Control Flow Revisited

Chris Piech

CS106A, Stanford University
Review
num_students = 700
My computer has space for about 2 billion boxes
# Create a variable, of type int
# called age with the value 30.
age = 30

# Modify age to be one greater.
age = age + 1

# Use the value in age (output it)
print("age is: "+age)
Binary Operators

+  Addition  
−  Subtraction  
%  Remainder  
/  Division  
*  Multiplication  
//  Int Division
Today’s Goal

1. Be able to use For / While / If in Java
Today’s Route

For Loops
Booleans
Simple Java

The River of Java

Review
Conditions
You are here

Piech, CS106A, Stanford University
Today’s Route

Simple Java

For Loops

Booleans

Conditions

Review

The River of Java

You are here
While Loop in Karel

```python
while front_is_clear() :
    body

if beepers_present() :
    body
```
While Loop Redux

while condition:
    body

if condition:
    body

The condition should be a “boolean” which is either True or False
Cool Example: Carbon Dating

C14 = 1.2 dpm

C14 = 13.6 dpm

\[ \text{age} = K \cdot \log \left( \frac{\% \text{ C14}}{100} \right) \]
def main():
    calculate_age_single_sample()

def calculate_age_single_sample():
    # use the half life formula to calculate the age
    # https://en.wikipedia.org/wiki/Radiocarbon_dating
    pct_left = float(input("% of natural c14 in your sample:"))
    age = math.log(pct_left / 100) * HALF_LIFE_CONSTANT
    print("Your sample is " + str(age) + " years old.")

* It calculates the age of a C14 sample
def main():
    while True:
        calculate_age_single_sample()

def calculate_age_single_sample():
    # use the half life formula to calculate the age
    # https://en.wikipedia.org/wiki/Radiocarbon_dating
    pct_left = float(input("% of natural c14 in your sample:"))
    age = math.log(pct_left / 100) * HALF_LIFE_CONSTANT
    print("Your sample is " + str(age) + " years old.")

* It calculates the age of a C14 sample

Piech, CS106A, Stanford University
Today’s Route

Review

Conditions

Booleans

For Loops

Simple Java

You are here

The River of Java
Today’s Route

The River of Java

Review

Conditions

Booleans

For Loops

Simple Java

You are here
Booleans

$1 < 2$
Booleans

1 < 2

True
Comparison Operators

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

* All have equal precedence
### 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>does not equal</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 equal precedence*
Comparison Operators

```python
if 1 < 2:
    print("1 is less than 2")

num = int(input("Enter a number: "))
if num == 0:
    print("That number is 0")
else:
    print("That number is not 0.")
```
num = int(input("Enter a number: "))

if num == 0:
    print("Your number is 0 ")
else:
    if num > 0:
        print("Your number is positive")
    else:
        print("Your number is negative")
num = int(input("Enter a number: 

if num == 0:
    print("Your number is 0 ")
elif num > 0:
    print("Your number is positive")
else:
    print("Your number is negative")
num = int(input("Enter a number: "))
if num == 0:
    print("Your number is 0 ")
elif num > 0:
    print("Your number is positive")
else:
    print("Your number is negative")
num = int(input("Enter a number: "))
if num == 0:
    print("Your number is 0 ")
elif num > 0:
    print("Your number is positive")
else:
    print("Your number is negative")
If Else Revisited

```python
num = int(input("Enter a number: "))
if num == 0:
    print("Your number is 0 ")
elif num > 0:
    print("Your number is positive")
else:
    print("Your number is negative")
```

Enter a number: 5
num = int(input("Enter a number: "))

if num == 0:
    print("Your number is 0 ")
elif num > 0:
    print("Your number is positive")
else:
    print("Your number is negative")

Enter a number: 5
num = int(input("Enter a number: "))

if num == 0:
    print("Your number is 0 ")
elif num > 0:
    print("Your number is positive")
else:
    print("Your number is negative")

Enter a number: 5
num = \texttt{int}(\texttt{input}("Enter a number: "))

\textbf{if} num == 0:
    \texttt{print}("Your number is 0 ")

\textbf{elif} num > 0:
    \texttt{print}("Your number is positive")

\textbf{else}:
    \texttt{print}("Your number is negative")

\textbf{Enter a number: 5}
num = \textbf{int}(\text{input("Enter a number: ")})

\textbf{if} num == 0:
    \text{print("Your number is 0 ")}

\textbf{elif} num > 0:
    \text{print("Your number is positive")}

\textbf{else}:
    \text{print("Your number is negative")}

Enter a number: 5
Your number is positive
If Else Revisited

```python
num = int(input("Enter a number: "))
if num == 0:
    print("Your number is 0 ")
elif num > 0:
    print("Your number is positive")
else:
    print("Your number is negative")
```

Enter a number: 5
Your number is positive
Use **while** and **if** statements in Python.

