Control Flow Revisited
CS106A, Stanford University
Housekeeping

- Assignment #1 due today
  - How long did you spend on it?
- Assignment #2 goes out today
  - It is due at 12:15pm on Friday, April 15th (one week)
- Invitation to participate in a study of learning introductory computing
  - Watch “Diary Study invitation” video on Canvas
Housekeeping II

• Extra office hours on Sundays from 3-5pm on Zoom
  – Details in post of Ed discussion Forum
• If you are comfortable doing so, please let us know your preferred name and your preferred pronouns
  – Also, if you can provide an audio recording of the pronunciation of your name
  – You can provide this information at the website: https://cs198.stanford.edu/profile
Review
A block (also called a suite) in Python is defined by a set of lines of code indented by (at least) the same amount.

A block ends when the next line is indented less than the line before it.

Example:

```python
main():
    move_to_wall()
    for i in range(3):
        turn_left()
    move()

def move_to_wall():
    while front_is_clear():
        move()
```

Statement Block

CS106A, Stanford University
A `block` (also called a `suite`) in Python is defined by a set of lines of code indented by (at least) the same amount.

- A block ends when the next line is indented less than the line before it.

Example:

```python
main():
    move_to_wall()
    for i in range(3):
        turn_left()
        move()

def move_to_wall():
    while front_is_clear():
        move()
```
num_students = 700
Your computer has space for millions of suitcases!
# Create a variable, of type int
called age with the value 31.

age = 31

# Use the value in age (output it)

print("age is: " + str(age))

# Modify age to be one greater.

age = age + 1
Recall, Arithmetic Operators

+ Addition
- Subtraction
* Multiplication
/ Division
Today’s Goal

1. Learn about Boolean variables
2. Use For / While / If statements in Python
Today’s Route

Core Python

You are here

Booleans

if / if-else

while loops

for loops

The River of Control Flow
George Boole

English Mathematician teaching in Ireland 1815 – 1864
Boole died of being too cool (literally)!
Boolean Variables

• Boolean variables only have values **True** or **False**
  
  \[
  x = \text{True} \\
  y = \text{False}
  \]

• Represent **logical** values

• Type is called **bool** in python
Boolean Variables

- Can also set value of Boolean variable by comparison
  - This is called a Boolean expression
    
    \[ p = 5.0 < 4.0 \]
    \[ q = 2 > 1 \]
## Comparison Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
<th>Example</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>==</code></td>
<td>equals</td>
<td><code>(1 + 1) == 2</code></td>
<td>True</td>
</tr>
<tr>
<td><code>!=</code></td>
<td>not equals</td>
<td><code>3.2 != 2.5</code></td>
<td>True</td>
</tr>
<tr>
<td><code>&lt;</code></td>
<td>less than</td>
<td><code>10 &lt; 5</code></td>
<td>False</td>
</tr>
<tr>
<td><code>&gt;</code></td>
<td>greater than</td>
<td><code>10 &gt; 5</code></td>
<td>True</td>
</tr>
<tr>
<td><code>&lt;=</code></td>
<td>less than or equal to</td>
<td><code>126 &lt;= 100</code></td>
<td>False</td>
</tr>
<tr>
<td><code>&gt;=</code></td>
<td>greater than or equal to</td>
<td><code>5.0 &gt;= 5.0</code></td>
<td>True</td>
</tr>
</tbody>
</table>

All have comparisons have equal precedence.
In order of precedence (assuming \( p \) and \( q \) are Boolean values):

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>not</td>
<td>if ( p ) is True then not ( p ) is False, and vice versa</td>
</tr>
<tr>
<td>and</td>
<td>( p ) and ( q ) is only True if ( p ) and ( q ) are both True.</td>
</tr>
<tr>
<td></td>
<td>So, it is False when either ( p ) or ( q ) are False.</td>
</tr>
<tr>
<td>or</td>
<td>( p ) or ( q ) is True if either ( p ) or ( q ) (or both) are True.</td>
</tr>
<tr>
<td></td>
<td>So, it is only False when both ( p ) and ( q ) are False.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operator</th>
<th>Example</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>not</td>
<td>not (2 &gt; 3)</td>
<td>True</td>
</tr>
<tr>
<td>and</td>
<td>(2 == 3) and (-1 &lt; 5)</td>
<td>False</td>
</tr>
<tr>
<td>or</td>
<td>(4 == 5) or (7 &gt;= 3)</td>
<td>True</td>
</tr>
</tbody>
</table>
Logical Operators

