**The while Loop**

Najbardziej podstawowym typem pętli, jaką mamy do dyspozycji, jest pętla while. Ten typ pętli powtarza się w oparciu o warunek, który mu przekazujemy. Oto ogólna struktura pętli while:

while CONDITION:

pass

WARUNEK w tej instrukcji działa tak samo, jak w przypadku instrukcji if. Kiedy zademonstrowaliśmy instrukcję if, najpierw próbowaliśmy ją wykonać, przekazując warunek True jako warunek. Zobaczmy, kiedy spróbujemy tego samego warunku za pomocą pętli while:

>>> while True:

... print("looping")

...

looping

looping

looping

looping

Ta pętla będzie trwała wiecznie, stworzyliśmy nieskończoną pętlę. Aby zatrzymać pętlę, naciśnij Ctrl-C. Nieskończone pętle są jednym z potencjalnych problemów z pętlami while, jeśli nie użyjemy warunku, który możemy zmienić w pętli, a następnie będzie trwać wiecznie, jeśli początkowo będzie to prawda. Oto, jak normalnie podejdziemy za pomocą pętli while, w której zmodyfikujemy coś o stanie każdej iteracji:

>>> count = 1

>>> while count <= 4:

... print("looping")

... count += 1

...

looping

looping

looping

looping

>>>

Możemy użyć innych pętli lub warunków wewnątrz naszych pętli; musimy tylko pamiętać o wcięciu czterech kolejnych spacji dla każdego kontekstu. Jeśli w kontekście zagnieżdżonym, chcemy przejść do następnej iteracji lub całkowicie zatrzymać pętlę. Mamy również dostęp do słów kluczowych continue i break:

>>> count = 0

>>> while count < 10:

... if count % 2 == 0:

... count += 1

... continue

... print(f"We're counting odd numbers: {count}")

... count += 1

...

We're counting odd numbers: 1

We're counting odd numbers: 3

We're counting odd numbers: 5

We're counting odd numbers: 7

We're counting odd numbers: 9

>>>

W tym przykładzie pokazujemy również, jak „interpolować łańcuch” w Pythonie 3, poprzedzając literał łańcuchowy f, a następnie używając nawiasów klamrowych do zastąpienia w zmiennych lub wyrażeniach (w tym przypadku wartość licznika).

Oto przykład użycia instrukcji break:

>>> count = 1

>>> while count < 10:

... if count % 2 == 0:

... break

... print(f"We're counting odd numbers: {count}")

... count += 1

...

We're counting odd numbers: 1