Containers

Python has built-in data types for storing anything from numbers and text (Ex: `int`, `float`, `str`) to common data structures (Ex: `list`, `set`, `dict`).

Manager: Recorder:

Presenter: Reflector:

Content Learning Objectives

After completing this activity, students should be able to:

- Create a list and access a specific element using an index.
- Describe operations that can be applied to a set of strings.
- Create a dictionary of strings and look up values by key.

Process Skill Goals

During the activity, students should make progress toward:

- Making predictions about the behavior of container methods. (Information Processing)

Model 1   Indexes and Values

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

```
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 value's sequential position starting at 0. For example, `primes[4]` is the value 11.

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

Questions (15 min)   **Start time:**

1. Consider the second element of `primes`.

   a) What is the index?   b) What is the value?

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. If an error occurs, write the type of error.

Python code	Shell output
`odd = [1, 3, 5, 7]`	
`odd`	
`odd[2]`	
`odd[4]`	
`len(odd)`	
`number = odd[1]`	
`number`	
`odd[1] = 2`	
`odd`	
`number`	

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. One of the lines in the table 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`.

Presenter: Write your team's answers for the last three questions, in large print, on a blank sheet of paper. When asked, hold up your answers to the entire class. Be ready to explain your team's answers.

Reflector: Take one minute to give feedback to your team. Describe one thing that the team is doing well, and one thing the team can be doing better.

Model 2   School Acronyms

Venn Diagram:
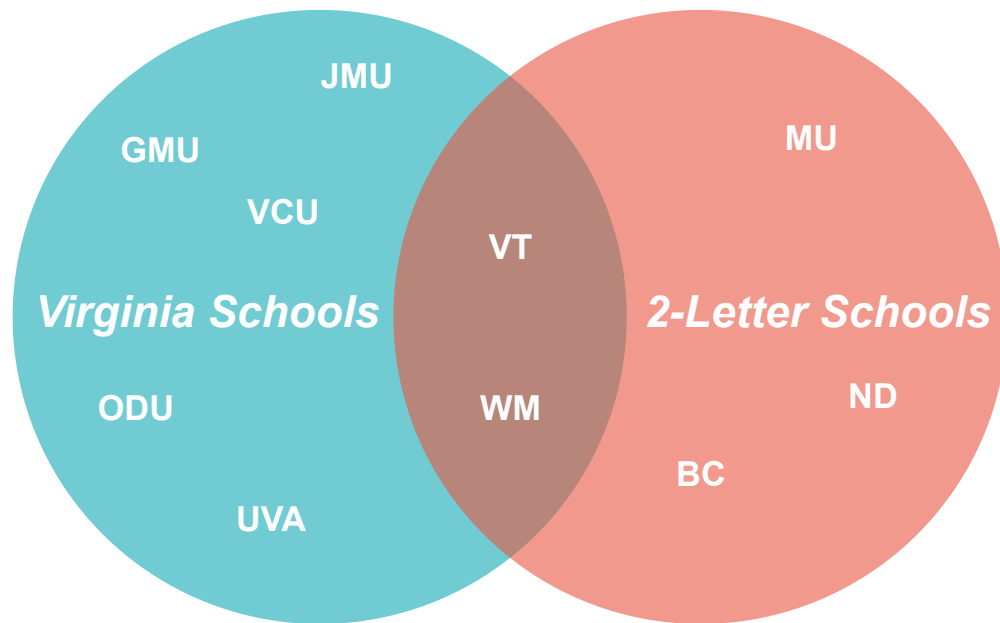
Python Code:

```python
va_schools = {"JMU", "GMU", "VCU", "VT", "ODU", "WM", "UVA"}
two_letter = {"MU", "VT", "ND", "WM", "BC"}
```

Questions (15 min)   Start time:

10. As a team, predict the value of each expression. Then run each expression in a Python Shell, and record the actual value. Discuss any differences between the predicted and actual values.

Python code	Predicted value	Actual value
`type(va_schools)`		
`len(two_letter)`		
`"ND" in va_schools`		
`"JMU" not in two_letter`		
`len(va_schools.union(two_letter))`		
`va_schools.intersection(two_letter)`		
`two_letter.difference(va_schools)`		

11. Identify the following operations used in the previous question.

    a) Built-in functions:

    b) Built-in operators:

    c) Methods of a `set`:

12. Predict the result of the following code. Then run the code in a Python Shell, and compare with your prediction.

```python
va_schools.remove("GMU")
va_schools.remove("ODU")
va_schools.remove("VCU")
print(va_schools)
```

13. Predict the result of the following code, based on the result of the previous question. Then run the code in a Python Shell, and compare with your prediction.

```python
va_schools.add("JMU")
va_schools.add("JMU")
va_schools.add("JMU")
print(va_schools)
```

14. Summarize the properties of a `set` object. For each answer, explain why.

    a) Is a `set` mutable or immutable?

    b) Can a `set` contain duplicate values?

    c) Is the order of `set` values predictable?

15. Identify differences between a `set` and a `list` in Python.

Model 3 Keys and Values

In Python, a *dictionary* stores `key: value` pairs. In the following example, the `key: value` pairs are separated by commas and wrapped in curly braces.

```python
elements = {'C': 'carbon', 'H': 'hydrogen', 'O': 'oxygen', 'N': 'nitrogen'}
```

Key	Value
'C'	'carbon'
'H'	'hydrogen'
'O'	'oxygen'
'N'	'nitrogen'

In contrast to lists and tuples, a dictionary is a *mapping* type. Values are referenced by *keys*, rather than by integer indexes.

Python code	Shell output
`type(elements)`	<class 'dict'>
`elements.keys()`	dict_keys(['C', 'H', 'O', 'N'])
`elements.values()`	dict_values(['carbon', 'hydrogen', 'oxygen', 'nitrogen'])
`elements['C']`	'carbon'
`atom = 'N'`	
`elements[atom]`	'nitrogen'
`elements[N]`	NameError: name 'N' is not defined
`elements['nitrogen']`	KeyError: 'nitrogen'
`elements[1]`	KeyError: 1
`len(elements)`	4
`elements['B'] = 'boron'`	
`elements.items()`	dict_items([('C', 'carbon'), ('H', 'hydrogen'), ...])

Questions (15 min)   Start time:

16. What is the data type of the keys in the `elements` dictionary?

17. List all the keys stored in the `elements` dictionary at the end of the table.

18. Explain the reason for the error after entering each of the following lines:

  a) `elements[N]`

  b) `elements['nitrogen']`

  c) `elements[1]`

19. Ignoring the "dict_items()" part, describe the contents and type of data returned by the `items()` method.

20. Write a Python expression that creates a dictionary for the seven days of the week: Sun=1, Mon=2, Tue=3, etc. Assign the dictionary to the variable `dow`.

21. If you assign two different values to the same key (i.e., two assignment statements with one value each), which value is stored in the dictionary? Justify your answer with an example.

22. Another way to store the data in Model 3 is to use two lists:

```
keys = ['C', 'H', 'O', 'N']
vals = ['carbon', 'hydrogen', 'oxygen', 'nitrogen']
```

What is a disadvantage of this approach? Explain your reasoning.