• What if we want to determine that a variable $x$ does not have a value of either 1 or 2?
• Might write something like how it sounds in English:
  $$p = (x \neq 1) \text{ or } (x \neq 2)$$
  – That’s a bug! $p$ will always be True
• Really, want to say:
  $$p = (x \neq 1) \text{ and } (x \neq 2)$$
• Bonus: you can chain tests for a range, like in algebra!
  $$p = (2 \leq x \leq 10)$$
  – Same thing as:
  $$p = (2 \leq x) \text{ and } (x \leq 10)$$
Short Circuit Evaluation

• Python stops evaluating a Boolean expression as soon as it knows the answer.

• Consider:

\[ p = (5 > 3) \text{ or } (4 \leq 2) \]

• The test \((4 \leq 2)\) is not performed!

• Example of useful case:

\[ p = (x \neq 0) \text{ and } ((y \% x) == 0) \]

  – Avoid division by 0 error, since \((y \% x) == 0\) is not performed when \(x\) is 0

• To compute remainder (\%), Python needs to do division.
Please...
NO FOOD OR DRINKS
Please...

NO FOOD OR DRINKS

is_allowed = not food or drinks
Please...

NO FOOD OR DRINKS

is_allowed = (not food) or drinks
Today’s Route

You are here

Core Python

The River of Control Flow

Booleans → if / if-else

for loops → while loops
if statements, revisited

Any Boolean expression or variable

```python
if condition:
    statements  # note indenting (block)

x = int(input("Enter a number: "))
if x < 10:
    print(x, "is less than 10")
print("Beat Cal!")
```
if statements, revisited

```python
if condition:
    statements
    (block)

# note indenting

x = int(input("Enter a number: "))
if x < 10:
    print(x, "is less than 10")
print("Beat Cal!")
```

Enter a number:
if statements, revisited

```
# note indenting

if condition:
    statements
(block)
```

```python
x = int(input("Enter a number: "))
if x < 10:
    print(x, "is less than 10")
print("Beat Cal!")
```

Enter a number: 8
if statements, revisited

if condition:
    statements
    (block)

    # note indenting

x = int(input("Enter a number: "))
if x < 10:
    print(x, "is less than 10")
print("Beat Cal!")

Enter a number: 8
if statements, revisited

Any Boolean expression or variable

```python
if condition:
    statements
    # note indenting
```

```python
x = int(input("Enter a number: "))
if x < 10:
    print(x, "is less than 10")
print("Beat Cal!")
```

Enter a number: 8
8 is less than 10
if statements, revisited

```python
if condition:
    statements
    # note indenting

x = int(input("Enter a number: "))
if x < 10:
    print(x, "is less than 10")
print("Beat Cal!")
```

Enter a number: 8
8 is less than 10
Beat Cal!
if statements, revisited

```
if condition:
    statements # note indenting
(block)
```

```
x = int(input("Enter a number: "))
if x < 10:
    print(x, "is less than 10")
print("Beat Cal!")
```

Enter a number:
if statements, revisited

if **condition**:  # note indenting
    **statements**
    (block)

```python
x = int(input("Enter a number: "))
if x < 10:
    print(x, "is less than 10")
print("Beat Cal!")
```

Enter a number: 15
if statements, revisited

if \textbf{condition}:
  \textbf{statements} # note indenting
  (block)

\begin{verbatim}
x = int(input("Enter a number: "))
if x < 10:
  print(x, "is less than 10")
print("Beat Cal!")
\end{verbatim}

Enter a number: 15
If statements, revisited

```python
if condition:
    statements  # note indenting (block)

x = int(input("Enter a number: "))
if x < 10:
    print(x, "is less than 10")
print("Beat Cal!")
```

Enter a number: 15
Beat Cal!
Nested if statements

```python
x = int(input("Enter a number: "))
if x > 0:
    print(x, "is positive")
    if x > 5:
        print(x, "is greater than 5")
        print("I like to stay positive")
    print("Thank you")
```

