Python Webcast #4
Overview

- Numbers and the math library
- Strings and slicing
- String functions
- Example problem
Example Number Variables

- x = 25
- y = 10000
- Variable1 = 23.4
- variable2 = 32.8
More number operations

- Some math functions are not built-in to Python, so they need to be imported.
- In order to import a “library” (set of functions/tools/objects), just type:
  - `import math`
- This library has several functions you can play with
  - `sqrt`, `log`, `factorial`, `trig functions`
- To get a full list of what a library has, type:
  - `dir(math)`
    ```
    ```
Using Math functions

- `math.sqrt(x)`
  - 5.0
  - Why is it 5.0? Because the sqrt function can only return float numbers

- `math.log10(y)`
  - 4.0

- You can also use the built-in numbers in math:
  - `math.pi`
    - 3.141592653589793
    - Notice that there are no (), because this is just a number, not a function
Strings

- First_string = “The”
- You can access individual characters in a string by referring to their index.
- The index is a zero-based counting system, where the first character is 0, the second is 1, and so on.
- First_string[0]
  ‘T’
String “slicing”

- You can also select a substring from a larger string. Use this table as an example:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[0]</td>
<td>[1]</td>
<td>[2]</td>
</tr>
<tr>
<td>‘T’</td>
<td>‘h’</td>
<td>‘e’</td>
</tr>
</tbody>
</table>

- Slice: 0
- First_string[1:3]
  ‘he’
More about Strings

- You can determine the length of a string by using the `len()` function:
  - `Sequence = 'ACATGGTGTTCAATACT'`
  - `len(Sequence)`
    - 17
- You can find the position of a substring in the larger string using the `str.find()` function
  - `Sequence.find('ATG')`
    - 2
- You can count the occurrences of a character in a string using `str.count()`
  - `Sequence.count('G')`
Example Problem walk-through

- Given this sequence:
  "ACGTATGC GCCG GGCATATA TACGATAGG C"
- Find the GC content (%) of the sequence
- \texttt{seq} = "CATGCCCCCCG GGCATATA TACGATAGG C"
- \texttt{seqlen} = \texttt{len(seq)}
- \texttt{G\_count} = \texttt{seq.count('G')}
- \texttt{C\_count} = \texttt{seq.count('C')}
- \texttt{GC\_count} = \texttt{C\_count} + \texttt{G\_count}
- \texttt{GC\_content} = \texttt{GC\_count} / \texttt{float(seqlen)}