While Loops

`While` Loops

`For` loops are an example of "definite iteration" meaning that the loop's body is run a predefined number of times. This differs from "indefinite iteration" which is when a loop repeats an unknown number of times and ends when some condition is met, which is what happens in a `while` loop. Here's an example of a `while` loop.

```python
card_deck = [4, 11, 8, 5, 13, 2, 8, 10]
hand = []

## adds the last element of the card_deck list to the hand list
## until the values in hand add up to 17 or more
while sum(hand)  < 17:
    hand.append(card_deck.pop())
```

This example features two new functions. `sum` returns the sum of the elements in a list, and `pop` is a list method that removes the last element from a list and returns it.

Components of a `While` Loop

1. The first line starts with the `while` keyword, indicating this is a `while` loop.

2. Following that is a condition to be checked. In this example, that's `sum(hand) <= 17`.

3. The `while` loop heading always ends with a colon `:`.

4. Indented after this heading is the body of the `while` loop. If the condition for the `while` loop is true, the code lines in the loop's body will be executed.

5. We then go back to the `while` heading line, and the condition is evaluated again. This process of checking the condition and then executing the loop repeats until the condition becomes false.

6. When the condition becomes false, we move on to the line following the body of the loop, which will be unindented.

The indented body of the loop should modify at least one variable in the test condition. If the value of the test condition never changes, the result is an infinite loop!

the value of the test condition never changes, the result is an infinite loop.