CS106A, Stanford University
x = int(input("Enter a number: "))
if x > 0:
    print(x, "is positive")
    if x > 5:
        print(x, "is greater than 5")
    print("I like to stay positive")
print("Thank you")

Enter a number:
```python
x = int(input("Enter a number: 
if x > 0:
    print(x, "is positive")
if x > 5:
    print(x, "is greater than 5")
print("I like to stay positive")
print("Thank you")
```

Enter a number: 3
x = int(input("Enter a number: "))
if x > 0:
    print(x, "is positive")
    if x > 5:
        print(x, "is greater than 5")
        print("I like to stay positive")
    print("Thank you")

Enter a number: 3
Nested `if` statements

```python
x = int(input("Enter a number: "))
if x > 0:
    print(x, "is positive")
    if x > 5:
        print(x, "is greater than 5")
        print("I like to stay positive")
print("Thank you")
```

Enter a number: 3
3 is positive
Nested if statements

```python
x = int(input("Enter a number: "))
if x > 0:
    print(x, "is positive")
    if x > 5:
        print(x, "is greater than 5")
        print("I like to stay positive")
    print("Thank you")
```

Enter a number: 3
3 is positive
Nested `if` statements

```python
x = int(input("Enter a number: "))
if x > 0:
    print(x, "is positive")
    if x > 5:
        print(x, "is greater than 5")
        print("I like to stay positive")
print("Thank you")
```

Enter a number: 3
3 is positive
I like to stay positive

Enter a number: 3
3 is positive
I like to stay positive
Nested if statements

```python
x = int(input("Enter a number: "))
if x > 0:
    print(x, "is positive")
    if x > 5:
        print(x, "is greater than 5")
        print("I like to stay positive")
    print("Thank you")
```

Enter a number: 3
3 is positive
I like to stay positive
Thank you
Nested if statements

```python
x = int(input("Enter a number: "))
if x > 0:
    print(x, "is positive")
    if x > 5:
        print(x, "is greater than 5")
        print("I like to stay positive")
print("Thank you")
```

Enter a number:
Nested if statements

```python
x = int(input("Enter a number: "))
if x > 0:
    print(x, "is positive")
    if x > 5:
        print(x, "is greater than 5")
    print("I like to stay positive")
print("Thank you")
```

Enter a number: 12
x = int(input("Enter a number: "))
if x > 0:
    print(x, "is positive")
    if x > 5:
        print(x, "is greater than 5")
    print("I like to stay positive")
print("Thank you")

Enter a number: 12
Nested if statements

```python
x = int(input("Enter a number: "))
if x > 0:
    print(x, "is positive")
    if x > 5:
        print(x, "is greater than 5")
        print("I like to stay positive")
    print("Thank you")
```

Enter a number: 12
12 is positive
Nested if statements

```python
x = int(input("Enter a number: "))
if x > 0:
    print(x, "is positive")
    if x > 5:
        print(x, "is greater than 5")
        print("I like to stay positive")
    print("Thank you")
```

Enter a number: 12
12 is positive
x = int(input("Enter a number: "))
if x > 0:
    print(x, "is positive")
    if x > 5:
        print(x, "is greater than 5")
    print("I like to stay positive")
print("Thank you")

Enter a number: 12
12 is positive
12 is greater than 5
Nested `if` statements

```python
x = int(input("Enter a number: 
if x > 0:
    print(x, "is positive")
    if x > 5:
        print(x, "is greater than 5")
        print("I like to stay positive")
print("Thank you")
```

Enter a number: 12
12 is positive
12 is greater than 5
I like to stay positive
 Nested if statements

```python
x = int(input("Enter a number: "))
if x > 0:
    print(x, "is positive")
    if x > 5:
        print(x, "is greater than 5")
        print("I like to stay positive")
    print("Thank you")
```

Enter a number: 12
12 is positive
12 is greater than 5
I like to stay positive
Thank you
Nested `if` statements

```python
x = int(input("Enter a number: "))
if x > 0:
    print(x, "is positive")
    if x > 5:
        print(x, "is greater than 5")
        print("I like to stay positive")
    print("Thank you")
```

