Agenda

• Announcements
• Overview Course Topics
• Range function
• List
• BMI
Course topics

• Variables
• Types and values
• String
• Function
• For loop
• If
• While loop
• Lists
• Set

• Tuples
• Dictionaries
• Sorting
• Recursion
range(start, end, increment)

• range(...) simply gives a list of numbers

\[
\text{range}(10) \\
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
\]

\[
\text{range}(2, 8) \\
[2, 3, 4, 5, 6, 7]
\]

\[
\text{range}(0, 10, 2) \\
[0, 2, 4, 6, 8]
\]

Note that 10 is not included.

Only 1 number makes it start from 0.

Increment by 2
Lists are simple (we love them)

```python
lst = []
points = [982, 234, -123]
zoo = ['snake', 'hog', 'tiger']
```

Example…
Lists under the hood

```python
zoo = ["snake", "hog", "tiger"]
```

<table>
<thead>
<tr>
<th>&quot;snake&quot;</th>
<th>&quot;hog&quot;</th>
<th>&quot;tiger&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

```python
for animal in zoo:
    print animal
```

snake
hog
tiger

Access individual elements

For loop for easy run-through of content
# Often Used List Functions

<table>
<thead>
<tr>
<th>ACCESS</th>
<th>MODIFY</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{item = lst[index]}</td>
<td>\texttt{lst(i) = “dog”}</td>
</tr>
<tr>
<td>Getting a specific item out of the list</td>
<td>Overwrites “dog” on the i’th place</td>
</tr>
<tr>
<td>\texttt{more = lst[start:stop]}</td>
<td>\texttt{lst.append(…)}</td>
</tr>
<tr>
<td>Slicing several items out of the list with index start &amp; stop</td>
<td>Append an element to lst, changing lst</td>
</tr>
<tr>
<td>\texttt{n = len(lst)}</td>
<td>\texttt{newLst = lst[:]}</td>
</tr>
<tr>
<td>Getting the length of the list. Number of item in it.</td>
<td>Makes a copy of lst into newLst</td>
</tr>
<tr>
<td>\texttt{lst.pop(index)}</td>
<td>\texttt{lst.extend(lst2)}</td>
</tr>
<tr>
<td>Remove and return element at position index in lst, so has side-effect of altering list and returns value. If you don’t give an index the last value is automatically popped.</td>
<td>Append every element of lst2 to lst</td>
</tr>
</tbody>
</table>
### Often Used List Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>if “dog” in zoo:</code></td>
<td><code>print dog</code> in keyword finds the element in the list.</td>
</tr>
<tr>
<td><code>lst.index(elt)</code></td>
<td>return index of <code>elt</code> in <code>lst</code>, error if <code>elt</code> not in <code>lst</code></td>
</tr>
<tr>
<td><code>lst.count(elt)</code></td>
<td>return number of occurrences of <code>elt</code> in <code>lst</code></td>
</tr>
<tr>
<td><code>max(lst)</code></td>
<td>returns maximal element in <code>lst</code></td>
</tr>
<tr>
<td><code>min(lst)</code></td>
<td>returns minimum element in <code>lst</code></td>
</tr>
<tr>
<td><code>sum(lst)</code></td>
<td>returns sum of elements in list <code>lst</code></td>
</tr>
<tr>
<td><code>lst.sort()</code></td>
<td>sorts the elements of <code>lst</code></td>
</tr>
</tbody>
</table>
In-Class Assignment

Write a tiny program that asks and calculate 1 persons BMI

1) Ask the users weight and height (pound, inch)
2) Save in tiny list
3) Use formula to calculate BMI (picking data from list):
   Pounds / (inch * inch) * 705
4) Print BMI
Alma Whitten

- Google: Engineering Lead for Privacy, Director for Privacy
  - Across marketing and engineering
  - Why Johnny Can't Encrypt

"It's more and more the case that every individual is going around with a cheap yet powerful data-capture device, and the ability to connect that device to powerful data services. ... We need to build powerful information tools that are equally available to everyone, and if we are sufficiently transparent about (those tools and how they use data), then the outside world gets to say, yes, it's worth it, we want those tools."