CSE 115
Introduction to Computer Science I
Announcements

Lab activities/Lab exams

submit regularly to autograder.cse.buffalo.edu
Announcements

Lab activities/Lab exams

submit regularly to autograder.cse.buffalo.edu

autograder enforces deadlines strictly
Announcements

Lab activities/Lab exams

submit regularly to autograder.cse.buffalo.edu

autograder enforces deadlines strictly

lab activities and lab exams are INDIVIDUAL WORK
Announcements

Lab activities/Lab exams

submit regularly to autograder.cse.buffalo.edu

autograder enforces deadlines strictly

lab activities and lab exams are INDIVIDUAL WORK

we will address submission issues / week 1 glitches
Announcements

Lab exams

You have 55 minutes to complete your work.

The second half of the recitation is reserved for other tasks/activities.
Road map

▶ Review ◄

control flow (sequencing, selection, repetition)

sequencing

selection
Review

relational expressions

Boolean expressions
## Relational operators

https://docs.python.org/3.7/library/stdtypes.html#comparisons

<table>
<thead>
<tr>
<th>Operation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;</td>
<td>strictly less than</td>
</tr>
<tr>
<td>&lt;=</td>
<td>less than or equal</td>
</tr>
<tr>
<td>&gt;</td>
<td>strictly greater than</td>
</tr>
<tr>
<td>&gt;=</td>
<td>greater than or equal</td>
</tr>
<tr>
<td>==</td>
<td>equal</td>
</tr>
<tr>
<td>!=</td>
<td>not equal</td>
</tr>
</tbody>
</table>
### Boolean operators

https://docs.python.org/3.7/library/stdtypes.html#boolean-operations-and-or-not

<table>
<thead>
<tr>
<th>Operation</th>
<th>Result</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>x or y</code></td>
<td>if <code>x</code> is false, then <code>y</code>, else <code>x</code></td>
<td>(1)</td>
</tr>
<tr>
<td><code>x and y</code></td>
<td>if <code>x</code> is false, then <code>x</code>, else <code>y</code></td>
<td>(2)</td>
</tr>
<tr>
<td><code>not x</code></td>
<td>if <code>x</code> is false, then <code>True</code>, else <code>False</code></td>
<td>(3)</td>
</tr>
</tbody>
</table>

**Notes:**

1. This is a short-circuit operator, so it only evaluates the second argument if the first one is false.
2. This is a short-circuit operator, so it only evaluates the second argument if the first one is true.
3. `not` has a lower priority than non-Boolean operators, so `not a == b` is interpreted as `not (a == b)`, and `a == not b` is a syntax error.
Boolean expressions

examples

True or False

a and b

x < y and y <= z

Convenient, but unusual across languages.

x < y <= z
Road map

Review

▶ control flow (sequencing, selection, repetition) ▼

sequencing

selection
Control flow

SEQUENCING

SELECTION

REPETITION
Control flow

Sequencing

Statements in a block are executed in sequence.

(i.e. one after the other, in the order written)

This is what we've seen so far.
Control flow

Selection

One of a set of instructions is executed, based on the outcome of a decision.

Exactly one of many possible branches is followed.
Control flow

Repetition

A block is repeated several times, based on the outcome of a decision.

Also called looping.
We use "flow charts" to help visualize the flow of control through a program, when appropriate.
\[ a = 12 \]
\[ b = 2 \times a + 1 \]
\[ c = b - a \]
Control flow
visualizing sequences

\[ a = 12 \]
\[ b = 2 \times a + 1 \]
\[ c = b - a \]
Control flow
visualizing sequences

\[
\begin{align*}
a &= 12 \\
b &= 2 \times a + 1 \\
c &= b - a
\end{align*}
\]
Control flow

Selection: the if statement

Here's an example of an if statement:

```python
if x < y :
    z = y
```
Control flow

Selection: the if statement

'if' is a keyword

```python
if x < y :
    z = y
```
Control flow

Selection: the if statement

'x < y' is a Boolean expression

```python
if x < y:
    z = y
```
Control flow

Selection: the if statement

\[
\text{if } x < y \\
\quad z = y
\]

:: is a delimiter
Control flow

Selection: the if statement

What follows the ':' is a block (sequence) of statements.

```
if x < y :
    z = y
```

Recall
Python refers to the a block of statements as a "suite".
Control flow
visualizing selection (if)

\[ x = 12 \]
\[ y = 14 \]

\[
\text{if } x < y : \\
\quad z = y \\
a = (x + y) - z
\]
Control flow
visualizing selection (if)

\[
x = 12
\]
\[
y = 14
\]
\[
\text{if } x < y :
\]
\[
z = y
\]
\[
a = (x + y) - z
\]
Control flow
visualizing selection (if)

\[ x = 12 \]
\[ y = 14 \]
\[ \text{if } x < y : \]
\[ \quad z = y \]
\[ a = (x + y) - z \]
Control flow
visualizing selection (if)

\[ x = 12 \]
\[ y = 14 \]
\[ \text{if } x < y : \]
\[ \quad z = y \]
\[ a = (x + y) - z \]
x = 12
y = 14
if x < y:
    z = y
else:
    z = x
a = (x + y) - z
Control flow
visualizing selection (if-else)

\[
x = 12 \\
y = 14 \\
\text{if } x < y : \\
\quad z = y \\
\text{else :} \\
\quad z = x \\
a = (x + y) - z
\]
Control flow

visualizing selection (if-else)

\[
x = 12 \\
y = 14 \\
\textbf{if } x < y : \\
\quad z = y \\
\textbf{else :} \\
\quad z = x \\
a = (x + y) - z
\]
Control flow
visualizing selection (if-else)

```
x = 12
y = 14
if x < y :
  z = y
else :
  z = x
a = (x + y) - z
```

```
x = 12
y = 14
x < y
    True
        z = y
    False
        z = x
```

```
a = (x + y) - z
```
Control flow
visualizing selection (if-else)

\[
x = 12 \\
y = 14 \\
\text{if } x < y : \\
\quad z = y \\
\text{else :} \\
\quad z = x \\
a = (x + y) - z
\]
Control flow

visualizing selection (if-elif-else)

\[ x = 12 \]
\[ y = 14 \]

```python
if x < y :
    z = y
elif x == y :
    z = 0
else :
    z = x
```

\[ a = (x + y) - z \]
Control flow

visualizing selection (if-elif-else)

\[ x = 12 \]
\[ y = 14 \]

if \( x < y \):
    \[ z = y \]

elif \( x == y \):
    \[ z = 0 \]

else:
    \[ z = x \]

\[ a = (x + y) - z \]
Control flow

visualizing selection (if-elif-else)

\[
x = 12 \\
y = 14 \\
\text{if } x < y : \\
\quad z = y \\
\text{elif } x == y : \\
\quad z = 0 \\
\text{else :} \\
\quad z = x \\
a = (x + y) - z
\]
Control flow

visualizing selection (if-elif-else)

\[
x = 12 \\
y = 14 \\
\text{if } x < y : \\
    z = y \\
\text{elif } x == y : \\
    z = 0 \\
\text{else : } \\
    z = x \\
a = (x + y) - z
\]
Control flow

visualizing selection (if-elif-else)

\[
x = 12
\]
\[
y = 14
\]

\[
\text{if } x < y : \\
\quad z = y
\]

\[
\text{elif } x == y : \\
\quad z = 0
\]

\[
\text{else : } \\
\quad z = x
\]

\[
a = (x + y) - z
\]