Enter a number:
Nested if statements

```python
x = int(input("Enter a number: "))
if x > 0:
    print(x, "is positive")
    if x > 5:
        print(x, "is greater than 5")
    print("I like to stay positive")
print("Thank you")
```

Enter a number: -2
Nested `if` statements

```python
x = int(input("Enter a number: "))
if x > 0:
    print(x, "is positive")
    if x > 5:
        print(x, "is greater than 5")
        print("I like to stay positive")
    print("Thank you")
```

Enter a number: -2
Nested if statements

```python
x = int(input("Enter a number: "))
if x > 0:
    print(x, "is positive")
    if x > 5:
        print(x, "is greater than 5")
        print("I like to stay positive")
    print("I like to stay positive")
print("Thank you")
```

Enter a number: -2
Thank you
# if-else statement

```python
if condition:
    statements (block)  # condition True
else:
    statements (block)  # condition False

num = int(input("Enter a number: "))
if (num % 2) == 0:
    print(num, "is even")
else:
    print(num, "is odd")
print("and so are you!")
```

Any Boolean expression or variable
if condition:
    statements (block) # condition True
else:
    statements (block) # condition False

num = int(input("Enter a number: "))
if (num % 2) == 0:
    print(num, "is even")
else:
    print(num, "is odd")
print("and so are you!")

Enter a number:
if condition:
    statements (block)  # condition True
else:
    statements (block)  # condition False

num = int(input("Enter a number: "))
if (num % 2) == 0:
    print(num, "is even")
else:
    print(num, "is odd")
print("and so are you!")

Enter a number: 4
if condition:
    statements (block)  # condition True
else:
    statements (block)  # condition False

num = int(input("Enter a number: "))
if (num % 2) == 0:
    print(num, "is even")
else:
    print(num, "is odd")
print("and so are you!")

Enter a number: 4
if condition:
    statements (block)  # condition True
else:
    statements (block)  # condition False

num = int(input("Enter a number: "))
if (num % 2) == 0:
    print(num, "is even")
else:
    print(num, "is odd")
print("and so are you!")

Enter a number: 4
4 is even
```python
if condition:
    statements (block)  # condition True
else:
    statements (block)  # condition False

num = int(input("Enter a number: "))
if (num % 2) == 0:
    print(num, "is even")
else:
    print(num, "is odd")
print("and so are you!")
```
if `condition`
  `statements (block)`  # condition True
else:
  `statements (block)`  # condition False

num = int(input("Enter a number: "))
if (num % 2) == 0:
    print(num, "is even")
else:
    print(num, "is odd")
print("and so are you!")
if condition:
    statements (block)  # condition True
else:
    statements (block)  # condition False

num = int(input("Enter a number: "))
if (num % 2) == 0:
    print(num, "is even")
else:
    print(num, "is odd")
print("and so are you!")

Enter a number: 11
if condition:
    statements (block) # condition True
else:
    statements (block) # condition False

num = int(input("Enter a number: "))
if (num % 2) == 0:
    print(num, "is even")
else:
    print(num, "is odd")
print("and so are you!")

Enter a number: 11
if condition:
  statements (block)  # condition True
else:
  statements (block)  # condition False

num = int(input("Enter a number: "))
if (num % 2) == 0:
  print(num, "is even")
else:
  print(num, "is odd")
print("and so are you!")

Enter a number: 11
if-else statement

```python
num = int(input("Enter a number: "))
if (num % 2) == 0:
    print(num, "is even")
else:
    print(num, "is odd")
print("and so are you!")
```

Enter a number: 11
11 is odd
if-else statement

```python
if condition:
    statements (block)  # condition True
else:
    statements (block)  # condition False
```

```python
num = int(input("Enter a number: "))
if (num % 2) == 0:
    print(num, "is even")
else:
    print(num, "is odd")
print("and so are you!")
```

Example:
Enter a number: 11
11 is odd
and so are you!
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!")
else:
    if num > 0:
        print(num, "is positive")
    else:
        print(num, "is negative")
Nested if-else statements

```python
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!")
else:
    if num > 0:
        print(num, "is positive")
    else:
        print(num, "is negative")
```