They are the same as in Karel, except that the *test* can be any expression that evaluates to **True** or **False**.
Amazing
Guess My Number

I am thinking of a number between 0 and 99...
Enter a guess: 50
Your guess is too high

Enter a new number: 25
Your guess is too low

Enter a new number: 40
Your guess is too low

Enter a new number: 45
Your guess is too low

Enter a new number: 48
Congrats! The number was: 48
**Guess My Number**

```python
secret_number = random.randint(1, 99)
print("I am thinking of a number between 1 and 99...")
guess = int(input("Enter a guess: "))
# True if guess is not equal to secret number
while guess != secret_number:
    # True if guess is less than secret number
    if guess < secret_number:
        print("Your guess is too low")
    else:
        print("Your guess is too high")

print("") # an empty line
guess = int(input("Enter a new guess: "))

print("Congrats! The number was: " + str(secret_number))
```

`92` int

**secret_number**
secret_number = random.randint(1, 99)
print("I am thinking of a number between 1 and 99...")
guess = int(input("Enter a guess: "))
# True if guess is not equal to secret number
while guess != secret_number:
    # True if guess is less than secret number
    if guess < secret_number:
        print("Your guess is too low")
    else:
        print("Your guess is too high")

print("") # an empty line
guess = int(input("Enter a new guess: "))

print("Congrats! The number was: " + str(secret_number))

secret_number = 92
int
### Guess My Number

```python
guess_number = random.randint(1, 99)
print("I am thinking of a number between 1 and 99...")
guess = int(input("Enter a guess: "))
# True if guess is not equal to secret number
while guess != secret_number:
    # True if guess is less than secret number
    if guess < secret_number:
        print("Your guess is too low")
    else:
        print("Your guess is too high")

print("")  # an empty line
guess = int(input("Enter a new guess: "))

print("Congrats! The number was: " + str(secret_number))
```

---

![Diagram](image)
secret_number = random.randint(1, 99)
print("I am thinking of a number between 1 and 99...")
guess = int(input("Enter a guess: "))
# True if guess is not equal to secret number
while guess != secret_number:
  # True if guess is less than secret number
  if guess < secret_number:
    print("Your guess is too low")
  else:
    print("Your guess is too high")

print("") # an empty line
guess = int(input("Enter a new guess: "))

print("Congrats! The number was: " + str(secret_number))
secret_number = random.randint(1, 99)
print("I am thinking of a number between 1 and 99...")
guess = int(input("Enter a guess: "))
# True if guess is not equal to secret number
while guess != secret_number:
    # True if guess is less than secret number
    if guess < secret_number:
        print("Your guess is too low")
    else:
        print("Your guess is too high")

print("") # an empty line
guess = int(input("Enter a new guess: "))

print("Congrats! The number was: " + str(secret_number))
secret_number = random.randint(1, 99)
print("I am thinking of a number between 1 and 99...")
guess = int(input("Enter a guess: "))
# True if guess is not equal to secret number
while guess != secret_number:
    # True if guess is less than secret number
    if guess < secret_number:
        print("Your guess is too low")
    else:
        print("Your guess is too high")

print(""") # an empty line
guess = int(input("Enter a new guess: "))

print("Congrats! The number was: " + str(secret_number))
secret_number = random.randint(1, 99)
print("I am thinking of a number between 1 and 99...")
guess = int(input("Enter a guess: "))
# True if guess is not equal to secret number
while guess != secret_number:
    # True if guess is less than secret number
    if guess < secret_number:
        print("Your guess is too low")
    else:
        print("Your guess is too high")
print(""") # an empty line
guess = int(input("Enter a new guess: "))
print("Congrats! The number was: " + str(secret_number))
**Guess My Number**

```python
secret_number = random.randint(1, 99)
print("I am thinking of a number between 1 and 99...")
guess = int(input("Enter a guess: "))

# True if guess is not equal to secret number
while guess != secret_number:
    # True if guess is less than secret number
    if guess < secret_number:
        print("Your guess is too low")
    else:
        print("Your guess is too high")

print("")  # an empty line

# Case scenarios and code execution

# During execution, if a guess is too low:
# guess = int(input("Enter a new guess: "))

print("Congrats! The number was: " + str(secret_number))
```

```
- guess: 95
- secret_number: 92
  ```
secret_number = random.randint(1, 99)
print("I am thinking of a number between 1 and 99...")
guess = int(input("Enter a guess: "))
# True if guess is not equal to secret number
while guess != secret_number:
    # True if guess is less than secret number
    if guess < secret_number:
        print("Your guess is too low")
    else:
        print("Your guess is too high")

