03

Um bom jeito de economizar

Douglas deseja comprar um celular novo e tem monitorado o preço do celular desejado através de um pequeno script de Python.

Ele salva todos os dias o preço atual do celular que ele deseja, e como ele monitora o preço de várias lojas e já está namorando este celular há muito tempo, ele acabou com uma lista muito grande, veja:

```
precos = [1525,1120,1464,1200,1330,1356,1312,1531,1232, 1234,1250,1114,1553,1147,1303,
```

Recentemente ele esbarrou com uma promoção e o celular está custando R$ 1025. Ele ficou interessado e quer saber se esse é o menor preço que ele já encontrou deste celular.

Qual dos códigos abaixo retorna o melhor preço até agora, para que o Douglas veja, de modo rápido, e decida se vale aproveitar a promoção ?

A

```
print( precos.min )
```

B

```
print(min precos)
```

C

```
print(not max(precos))
```

✓

```
print( min(precos))
```

Correto! A função **min** nos retorna o menor item de uma lista.

Se desejamos descobrir o menor item de uma Lista, podemos utilizar a função **min()**, e passar a lista para ela como parâmetro. Então, no caso do Douglas, para descobrir o menor preço do celular até hoje, faríamos assim:

```
print( min(precos))
```

E teríamos como resposta **1015**, ou seja a promoção de R$ 1025 não valeria a pena!

É importante notar também que só conseguimos utilizar a função **min()** em listas de mesmo tipo, então por exemplo na lista abaixo:

```
precos = [ 1050, 'mil reais', 1020];
```

O Python não vai saber comparar a String `mil reais` com os inteiros numéricos. Então sempre que utilizarmos o **min()** a lista deve possuir todos os elementos do mesmo tipo.

E claro, assim como temos a função **min()** que nos retorna o menor item da Lista, também temos a função **max()** que nos retorna o maior item da mesma.

PRÓXIMA ATIVIDADE