Enter a number:
Nested *if-else* statements

```python
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!")
else:
    if num > 0:
        print(num, "is positive")
    else:
        print(num, "is negative")
```

Enter a number: 6
num = int(input("Enter a number: 
if num == 0: 
    print("Zero is my hero!\n")
else: 
    if num > 0: 
        print(num, "is positive")
    else: 
        print(num, "is negative")

Enter a number: 6
Nested if–else statements

```python
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!")
else:
    if num > 0:
        print(num, "is positive")
    else:
        print(num, "is negative")
```

Enter a number: 6
Nested if-else statements

```python
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!")
else:
    if num > 0:
        print(num, "is positive")
    else:
        print(num, "is negative")
```

Enter a number: 6
Nested \texttt{if-else} statements

\begin{verbatim}
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!")
else:
    if num > 0:
        print(num, "is positive")
    else:
        print(num, "is negative")
\end{verbatim}

Enter a number: 6
6 is positive
Nested \texttt{if-else} statements

\begin{verbatim}
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!")
else:
    if num > 0:
        print(num, "is positive")
    else:
        print(num, "is negative")
\end{verbatim}

Enter a number: 6
6 is positive
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!")
else:
    if num > 0:
        print(num, "is positive")
    else:
        print(num, "is negative")
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!")
else:
    if num > 0:
        print(num, "is positive")
    else:
        print(num, "is negative")
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!")
else:
    if num > 0:
        print(num, "is positive")
    else:
        print(num, "is negative")
num = int(input("Enter a number: 
if num == 0:
    print("Zero is my hero!"")
elif num > 0:
    print(num, "is positive")
else:
    print(num, "is negative")
Bring me the `elif`

```python
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!")
elif num > 0:
    print(num, "is positive")
else:
    print(num, "is negative")
```
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!")
elif num > 0:
    print(num, "is positive")
else:
    print(num, "is negative")

Enter a number: -2
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!"")
elif num > 0:
    print(num, "is positive")
else:
    print(num, "is negative")

Enter a number: -2
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!")
elif num > 0:
    print(num, "is positive")
else:
    print(num, "is negative")

Enter a number: -2
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!"")
elif num > 0:
    print(num, "is positive")
else:
    print(num, "is negative")

Enter a number: -2
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!")
elif num > 0:
    print(num, "is positive")
else:
    print(num, "is negative")
Enter a number: -2
-2 is negative
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!")
elif num > 0:
    print(num, "is positive")
else:
    print(num, "is negative")

Enter a number: -2
-2 is negative
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!"")
elif num > 0:
    print(num, "is positive")
else:
    print(num, "is negative")

Enter a number:
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!")
elif num > 0:
    print(num, "is positive")
else:
    print(num, "is negative")

Enter a number: 0
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!")
elif num > 0:
    print(num, "is positive")
else:
    print(num, "is negative")

Enter a number: 0
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!")
elif num > 0:
    print(num, "is positive")
else:
    print(num, "is negative")

Enter a number: 0
Zero is my hero!
num = int(input("Enter a number: "))
if num == 0:
    print("Zero is my hero!"")
elif num > 0:
    print(num, "is positive")
else:
    print(num, "is negative")

Done!

Enter a number: 0
Zero is my hero!
It works for grades too!

```python
grade = int(input("Enter a score: "))
if grade >= 90:
    print("A")
elif grade >= 80:
    print("B")
elif grade >= 70:
    print("C")
else:
    print("Keep trying!")
```
It works for grades too!

```python
grade = int(input("Enter a score: "))
if grade >= 90:
    print("A")
elif grade >= 80:
    print("B")
elif grade >= 70:
    print("C")
else:
    print("Keep trying!")
```

Enter a score:
It works for grades too!

```python
grade = int(input("Enter a score: "))
if grade >= 90:
    print("A")
elif grade >= 80:
    print("B")
elif grade >= 70:
    print("C")
else:
    print("Keep trying!")
```

Enter a score: 82
It works for grades too!

```python
grade = int(input("Enter a score: "))
if grade >= 90:
    print("A")
elif grade >= 80:
    print("B")
elif grade >= 70:
    print("C")
else:
    print("Keep trying!")

Enter a score: 82
```
It works for grades too!

```python
grade = int(input("Enter a score: "))
if grade >= 90:
    print("A")
elif grade >= 80:
    print("B")
elif grade >= 70:
    print("C")
else:
    print("Keep trying!")
```