print("") # an empty line
guess = int(input("Enter a new guess: "))

print("Congrats! The number was: " + str(secret_number))
secret_number = random.randint(1, 99)
print("I am thinking of a number between 1 and 99...")
guess = int(input("Enter a guess: "))
# True if guess is not equal to secret number
while guess != secret_number:
    # True if guess is less than secret number
    if guess < secret_number:
        print("Your guess is too low")
    else:
        print("Your guess is too high")

print("")  # an empty line
guess = int(input("Enter a new guess: "))

print("Congrats! The number was: " + str(secret_number))

---

- **guess**: `95`
- **int**: `95`
- **secret_number**: `92`
guess_my_number = random.randint(1, 99)
print("I am thinking of a number between 1 and 99...")
guess = int(input("Enter a guess: "))
# True if guess is not equal to secret number
while guess != secret_number:
    # True if guess is less than secret number
    if guess < secret_number:
        print("Your guess is too low")
    else:
        print("Your guess is too high")

print("") # an empty line
guess = int(input("Enter a new guess: "))

print("Congrats! The number was: " + str(secret_number))
```
secret_number = random.randint(1, 99)
print("I am thinking of a number between 1 and 99...")
guess = int(input("Enter a guess: "))

# True if guess is not equal to secret number
while guess != secret_number:
    # True if guess is less than secret number
    if guess < secret_number:
        print("Your guess is too low")
    else:
        print("Your guess is too high")

print(""")  # an empty line
guess = int(input("Enter a new guess: "))

print("Congrats! The number was: " + str(secret_number))
```
secret_number = random.randint(1, 99)
print("I am thinking of a number between 1 and 99...")
guess = int(input("Enter a guess: "))
# True if guess is not equal to secret number
while guess != secret_number:
    # True if guess is less than secret number
    if guess < secret_number:
        print("Your guess is too low")
    else:
        print("Your guess is too high")

print("Congrats! The number was: " + str(secret_number))
Guess My Number

```python
secret_number = random.randint(1, 99)
print("I am thinking of a number between 1 and 99...")
guess = int(input("Enter a guess: "))
# True if guess is not equal to secret number
while guess != secret_number:
    # True if guess is less than secret number
    if guess < secret_number:
        print("Your guess is too low")
    else:
        print("Your guess is too high")

print("") # an empty line

# an empty line

guess = int(input("Enter a new guess: "))

print("Congrats! The number was: " + str(secret_number))
```

92  int  92  int  
 guess  secret_number
guess_number = random.randint(1, 99)
print("I am thinking of a number between 1 and 99...")
guess = int(input("Enter a guess: "))
# True if guess is not equal to secret number
while guess != secret_number:
    # True if guess is less than secret number
    if guess < secret_number:
        print("Your guess is too low")
    else:
        print("Your guess is too high")

print("") # an empty line
guess = int(input("Enter a new guess: "))

print("Congrats! The number was: " + str(secret_number))
secret_number = random.randint(1, 99)
print("I am thinking of a number between 1 and 99...")
guess = int(input("Enter a guess: "))

while guess != secret_number:
    if guess < secret_number:
        print("Your guess is too low")
    else:
        print("Your guess is too high")

print("") # an empty line

print("Congrats! The number was: " + str(secret_number))
Guess My Number

```python
secret_number = random.randint(1, 99)
print("I am thinking of a number between 1 and 99...")
guess = int(input("Enter a guess: "))
# True if guess is not equal to secret number
while guess != secret_number:
    # True if guess is less than secret number
    if guess < secret_number:
        print("Your guess is too low")
    else:
        print("Your guess is too high")
print("Congrats! The number was: " + str(secret_number))
```

```python
print("Congrats! The number was: " + str(secret_number))
```
• **sentinel**: A value that signals the end of user input.
  – **sentinel loop**: Repeats until a sentinel value is seen.

