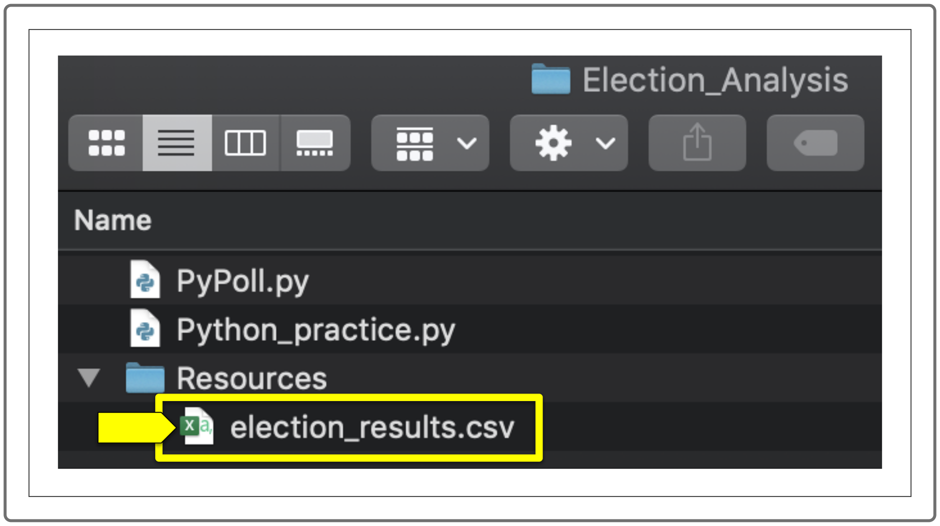
3.4.1 Python Dependencies, Modules, and Packages

After inspecting the dataset using Excel, you tell Tom and Seth that you are ready to open the file and begin the election audit. Seth explains to you that before you open the CSV, you need to make a connection to the file by using your computer's directory "path" to that file. Once you connect to your file, you will be able to access the contents of the file. However, you need to use a programming tool for this specific purpose. Tom will show you how to connect to the file and then walk you through how to use some programming tools that will give you access to the contents in the CSV file.

For data analysis, the process of opening and reading files is important. All programming languages have a set of methods that will open a file and read the file. The programming language you are using requires very precise directions for the path to the file.

The `election_results.csv` file should be located in the Resources folder, as shown in the following image.



If we were writing code in `PyPoll.py`, the path to the CSV can be written `Resources/election_results.csv`. If `election_results.csv` was not in the Resources folder but at the same "level" as `PyPoll.py`, then the path would be `election_results.csv`.

IMPORTANT

> Different operating systems use different path separators to separate files and folders.
>
> macOS: Use a forward slash to separate folders and files. "/"
>
> Windows: Uses backslashes to separate folders and files. "\"

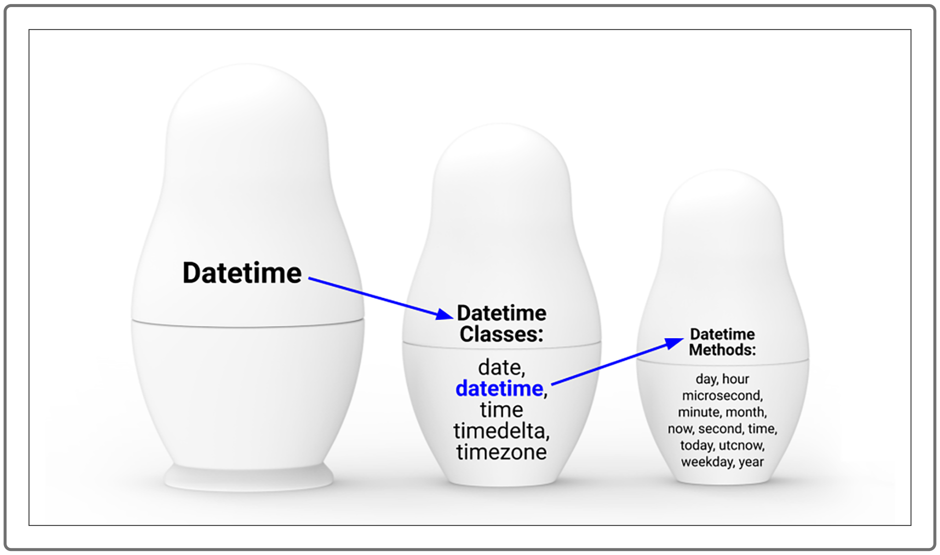
Now that we know how to connect to our `election_results.csv` file, we need to read, process, and parse, or analyze, the data in the file. To do this will require us to use programming tools like you would use hardware tools in a toolbox. These tools, in programming parlance, are called dependencies, packages, and modules.