Enter a score: 82
It works for grades too!

```python
grade = int(input("Enter a score: "))
if grade >= 90:
    print("A")
elif grade >= 80:
    print("B")
elif grade >= 70:
    print("C")
else:
    print("Keep trying!")
```

Enter a score: 82
B
grade = int(input("Enter a score: "))
if grade >= 90:
    print("A")
elif grade >= 80:
    print("B")
elif grade >= 70:
    print("C")
else:
    print("Keep trying!")

Enter a score: 82
B
Amazing!
Today's Route

You are here

Core Python

The River of Control Flow

Booleans

if / if-else

for loops

while loops
**while loop**

```
while condition:
    statements
    # note indenting
```

```
x = 10
while x > 1:
    x /= 2
    print(x)
```
while loop

```
x = 10
while x > 1:
    x /= 2
    print(x)
```

Any Boolean expression or variable
while loop

while **condition**:  # note indenting
    statements
    (block)

x = 10
while x > 1:
    x /= 2
    print(x)
while loop

while \textbf{condition}: \\
\textit{statements} \\
\textit{(block)}

# note indenting

x = 10
while x > 1:
    x /= 2
    print(x)
while loop

while **condition**:  # note indenting
    **statements**
    (block)

x = 10
while x > 1:
    x /= 2
    print(x)

5.0
while loop

```python
while condition:  # note indenting
    statements
    (block)
```

```python
x = 10
while x > 1:
    x /= 2
    print(x)
```

5.0
while loop

while condition:
    statements
# note indenting
(block)

x = 10
while x > 1:
    x /= 2
    print(x)

5.0

Any Boolean expression or variable
while loop

```
while condition:
    statements
    # note indenting
    (block)
```

```python
x = 10
while x > 1:
    x /= 2
    print(x)
```

```
5.0
2.5
```
while loop

```python
while condition:
    statements
    # note indenting
```

```
x = 10
while x > 1:
    x /= 2
    print(x)
```

```
5.0
2.5
```
while loop

while condition:
    statements
    # note indenting

(x = 10)
while x > 1:
    x /= 2
    print(x)

5.0
2.5
while loop

while condition:
    statements
    # note indenting
(block)

x = 10
while x > 1:
    x /= 2
    print(x)

5.0
2.5
1.25
while loop

while condition:
    statements
    (block)

# note indenting

x = 10

while x > 1:
    x /= 2
    print(x)

5.0
2.5
1.25
```python
while condition:
    statements
    # note indenting

x = 10
while x > 1:
    x /= 2
    print(x)
```

5.0
2.5
1.25
while condition:
    statements
    # note indenting

x = 10
while x > 1:
    x /= 2
    print(x)

5.0
2.5
1.25
0.625
while loop

```python
while condition:
    statements
    # note indenting
(block)
```

```python
x = 10
while x > 1:
    x /= 2
    print(x)
```

```
5.0
2.5
1.25
0.625
```
**while loop**

while **condition**:
    **statements**
    (block)

# note indenting

```python
x = 10
while x > 1:
    x /= 2
    print(x)
```

Done!

5.0
2.5
1.25
0.625
Conditions in Python

You can use if, if-else, and while statements in Python.

They are the same as in Karel, except that the condition can be any expression that evaluates to True or False (Boolean).
SENTINEL = -1

def main():
    total = 0
    value = int(input("Value? "))
    while value != SENTINEL:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))
SENTINEL = -1

def main():
    total = 0
    value = int(input("Value? "))
    while value != SENTINEL:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))
SENTINEL = -1

def main():
    total = 0
    value = int(input("Value? "))
    while value != SENTINEL:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))

Value?
def main():
    total = 0
    value = int(input("Value? "))
    while value != SENTINEL:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))

Value? 4
A Loop-and-a-Half?

```
SENTINEL = -1

def main():
    total = 0
    value = int(input("Value? "))
    while value != SENTINEL:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))

Value? 4
```
SENTINEL = -1

def main():
    total = 0
    value = int(input("Value? "))
    while value != SENTINEL:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))

Value? 4
SENTINEL = -1

def main():
    total = 0
    value = int(input("Value? "))
    while value != SENTINEL:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))

Value? 4
Value?
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    value = int(input("Value? "))
    while value != SENTINEL:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))

Value? 4
Value? 6
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    value = int(input("Value? "))
    while value != SENTINEL:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))

Value? 4
Value? 6
A Loop-and-a-Half?

