Loops

- Provide a way to run code multiple times
- DRY - Don't Repeat Yourself
- Two types of loops
  - For Loops
  - While Loops
Bart could have used a loop
For vs. While

For Loop
• Simpler

While Loop
• More complex
For vs. While

For Loop
- Simpler
- Finite set of iterations

While Loop
- More complex
- Unknown number of iterations
<table>
<thead>
<tr>
<th>For vs. While</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>For Loop</strong></td>
</tr>
<tr>
<td>- Simpler</td>
</tr>
<tr>
<td>- Finite set of iterations</td>
</tr>
<tr>
<td>- Automatic iteration</td>
</tr>
<tr>
<td><strong>While Loop</strong></td>
</tr>
<tr>
<td>- More complex</td>
</tr>
<tr>
<td>- Unknown number of iterations</td>
</tr>
<tr>
<td>- Manual iteration</td>
</tr>
</tbody>
</table>
For vs. While

For Loop
- Simpler
- Finite set of iterations
- Automatic iteration
- Some good uses:

While Loop
- More complex
- Unknown number of iterations
- Manual iteration
- Some good uses:
For vs. While

For Loop
- Simpler
- Finite set of iterations
- Automatic iteration
- Some good uses:
  - Use whenever possible, usually for iterating over a fixed size (e.g. counting to X)

While Loop
- More complex
- Unknown number of iterations
- Manual iteration
- Some good uses:
For vs. While

For Loop

• Simpler
• Finite set of iterations
• Automatic iteration
• Some good uses:
  • Use whenever possible, usually for iterating over a fixed size (e.g. counting to X)

While Loop

• More complex
• Unknown number of iterations
• Manual iteration
• Some good uses:
  • Run for an indeterminate amount of iteration
  • Reading files of undetermined size
For Loop

```python
for i in range(10):
    print("Count is:", i)
```

- `i` - short for index or iterator
- `range()` - returns an iterator that generates the numbers in the range on demand.
  - `range(10) == range(0,10)`
  - `range(0,10)` returns the values 0, 1, .., 9
  - `range(10, 0, -1)` returns the values 9, 8, .., 0
- See Examples
While Loop

i = 0  # initialize loop counter
while i < 10:
    print(i)
    i = i + 1  # increment loop counter

0) initialize loop counter  
   (done by the loop in 'For Loops')
1) check condition
2) increment counter  
   (done by the loop in 'For Loops')
Are you stuck in an infinite loop?

NO

YES
Infinite Loop

```python
while True:
    print("Hello")
```
Breaking out of a Loop

print("before while")
while True:
    break
print("after a while")
i = 0
while i < 100:
    i = i + 1
    if (i % 2):
        continue
    print("Odd: ", i)