• Example: Write a program that prompts the user for numbers until the user types -1, then output the total of the numbers.
  – In this case, -1 is the sentinel value.

  Type a number: 10
  Type a number: 20
  Type a number: 30
  Type a number: -1
  total is 60
Example: Sentinel Loops

# fencepost problem!
# ask for number - post
# add number to total - fence

total = 0
while num != -1:
    num = int(input("Enter a number: "))
    total += num

print("total is " + total)
# Example: Sentinel Loops

```python
# fencepost problem!
# ask for number - post
# add number to total - fence

total = 0
num = int(input("Enter a number: "))
while num != -1:
    total += num
    num = int(input("Enter a number: "))

print("total is " + total)
```
# fencepost problem!
# ask for number - post
# add number to total - fence

total = 0
num = int(input("Enter a number: "))
while num != -1:
    total += num
    num = int(input("Enter a number: "))

print("total is " + total)
# fencepost problem!
# ask for number - post
# add number to total - fence

total = 0
num = int(input("Enter a number: 

while num != -1:
    total += num
    num = int(input("Enter a number: 

print("total is " + total)
# fencepost problem!
# ask for number - post
# add number to total - fence

```python
total = 0
num = int(input("Enter a number: "))
while num != -1:
    total += num
    num = int(input("Enter a number: "))

print("total is "+total)
```
# fencepost problem!
# ask for number - post
# add number to total - fence

```python
total = 0
num = int(input("Enter a number: "))
while num != -1:
    total += num
    num = int(input("Enter a number: "))

print("total is " + str(total))
```

Example: Sentinel Loops
Example: Sentinel Loops

# fencepost problem!
# ask for number - post
# add number to total - fence

total = 0
num = int(input("Enter a number: "))

while num != -1:
    total += num
    num = int(input("Enter a number: "))

print("total is " + total)
# fencepost problem!
# ask for number - post
# add number to total - fence

```python
total = 0
num = int(input("Enter a number: "))
while num != -1:
    total += num
    num = int(input("Enter a number: "))

print("total is " + str(total))
```

Example: Sentinel Loops
Example: Sentinel Loops

# fencepost problem!
# ask for number - post
# add number to total - fence

total = 0
num = int(input("Enter a number: "))
while num != -1:
    total += num
    num = int(input("Enter a number: "))

print("total is " + total)
# fencepost problem!
# ask for number - post
# add number to total - fence

total = 0
num = int(input("Enter a number: "))
while num != -1:
    total += num
    num = int(input("Enter a number: "))
print("total is " + str(total))
# fencepost problem!
# ask for number - post
# add number to total - fence

total = 0
num = int(input("Enter a number: "))

while num != -1:
    total += num
    num = int(input("Enter a number: "))

print("total is " + total)
Example: Sentinel Loops

# fencepost problem!
# ask for number - post
# add number to total - fence

total = 0
num = int(input("Enter a number: "))
while num != -1:
    total += num
    num = int(input("Enter a number: "))

print("total is " + str(total))
Example: Sentinel Loops

# fencepost problem!
# ask for number - post
# add number to total - fence

total = 0
num = int(input("Enter a number: "))
while num != -1:
    total += num
    num = int(input("Enter a number: "))

print("total is " + total)
# fencepost problem!
# ask for number - post
# add number to total - fence

total = 0
num = int(input("Enter a number: "))
while num != -1:
    total += num
    num = int(input("Enter a number: "))

print("total is " + str(total))
# fencepost problem!
# ask for number - post
# add number to total - fence

total = 0
num = int(input("Enter a number: "))

while num != -1:
    total += num
    num = int(input("Enter a number: "))

print("total is " + str(total))
Example: Sentinel Loops

# fencepost problem!
# ask for number - post
# add number to total - fence

total = 0
num = int(input("Enter a number: "))
while num != -1:
    total += num
    num = int(input("Enter a number: "))

print("total is " + total)
Example: Sentinel Loops

# fencepost problem!
# ask for number - post
# add number to total - fence

total = 0
num = int(input("Enter a number: "))
while num != -1:
    total += num
    num = int(input("Enter a number: "))

print("total is " + total)
Example: Sentinel Loops

# fencepost problem!
# ask for number - post
# add number to total - fence

total = 0
while True:
    num = int(input("Enter a number: "))
    if num == -1:
        break  # immediately exits loop
    total += num
print("total is " + total)
Logical Operators

