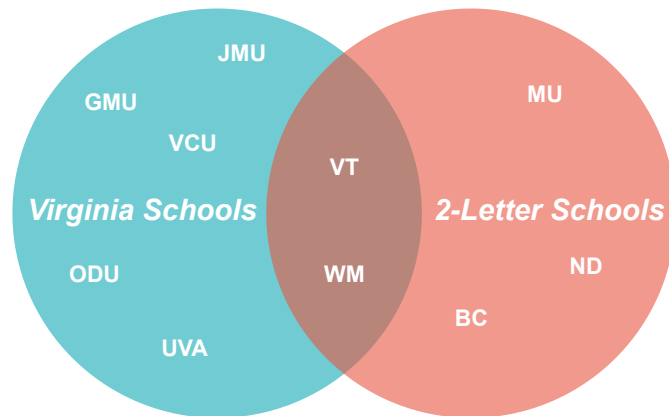


Sets and Dictionaries

A set is a group of things of the same kind. A dictionary is a set of key-value pairs. Sets and dictionaries are useful in programming, because they can store and look up unique values.

Unit 1 School Acronyms

Venn Diagram:



Python Code:

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

Questions

1. As a team, predict the value of each expression. Type the two lines of Python code in the Python shell, 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
<code>type(va_schools)</code>		
<code>len(two_letter)</code>		
<code>"ND" in va_schools</code>		
<code>"JMU" not in two_letter</code>		
<code>len(va_schools.union(two_letter))</code>		
<code>va_schools.intersection(two_letter)</code>		
<code>two_letter.difference(va_schools)</code>		

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

a) Built-in functions:

b) Built-in operators:

c) Methods of a `set`:

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

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

4. 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.

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

5. 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?

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

Unit 2 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.

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

Questions

7. What is the data type of the keys in the elements dictionary?
8. List all the keys stored in the elements dictionary at the end of the table.

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

a) `elements[N]`

b) `elements['nitrogen']`

c) `elements[1]`

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

11. 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`.

12. 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.

13. Another way to store the data in ?? 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.