

## ***Activity 3: Types***

Python has a wide variety of built-in types for storing anything from numbers and text (e.g., `int`, `float`, `str`) to common data structures (e.g., `list`, `tuple`, `dict`).

### **Content Learning Objectives**

*After completing this activity, students should be able to:*

- Identify the name and syntax defining basic data types.
- Reference a specific element of a sequence by an index.
- Compare and contrast numeric and sequence data types.

### **Process Learning Objectives**

*After completing this activity, students should make progress toward:*

- Providing feedback on how well other team members are working. (Teamwork)



## Model 1 Numbers

Every value in Python belongs to a **data type** which determines what can be done with the data. Enter the following code, one line at a time, into a Python Shell. Record the output for each line (if any) in the second column.

Python code	Shell Output
<code>integer = 3</code>	
<code>type(integer)</code>	
<code>type("integer")</code>	
<code>pi = 3.1415</code>	
<code>type(pi)</code>	
<code>word = str(pi)</code>	
<code>word</code>	
<code>number = float(word)</code>	
<code>print(word * 2)</code>	
<code>print(number * 2)</code>	
<code>print(word + 2)</code>	
<code>euler = 2.7182</code>	
<code>int(euler)</code>	
<code>round(euler)</code>	

### Questions (15 min)

start  
time:

1. Guess what the data type is (`int`, `float`, or `str`) of the following values? (Check with



the `type` function in a Python Shell.)

- a. `pi`
  - b. `integer`
  - c. `word`
  - d. `number`
2. List the function calls that convert a value to a new data type.
  3. How can we tell if something is a string when we type it?
  4. What is the value of `3 + 3 + 3`? What is the value of `.3 + .3 + .3`? If you enter these expressions into a Python Shell, what do you notice about the results?
  5. What data type would we use for 123.45?
  6. What is the difference between the numeric data types (`int` and `float`)



## Model 2 Lists

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

```
primes = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29]
```

Each *element* of the list can be referenced by an *index*, which is its position starting at 0. For example, `primes[4]` is 11.

index	0	1	2	3	4	5	6	7	8	9
value	2	3	5	7	11	13	17	19	23	29

**Do not type anything yet! Read the questions first!**

Python code	Shell Output
<code>odd = [1, 3, 5, 7]</code>	
<code>odd</code>	
<code>odd[2]</code>	
<code>odd[4]</code>	
<code>len(odd)</code>	
<code>number = odd[1]</code>	
<code>number</code>	
<code>odd[1] = 2</code>	
<code>odd</code>	
<code>number</code>	

### Questions (15 min)



7. What is the index of the second element of `primes`? What is the value at that index?
8. Type each line of code in a Python Shell and write the corresponding output in the space above. If an error occurs, raise your hand. Place an asterisk (\*) next to any output that surprised you.
9. How do you reference the value of the 3rd element of `odd`?
10. What does the output of the `len()` function tell you about the list?
11. The output of the **Model 2** displayed an error. Explain the reason for the error.
12. Write a statement that assigns a list of three integers to the variable named `run`.
13. Write a statement that assigns the value 100 to the last element of `run`.
14. Write a statement that assigns the first value of `run` to a variable named `first`.