In order of precedence:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Example</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>not</td>
<td>not (2 == 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>(2 == 3) or (-1 &lt; 5)</td>
<td>True</td>
</tr>
</tbody>
</table>

Can "chain" tests as in algebra

# assume x is 15
2 <= x <= 10

# identical version
2 <= x and x <= 10
Precedence:

- arithmetic > comparison > not > and/or

5 * 7 >= 3 + 5 * (7 - 1) and not False

https://docs.python.org/3/reference/expressions.html#operator-totalmary
Precedence Madness

Precedence:

- arithmetic > comparison > not > and/or

5 * 7 >= 3 + 5 * (7 - 1) and not False

5 * 7 >= 3 + 5 * 6 and not False

https://docs.python.org/3/reference/expressions.html#operator-totalmary
Precedence:

arithmetic > comparison > not > and/or

5 * 7 >= 3 + 5 * (7 - 1) and not False
35 >= 3 + 5 * 6 and not False

https://docs.python.org/3/reference/expressions.html#operator-totalmary
Precedence:

arithmetic > comparison > not > and/or

5 * 7 >= 3 + 5 * (7 - 1) and not False
35 >= 3 + 5 * 6 and not False
35 >= 3 + 30 and not False

https://docs.python.org/3/reference/expressions.html#operator-totalmary
Precedence Madness

Precedence:

- arithmetic > comparison > not > and/or

- $5 \times 7 \geq 3 + 5 \times (7 - 1)$ and not False
- $35 \geq 3 + 5 \times 6$ and not False
- $35 \geq 3 + 30$ and not False
- $35 \geq 33$ and not False

https://docs.python.org/3/reference/expressions.html#operator-totalmary
Precedence:

arithmetic > comparison > not > and/or

5 * 7 >= 3 + 5 * (7 - 1) and not False
35 >= 3 + 5 * 6 and not False
35 >= 3 + 30 and not False
35 >= 33 and not False
True and not False

https://docs.python.org/3/reference/expressions.html#operator-totalmary
Precedence:

arithmetic > comparison > not > and/or

5 * 7 >= 3 + 5 * (7 - 1) and not False
35 >= 3 + 5 * 6 and not False
35 >= 3 + 30 and not False
35 >= 33 and not False
True and not False
True and True

https://docs.python.org/3/reference/expressions.html#operator-totalmary
Precedence:


\[
\begin{align*}
5 \times 7 &\geq 3 + 5 \times (7 - 1) \text{ and not False} \\
35 &\geq 3 + 5 \times 6 \text{ and not False} \\
35 &\geq 3 + 30 \text{ and not False} \\
35 &\geq 33 \text{ and not False} \\
\text{True and not False} \\
\text{True and True} \\
\text{True}
\end{align*}
\]

https://docs.python.org/3/reference/expressions.html#operator-totalmary
George Boole

English Mathematician teaching in Ireland 1815 – 1864
Boole died of being too cool
# Store expressions that evaluate to True/False

\[
x = 1 < 2 \quad \# \text{True}
\]
\[
y = 5.0 == 4.0 \quad \# \text{False}
\]
# Store expressions that evaluate to True/False
x = 1 < 2  # True
y = 5.0 == 4.0  # False

# Directly set to True/False
is_sheltering = True
is_raining = False
# Store expressions that evaluate to True/False
x = 1 < 2  # True
y = 5.0 == 4.0  # False

# Directly set to True/False
is_sheltering = True
is_raining = False

play_again = input('Play again? "y" or "n"') == 'y'
if play_again:
    ...

Piech, CS106A, Stanford University
Please... NO FOOD OR DRINKS

is_allowed = not food or drinks

*know your logical precedence
is_allowed = not food or drinks
False

*know your logical precedence
Please...

