um valor do tipo `int`, enquanto o Python 2 devolve o tipo `float`.

Além disso, arredondando o valor `4.5` com Python 2, dará o valor `5.0` e no Python 3 dará o valor `4`! Veja a imagem abaixo comparando Python 2 com Python 3:



The image shows two side-by-side terminal windows. The left window is titled 'ichat-api - Python - bash...' and shows the commands `>>> round(3.5)` resulting in `4.0` and `>>> round(4.5)` resulting in `5.0`. The right window is titled 'nico - Python - bash - 24x23' and shows the same commands resulting in `4` and `4`. Both windows have a blue 'Python' logo at the bottom.

O Python 3 usa uma forma de arredondar, que também é chamado de *Banker's rounding* e **sempre arredonda para o próximo valor par**.

Mais informações se encontram na documentação do Python 3:

<https://docs.python.org/3.5/library/functions.html#round> (<https://docs.python.org/3.5/library/functions.html#round>).