SENTINEL = -1
def main():
    total = 0
    value = int(input("Value? "))
    while value != SENTINEL:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))

Value? 4
Value? 6
SENTINEL = -1

def main():
    total = 0
    value = int(input("Value? "))
    while value != SENTINEL:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))

Value? 4
Value? 6
Value?
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    value = int(input("Value? "))
    while value != SENTINEL:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))

Value? 4
Value? 6
Value? 11
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    value = int(input("Value? "))
    while value != SENTINEL:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))

Value? 4
Value? 6
Value? 11
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    value = int(input("Value? "))
    while value != SENTINEL:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))

Value? 4
Value? 6
Value? 11
SENTINEL = -1

def main():
    total = 0
    value = int(input("Value? "))
    while value != SENTINEL:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))

Value? 4
Value? 6
Value? 11
Value?
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    value = int(input("Value? "))
    while value != SENTINEL:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))

Value? 4
Value? 6
Value? 11
Value? -1
SENTINEL = -1

def main():
    total = 0
    value = int(input("Value? "))
    while value != SENTINEL:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))

Value? 4
Value? 6
Value? 11
Value? -1
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    value = int(input("Value? "))
    while value != SENTINEL:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))

Value? 4
Value? 6
Value? 11
Value? -1
Total = 21
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    value = int(input("Value? "))
    while value != SENTINEL:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))

10 panels, 11 fenceposts

CS106A, Stanford University
SENTINEL = -1

def main():
    total = 0
    value = int(input("Value? "))
    while value != SENTINEL:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0

    while value != SENTINEL:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))

UnboundLocalError: local variable 'value' referenced before assignment
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0

    while True:
        total += value
        value = int(input("Value? "))
    print("Total = " + str(total))
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    while True:
        total += value
        value = int(input("Value? "))
        print("Total = " + str(total))
SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? "))
        total += value
        print("Total = " + str(total))
def main():
    total = 0
    while True:
        value = int(input("Value? "))
        total += value
    print("Total = " + str(total))
SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? "))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? "))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))

Immediately break out of current loop
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? "))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))

    value
    total
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? "))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))
def main():
    total = 0
    while True:
        value = int(input("Value? "))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))

Value?
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? "))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))

Value? 4
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? "))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))

Value? 4
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? "))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))

Value? 4
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? "))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))

Value? 4
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? "))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))

Value? 4
Value? 6
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? " ))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))

Value? 4
Value? 6
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? "))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))

Value? 4
Value? 6
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? "))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))

Value? 4
Value? 6
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? "))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))

Value? 4
Value? 6
Value? 11
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? "))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))

Value? 4
Value? 6
Value? 11
SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? "))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))

Value? 4
Value? 6
Value? 11
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? "))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))

Value? 4
Value? 6
Value? 11
SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? "))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))

Value? 4
Value? 6
Value? 11
Value? -1
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? "))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))

Value? 4
Value? 6
Value? 11
Value? -1
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? "))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))

Value? 4
Value? 6
Value? 11
Value? -1
SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? "))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))

Value? 4
Value? 6
Value? 11
Value? -1
Total = 21
A Loop-and-a-Half?

