Model 1  Lists

A variable can hold multiple values in the form of a list. The values are separated by commas and wrapped in square brackets. For example:

\[
\text{primes} = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29]
\]

Each element of the list can be referenced by an index, which is the sequential position starting at 0. For example, \text{primes}[4] is 11.

<table>
<thead>
<tr>
<th>index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>11</td>
<td>13</td>
<td>17</td>
<td>19</td>
<td>23</td>
<td>29</td>
</tr>
</tbody>
</table>

Do not type anything yet! Read the questions first!

<table>
<thead>
<tr>
<th>Python code</th>
<th>Shell output</th>
</tr>
</thead>
<tbody>
<tr>
<td>odd = [1, 3, 5, 7]</td>
<td></td>
</tr>
<tr>
<td>odd</td>
<td></td>
</tr>
<tr>
<td>odd[2]</td>
<td></td>
</tr>
<tr>
<td>odd[4]</td>
<td></td>
</tr>
<tr>
<td>len(odd)</td>
<td></td>
</tr>
<tr>
<td>number = odd[1]</td>
<td></td>
</tr>
<tr>
<td>number</td>
<td></td>
</tr>
<tr>
<td>odd[1] = 2</td>
<td></td>
</tr>
<tr>
<td>odd</td>
<td></td>
</tr>
<tr>
<td>number</td>
<td></td>
</tr>
</tbody>
</table>

Questions (10 min)  

1. What is the index of the second element of \text{primes}? What is the value at that index?

2. How does the index number compare to the position of the element?
3. Type each line of code in a Python Shell and write the corresponding output in the space above. If an error occurs, write what type of error. Place an asterisk (*) next to any output for which you were surprised.

4. How did you reference the value of the 3rd element of `odd`?

5. What did the output of the `len()` function tell you about the list?

6. The output of Model 1 displayed an error. Explain the reason for the error.

7. Write a statement that assigns a list of three integers to the variable `run`.

8. Write a statement that assigns the value 100 to the last element of `run`.

9. Write a statement that assigns the first value of `run` to a variable named `first`. 
# Model 2  Sequences

Lists and strings are examples of sequence types. Complete the table below to explore how sequences work.

<table>
<thead>
<tr>
<th>Python code</th>
<th>Shell output</th>
</tr>
</thead>
<tbody>
<tr>
<td>seq1 = &quot;one two&quot;</td>
<td></td>
</tr>
<tr>
<td>type(seq1)</td>
<td></td>
</tr>
<tr>
<td>len(seq1)</td>
<td></td>
</tr>
<tr>
<td>seq1[1]</td>
<td></td>
</tr>
<tr>
<td>seq1[1] = '1'</td>
<td></td>
</tr>
<tr>
<td>seq2 = &quot;one&quot;, &quot;two&quot;</td>
<td></td>
</tr>
<tr>
<td>type(seq2)</td>
<td></td>
</tr>
<tr>
<td>len(seq2)</td>
<td></td>
</tr>
<tr>
<td>seq2[1]</td>
<td></td>
</tr>
<tr>
<td>seq2[1] = '1'</td>
<td></td>
</tr>
<tr>
<td>seq3 = [&quot;one&quot;, &quot;two&quot;]</td>
<td></td>
</tr>
<tr>
<td>type(seq3)</td>
<td></td>
</tr>
<tr>
<td>seq3[1]</td>
<td></td>
</tr>
<tr>
<td>seq3[1] = 1</td>
<td></td>
</tr>
<tr>
<td>seq4 = (&quot;one&quot;, 1)</td>
<td></td>
</tr>
<tr>
<td>type(seq4)</td>
<td></td>
</tr>
<tr>
<td>number = 12345</td>
<td></td>
</tr>
<tr>
<td>number[3]</td>
<td></td>
</tr>
</tbody>
</table>

### Questions (15 min)

10. How does a sequence type differ from a number? (See the last row of the table.)

11. What are the names of the three sequence types introduced in Model 2?
12. How does the syntax of creating a tuple differ from creating a list?

13. Is there more than one way (syntax) to create a tuple? Justify your answer.

14. Which sequence types allow their elements to be changed? Which do not?

15. Is it possible to store values of different types in a sequence? If yes, give an example from the table; if no, explain why not.

16. Summarize the difference between lists and tuples. How do they look differently, and how do they work differently?