NO FOOD OR DRINKS

is_allowed = not food or drinks

False    True

*know your logical precedence
Today’s Route

You are here

Simple Java

For Loops

Booleans

Conditions

Review

The River of Java

Piech, CS106A, Stanford University
Today’s Route

The River of Java

Review

Conditions

Booleans

For Loops

Simple Java

You are here
How would you print “Python rocks socks” 100 times
def run() :
    for i in range(100):
        print(“Python rocks socks!”)
for i in range(100):
    print("Python rocks socks!")

i = 0
while i < 100:
    print("Python rocks socks!")
i += 1
For Loop Redux

Create a counting variable \( i \)

\[
\text{for } i \text{ in } \text{range}(100):
    \text{print(”Python rocks socks!”)}
\]

Which takes on the values 0 to 99 one at a time
for i in range(3):
    print(“Python rocks socks!”)
For Loop Redux

```python
for i in range(3):
    print("Python rocks socks!")
```

`range(3) -> 0, 1, 2`
For Loop Redux

```
for i in range(3):
    print("Python rocks socks!")
```

range(3) -> 0, 1, 2
For Loop Redux

i  0

range(3) -> 0, 1, 2

```python
for i in range(3):
    print("Python rocks socks!")
```

Python rocks socks
For Loop Redux

```
for i in range(3):
    print("Python rocks socks!")
```

Output:

```
Python rocks socks
```
for loop redux

for i in range(3):
    print("Python rocks socks!")

range(3) -> 0, 1, 2
For Loop Redux

\[
\begin{array}{|c|}
\hline
i & 1 \\
\hline
\end{array}
\]

range(3) → 0, 1, 2

for \( i \) in range(3):
    print("Python rocks socks!")

Python rocks socks
Python rocks socks
For Loop Redux

i  1

range(3) -> 0, 1, 2

```python
for i in range(3):
    print("Python rocks socks!")
```

Python rocks socks
Python rocks socks
For Loop Redux

```
for i in range(3):
    print("Python rocks socks!")
```

```
Python rocks socks
Python rocks socks
```
For Loop Redux

for i in range(3):
    print("Python rocks socks!")

range(3) -> 0, 1, 2
For Loop Redux

```
for i in range(3):
    print("Python rocks socks!")
```

```python
Python rocks socks
Python rocks socks
Python rocks socks
```
For Loop Redux

for i in range(3):
    print("Python rocks socks!")

range(3) -> 0, 1, 2

Python rocks socks
Python rocks socks
Python rocks socks
You can use the for loop variable
How would you print the first 100 even numbers?
Printing Even Numbers

0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
for i in range(3):
    print(i * 2)
Printing Even Numbers

```python
for i in range(3):
    print(i * 2)
```
Printing Even Numbers

```
for i in range(3):
    print(i * 2)
```

i | 0

---

Piech, CS106A, Stanford University
Printing Even Numbers

```
for i in range(3):
    print(i * 2)
```

i 0
Printing Even Numbers

```
for i in range(3):
    print(i * 2)
```

For Loop Redux

0
Printing Even Numbers

```
for i in range(3):
    print(i * 2)
```

0
Printing Even Numbers

```python
i 1

for i in range(3):
  print(i * 2)
```

For Loop Redux

0
2
Printing Even Numbers

```python
for i in range(3):
    print(i * 2)
```

Output:

```
0
2
```
Printing Even Numbers

for i in range(3):
    print(i * 2)
Printing Even Numbers

```
for i in range(3):
    print(i * 2)
```

For Loop Redux

```
0
2
4
```
Printing Even Numbers

```
for i in range(3):
    print(i * 2)
```

Output:

```
0
2
4
```
Printing Even Numbers

```python
for i in range(3):
    print(i * 2)
```

```
0
2
4
```
# our solution  
for i in range(3):
    print(i * 2)

# equivalently
for i in range(0, 6, 2):
    print(i)
Welcome to the CS106A game show!
Choose a door and win a prize
Door: 2
You chose door 2
You win $[blackacted]

* To be delivered via amazon gift cards

Piech, CS106A, Stanford University
Choose a Door

door = int(input("Door: "))
# while the input is invalid
while door < 1 or door > 3 :
    # tell the user the input was invalid
    print("Invalid door!"")
    # ask for a new input
    door = int(input("Door: "))

or
and
prize = 4

if door == 1:
    prize = 2 + 9 // 10 * 100

elif door == 2:
    locked = prize % 2 != 0
    if not locked:
        prize += 6

elif door == 3:
    for i in range(door):
        prize += i
That’s all

def main() :
    for i in range(999999):
        print(“You rock!”)
    print(“See you on Monday”)

Piech, CS106A, Stanford University
Today’s Route

Simple Java

For Loops

Booleans

Conditions

Review

You are here

The River of Java

Piech, CS106A, Stanford University
Today’s Goal

1. Be able to use For / While / If in Java