Kickstart 2012

Day 1
Intro and Basics

Stephanie Rogers and Amy Pavel
Welcome 😊

- Who are we?
What is Computer Science?

- Problem Solving
- Building things

- CS is everywhere
  - Internet
  - Phone/Web Applications
  - Vehicles
  - Genetics
  - And more!
What is Computer Science? (cont)

- Programming
  - Art and science of constructing artifacts that perform computations
  - Programming languages
What is Kickstart?

- Not 1’s and 0’s
- Implementing programs
- Producing a tangible result!
  - PICTURES
- An Intro to Jython
Programming Languages

- Communication with computers
- Different encodings of instructions for machines
- The language we are using: Jython
  - Jython is Python!
  - Java-based Python
- Ice Breaker – Partner in common (talent)
- Logins!
Environment - JES

- **Jython Environment for Students**
- Programming area: the editor, writing programs
- Command area: Entering commands
Meet Jython - Data

**Data:** stuff we manipulate

- integers:  2  -1  13
- strings:  “hello world”
- booleans:  true, false
- lists:  [1, 2, 3]
- More later

```python
>>> 2
2
>>> "hello world"
'hello world'
```
Meet Jython - Functions

**Functions**: rules for manipulate data

- Primitive expressions: +, -, *, /, ...
- Built-in functions: sum, abs, ...
- Self-defined function: def square(x): ...

Can take any number of arguments
Meet Jython - Expressions

Expressions

- Combining functions with data
- Jython evaluates these expressions for you

```python
>>> 2+3
?
>>> sum(2, 3)
?
>>> abs(-2)
?
>>> print('hello world')
?
```
Calling functions

- **Remember**
  - Functions: rules for manipulating data
  - Can take any number of arguments

```python
>>> x = sum(4, 3)
>>> y = abs(-9)
>>> max(x, y)
9

>>> Can we do all this in one line?
```
Calling functions

- **Remember**
  - Functions: rules for manipulating data
  - Can take any number of arguments

```python
>>> x = sum(4, 3)
>>> y = abs(-9)
>>> max(x, y)
9
```

```python
>>> Can we do all this in one line?

Nesting
```
Calling functions

- **Remember**
  - Functions: rules for manipulating data
  - Can take any number of arguments

```python
>>> x = sum(4, 3)
>>> y = abs(-9)
>>> max(x, y)
9

>>> max( sum(4, 3) , abs(-9) )
?
```
Meet Jython - Numbers

+, -, *, /, %, >, >=, ==, !=, <, <=

>>> 2 + 3
5

>>> (5 * 8) + 2
42

>>> 40 / 5
8.0

>>> 11 % 3
2

>>> 4 > 3
True

>>> 6 <= 5
False

>>> 6 == (3+3)
True

>>> 6 != 5
True
Meet Jython – Logic

**Booleans:** True, False

**Logical operators:** and, or, not, >, >= ...

<table>
<thead>
<tr>
<th>and</th>
<th>or</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;&gt;&gt; (4 &gt; 3) and (4 &lt; 5)</td>
<td>&gt;&gt;&gt; (4 &gt; 3) or (4 &gt; 5)</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>&gt;&gt;&gt; True and False</td>
<td>&gt;&gt;&gt; False or False</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>&gt;&gt;&gt; True and True</td>
<td>&gt;&gt;&gt; True or False</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>
Meet Jython - Assignment

Variables
Name our data and functions for use later

```python
>>> x = 3
>>> print x + 1
4
```
Meet Jython - Strings

Indexing + concatenation

```python
>>> "hi " + "stephanie"
??

>>> print("hello, world")
??

>>> name = "stephanie"

>>> name[0]
??
```
Meet Jython - Lists

Indexing & concatenation

```python
>>> [1, 2, 3, 4]

>>> [1, 2] + [3, 4]

>>> alphabet = ['a', 'b', 'c']

>>> alphabet[2]
```

```python
??
```
Administrivia

- Website: inst.eecs.berkeley.edu/~cs98-tr
- Lab Structure
  - Lecture
  - Interactive practice - labs
  - Projects
- Send us your pictures daily!
Try it yourself

- Lab Exercise 0 & 1
Pictures
Picture Functions

- `pickAFile()`
  - Allows the user to pick a file
  - Takes no argument!
pickAFile() leads to... The File Picker! - UI
Picture Functions

- `pickAFile()`

- `makePicture(filename)`
  - creates and returns a picture object

- `show(picture)`
  - displays a picture in a window
Showing a picture

- **Steps**
  1. Choose a file
  2. Make it into a “picture”
  3. Show the picture

```java
myFile = pickAFile()
pic = makePicture(myFile)
show(pic)
```

**Alt: Nesting**

```java
show(makePicture(pickAFile()))
```
Defining our own functions

def <name>(<arguments names>):
    return <expression>

- Functions:
  - function name
  - input values
  - Body
Defining our own functions

Structure of a function

- `def`
- `function name`
- `input values between parentheses`
- `colon`
- `body` (indentation matters = 2 spaces)

```python
def addSquares(x, y):
    squareX = x*x
    squareY = y*y
    return squareX + squareY
```

Nesting?
Blocking is indicated for you in JES

- Statements with same indentation = same block
- same block is enclosed in a blue box

**DEMO**
Try it yourself

- Lab Exercise 2 & 3

- (~15-20 minutes)