Dependencies

Dependencies are modules and packages, or a programming script that someone else has written, that allows you to increase the functional programming of your code, or speed and efficiency.

Python is an open-source language, which means that it is easy for others to write dependencies that can be used with Python. You can think of the relationship between dependencies, modules, and packages like Russian nesting dolls.

Dependencies are the largest "doll," like the Python **datetime module**. Inside the datetime module are functions, classes, or variables, which are the second-largest doll. The methods used for functions and classes are in the third-largest doll.


Datetime
Datetime Classes:
date, datetime, time timedelta, timezone
Datetime Methods:
day, hour microsecond, minute, month, now, second, time, today, utcnow, weekday, year

This is how we would use the datetime module to get today's date using VS Code.

```
# Import the datetime class from the datetime module.
import datetime
# Use the now() attribute on the datetime class to get the present time.
now = datetime.datetime.now()
# Print the present time.
print("The time right now is ", now)
```

Let's break down what is happening in the code above as it relates to the dependency "datetime". The datetime module comes with our Python installation.

1. To use the `datetime` module all we need to do is to import it using `import datetime`.
2. In line 4, we declare the `now` variable to hold the time right "now".
3. The `now` variable is set equal to `datetime.datetime.now()`, where:
    - The first `datetime` is the datetime module, (first doll).

- The second `datetime` is the datetime class (second doll).
- Then we use the datetime attribute, `now()`, (third doll) on the datetime class, i.e., `datetime.now()`, to get the current time.

When we run this code, the output will be the current time at the moment the code is run, and will look similar to the following:

```
The time right now is 2019-09-18 14:11:42.394131
```

To reduce the confusion on importing a module with the same name as a class we can use an abbreviated alias `dt` for the datetime module.

```
# Import the datetime class from the datetime module.
import datetime as dt
# Use the now() attribute on the datetime class to get the present time.
now = dt.datetime.now()
# Print the present time.
print("The time right now is ", now)
```

Packages

Packages are folders that contain a set of Python modules. The folders in the packages may contain various subpackages, or other folders. To import packages, we use the `import` statement, as we did with the datetime module.

Modules

Modules are a separate software component. They are usually Python files with a `.py` extension. The name of the module will be the name of the file. A Python module can have functions, classes, or variables defined and implemented.

Modules can be used in a variety of applications and functions with other programs. They may contain hundreds or thousands of lines of code, so it would be foolish to write or repurpose the code every time you need to use it. This type of programming can lead to many syntax or logical errors in the program that would require an enormous amount of time to correct.

Modules are easy to use and maintain, and they provide reusability with a simple statement like `import datetime`. To use a specific function, class, or variable from a module, you use a statement like `from import`.

If your script requires the use of programs, modules, and packages, one of the first steps is to import dependencies for your Python script.

Let's put this concept to good use and walk through how to read a CSV file by using the CSV module.

The CSV Module

In Python there's a built-in module called `csv`, which allows users to easily pull in data from external CSV files and perform operations on them.

The `csv` module is imported by using the `import` statement followed by the module name, `csv`.

The `csv` module has many functions that allow us to read and write tabular data in CSV format. With the `csv` module, we can read data from a file that was generated by Excel and write data to a file in a format that can be read by Excel.

To see all the functions available in the `csv` module, follow these steps:

1. Launch the Python interpreter.
2. Type `import csv` to import the module.
3. Press Enter.
4. Type `dir(csv)`. The "dir" is short for "directory".

The Python interpreter should look like this:

```
>>> import csv
>>> dir(csv)
```

Press Enter. The output will look like this:

```
['Dialect', 'DictReader', 'DictWriter', 'Error', 'OrderedDict', 'QUOTE_ALL',
```

If you look closely at the output, you'll see a function called `reader`. We'll use this function to read the CSV file that contains the election data.

Using the `dir()` function, we can pass:

1. A Python module, like the `csv` module. The `dir()` function will return all the functions available in the `csv` module.
2. A variable, like a dictionary `{'key':'value'}`, for example the `counties_dict` dictionary. The `dir()` function will return all the functions available on that variable.

```
>>> dir({'Arapahoe': 422829, 'Denver': 463353, 'Jefferson': 432438})

['__class__', '__contains__', '__delattr__', '__delitem__', '__dir__', '
```

3. A data type, like `str`. The `dir()` function will return all the attributes and methods that can be used with the `str` data type.

```
>>> dir(str) '__add__', '__class__', '__contains__', '__delattr__', '__d
```

SKILL DRILL

For the following data types and data structures, use the `dir()` function to find all the attributes and methods.

1. `int`
2. `float`
3. `bool`
4. `list`
5. `tuple`
6. `dict`
7. `datetime`

Here are some other modules that may be useful later. You will have to import these first.

- `random`

- `numpy`