

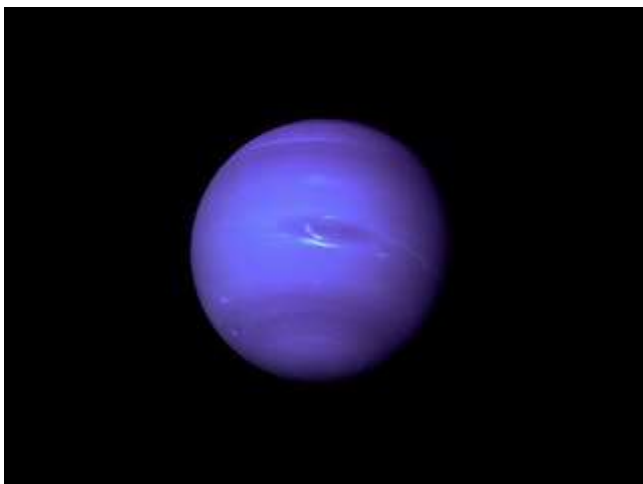
Trabalhar com números em listas

4 minutos

Até agora, você usou nomes de planeta em uma lista. Você deve estar se perguntando como trabalhar com outros tipos de dados, como números.

Você sabia que a gravidade em outros planetas é mais forte ou mais fraca, dependendo da massa e do tamanho do planeta? A gravidade é geralmente medida em *G*. A gravidade da terra é 1 e nos demais planetas ela é medida em [relação à Terra](#) .

A gravidade na Lua é 0,166 *G*, que é o motivo pelo qual os astronautas podem pular tão alto na Lua. A gravidade em Netuno é de 1,12 *G*, portanto, saltar é mais difícil. Até mesmo atletas olímpicos teriam dificuldades de pular acima de 2 metros em Netuno.



Armazenar números em listas

Para armazenar números com casas decimais no Python, use o tipo `float`. Para criar um `float`, insira um número com casas decimais e atribua-o a uma variável:

Python

```
gravity_on_earth = 1.0  
gravity_on_the_moon = 0.166
```

O seguinte código cria uma lista que mostra as forças gravitacionais de todos os oito planetas do sistema solar, em *G*:

Python

```
gravity_on_planets = [0.378, 0.907, 1, 0.377, 2.36, 0.916, 0.889, 1.12]
```

Nessa lista, `gravity_on_planets[0]` é a gravidade de Mercúrio (0,378 G), `gravity_on_planets[1]` é a gravidade em Vênus (0,907 G) e assim por diante.

Na Terra, um ônibus de dois andares pesa 124.054 Newtons (N). Em Mercúrio, onde a gravidade é 0,378 G, o mesmo ônibus pesa 124,054 Newtons multiplicados por 0,378. No Python, para multiplicar dois valores, use o símbolo `*`.

No seguinte exemplo, você pode trabalhar com o peso do ônibus de dois andares em diferentes planetas obtendo os valores na lista:

Python

```
gravity_on_planets = [0.378, 0.907, 1, 0.377, 2.36, 0.916, 0.889, 1.12]
bus_weight = 124054 # in Newtons, on Earth

print("On Earth, a double-decker bus weighs", bus_weight, "N")
print("On Mercury, a double-decker bus weighs", bus_weight *
gravity_on_planets[0], "N")
```

Output

```
On Earth, a double-decker bus weighs 124054 N
On Mercury, a double-decker bus weighs 46892.4 N
```

Usar `min()` e `max()` com listas

O Python tem funções internas para calcular o maior e o menor número em uma lista. A função `max()` retorna o maior número e a função `min()` retorna o menor número existente na lista. Portanto, `min(gravity_on_planets)` retorna o menor número da `gravity_on_planets` lista, que é 0,377 (Marte).

O seguinte código calcula os pesos mínimo e máximo do sistema solar usando essas funções:

Python

```
gravity_on_planets = [0.378, 0.907, 1, 0.377, 2.36, 0.916, 0.889, 1.12]
bus_weight = 12650 # in Newtons, on Earth

print("On Earth, a double-decker bus weighs", bus_weight, "N")
print("The lightest a bus would be in the solar system is", bus_weight *
min(gravity_on_planets), "N")
```

```
print("The heaviest a bus would be in the solar system is", bus_weight *  
max(gravity_on_planets), "N")
```

Output

```
On Earth, a double-decker bus weighs 124054 N  
The lightest a bus would be in the solar system is 46768.35 N  
The heaviest a bus would be in the solar system is 292767.44 N
```

Unidade seguinte: Manipular dados de lista

[Continuar >](#)