SENTINEL = -1

def main():
    total = 0
    while True:
        value = int(input("Value? "))
        if value == SENTINEL:
            break
        total += value
    print("Total = " + str(total))

Value? 4
Value? 6
Value? 11
Value? -1
Total = 21
for loop

This is called an index variable. Can have any variable name.

```python
for i in range(count):
    statements  # note indenting (block)
```

```python
for i in range(3):
    print(i)
```
for i in range(3):
    print(i)

for i in range(count):
    statements
    # note indenting (block)

Index starts counting from 0
Counts up to (count - 1)

This is called an index variable.
Can have any variable name.
This is called an index variable. Can have any variable name.

```python
for i in range(count):
    statements # note indenting (block)
```

Index starts counting from 0
Counts up to (count – 1)

```
for i in range(3):
    print(i)
```

0
for loop

This is called an index variable. Can have any variable name.

```python
for i in range(count):
    statements  # note indenting (block)
```

Index starts counting from 0
Counts up to (count – 1)

```
for i in range(3):
    print(i)
```

```
i
0
```
for loop

This is called an index variable. Can have any variable name.

for i in range(count):
    statements
    # note indenting (block)

for i in range(3):
    print(i)

Index starts counting from 0
Counts up to (count – 1)

<table>
<thead>
<tr>
<th>i</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>
This is called an **index** variable. Can have any variable name.

```python
for i in range(count):
    statements # note indenting (block)
```

Index starts counting from 0 Counts up to (count – 1)

```
for i in range(3):
    print(i)
```

```
0
1
```
for i in range(`count`):
    statements  # note indenting (block)

for i in range(3):
    print(i)

This is called an **index** variable. Can have any variable name.

Index starts counting from 0
Counts up to (count – 1)
for loop

This is called an `index` variable. Can have any variable name.

```python
for i in range(count):
    statements
# note indenting (block)
```

For `i` in `range(3)`:
- `i` starts counting from 0
- Counts up to `(count - 1)`

<table>
<thead>
<tr>
<th>i</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>
for loop

This is called an index variable. Can have any variable name.

```python
for i in range(count):
    statements # note indenting (block)
```

for i in range(3):
    print(i)

Done!

Index starts counting from 0
Counts up to (count – 1)

```
<table>
<thead>
<tr>
<th>i</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>
```
Write a program to print “Python rocks my socks” 100 times
def main():
    for i in range(100):
        print("Python rocks my socks")
for loop redux

- **i** starts at 0, counts up to 99 (inclusive)

```python
def main():
    for i in range(100):
        print("Python rocks my socks")
```

Can think of it like this:

```python
def main():
    i = 0
    while i < 100:
        print("Python rocks my socks")
        i += 1
```
for loops – the advanced course!

```python
for i in range(start, end, step):
    statements (block)
```

Starts counting at `start`
Stops counting before `end`
Each iteration, add `step` to counter

```python
for i in range(1,7,2):
    print(i)
```
for loops – the advanced course!

for i in range(start, end, step):
    statements (block)

- Starts counting at start
- Stops counting before end
- Each iteration, add step to counter

```python
for i in range(1, 7, 2):
    print(i)
```

Output:
```
1
3
5
```
for loops – the advanced course!

for i in range(start, end, step):
    statements (block)

    Starts counting at start
    Stops counting before end
    Each iteration, add step to counter

for i in range(1, 7, 2):
    print(i)
for loops – the advanced course!

for i in range(start, end, step):
    statements

    (block)

    Starts counting at start
    Stops counting before end
    Each iteration, add step to counter

for i in range(1, 7, 2):
    print(i)

    i 3

    3

    1
for loops – the advanced course!

```python
for i in range(start, end, step):
    statements
```

- Starts counting at `start`
- Stops counting before `end`
- Each iteration, add `step` to counter

```python
for i in range(1, 7, 2):
    print(i)
```

Output:
```
1
3
```

Example:
```
for i in range(1, 7, 2):
    print(i)
```

Output:
```
1
3
```
for loops – the advanced course!

for i in range(start, end, step):
    statements
    (block)

    Starts counting at start
    Stops counting before end
    Each iteration, add step to counter

for i in range(1, 7, 2):
    print(i)

i
5

1
3
for loops – the advanced course!

for i in range(start, end, step):
    statements
    (block)

    Starts counting at start
    Stops counting before end
    Each iteration, add step to counter

for i in range(1, 7, 2):
    print(i)

    1
    3
    5

for loops – the advanced course!

for i in range(start, end, step):
    statements
    (block)

    Starts counting at start
    Stops counting before end
    Each iteration, add step to counter

for i in range(1, 7, 2):
    print(i)

    Done!

1
3
5

5
Putting it all together:
guessnumber.py
1. Be able to use For / While / If in Python
Have a good weekend!