Tutorial Objectives

- Practice writing code using **strings, lists, and dictionaries**, as well as the other control structures we have used previously

In this tutorial we will be building an interactive story (like one of those "Choose your own adventure" novels). The story will be built out of a sequence of scenes stored in a somewhat tricky data structure. In part 1 you'll learn about the structure of each individual scene. In part 2, you'll build a full story of scenes into one large structure of data. In part 3, you'll display a single scene to the screen. And in part 4, finally you'll build the full story-telling process. To start, create a file called story.py.

**Part 1 – Setting the Scene**

Each scene in this story is a block of text to read as well as several pieces of information required to allow some user choices. Each scene will be made up of the following pieces of information:

- A short label to refer to the scene.
- The text of the scene.
- The text for each possible choice at the end of the scene.
- The labels for the next scene for each choice.

The label lets us refer to the scene. The text lets the user read the scene. The choices at the end of the scene provide the user with options after reading the scene text. And the labels for the next scenes will help us move from scene to scene with each user choice.

**As an example:**

- Scene label: ‘Approach’
- Scene text: "Following the map from the old man in the tavern, you arrive at a large hill, covered with ancient standing stones forming the shape of a skull if viewed from a high vantage point."
- Choices: "Enter the Tomb of Horrors", "Run Away!"
- Next scenes: ‘Entrance’, ‘Runaway’
To hold all of this information we'll need a data structure. This information is in no particular order, and the descriptions for each piece of information will make good keys for a dictionary.

Build a dictionary `scene = { ... }` for a single scene with values for the following keys:

- `sceneText`
- `choices`
- `nextScene`

To build the scene, you can use the text from the example above or write your own scene. Note that the values for `choices` and `nextScene` are multiple strings, so you'll need to use lists for these. The Scene Label will be used in Part 2, and is not included in the scene dictionary.

It may be helpful to know that you can use a backslash to spread a line of code over multiple lines. E.g.:

```python
scene = { 'sceneText': "...", \
        'choices': [...], \
        'nextScene': [...]}
```

When complete, test that you are able to access the individual elements of the scene. **Note**: to access the individual elements from choices or nextScene you will need to provide TWO sets of square brackets. One to access the list from inside the scene dictionary, and one access an element from that list.

```python
print(scene['sceneText'])
print(scene['choices'][0])
```

**Part 2 – A Story Worth Telling**

Once you're clear on the structure of a single scene its time to write a whole story. A story will be made up of several scenes, so we'll need a data structure to hold all of the scenes together. Since our story will be non-linear (thanks to the user's choices), a list is not an ideal data structure for this problem. A dictionary on the other hand is non-sequential, and we have scene labels (see Part 1) in each scene that can act as keys.

Make an empty dictionary called `story = {}`, that will hold all of the scenes of our story (including the scene we built in Part 1). Each key in the `story` dictionary will
be the label for a scene. Each value in the story dictionary will be the details for a scene (which we defined as a dictionary in Part 1). That is, story will be a dictionary of dictionaries.

Build the story dictionary with a complete story of scenes. To do this you will need to create several scenes like the one you built in Part 1. For each scene you should add it to the story dictionary with a unique label.

E.g. story['Approach'] = <dictionary from part 1>
    story['Entrance'] = {...}

A path through the story starts at the 'Approach' scene and follows each of the user's choices to the next selected scene. For example, if the user chooses to "Enter the Tomb of Horrors" (which is the first choice in choices), then the next scene will be the one labelled "Entrance" (which is the first label in nextScene). Scenes that have no choices are end points in the story.

You can write your own story, or if you're not feeling creative today, you can use the text from this story. You will still need to translate it into the scene structure in your code.

Part 3 – Presenting a Scene

With the story dictionary built its time to start telling your story. Let's start by displaying a single scene. Create a main() function to put your story telling code in. Start with a variable to hold the current scene's data. This will make the code we have to write simpler by not having to access the story dictionary every time.

    sceneData = story['Approach']

With that in place its time to display the data in the scene. Start by printing the scene's text, so the user can read this step of the story. Next, if the scene has any choices, display each of the choices to the user, and then prompt them to select one.

Note, you will need to provide simple labels (e.g. numbers) so that the user can easily select an option. We will handle the user's selection in Part 4. For now, ensure that you can display scenes that have choices as well as scenes that don't. Scenes without choices should simply have the scene text printed, with no user prompt given.

Part 4 – Telling a story

This part describes the mechanics of making the story telling process going from scene to scene according to the user's choices.
The story telling process will be a loop that does the following:

1. Display the current scene
2. Get the user's choice
3. Set the current scene to the chosen next scene

This loop should repeat until the story reaches a scene with no choices.

To start, create a variable at the top of your main() function called `currentScene`, with the value "Approach" (or whatever your first scene is labelled). This variable will keep track of which scene in the story we're currently on.

Next, change the `sceneData` variable from Part 3 to load the data of `story[currentScene]` instead of 'Approach'. Test that this lets you change the scene by changing the label inside `currentScene` (manually for now).

Next, wrap the code you wrote in Part 3 to display a single scene, inside a loop. This loop should continue to repeat until the `currentScene` variable is an empty string ("").

Lastly, we need to handle the user's choices to change scenes. For this tutorial, you may assume the user enters good input. (For an extra challenge, don't assume good input.) Once the user has made their choice, set the value of `currentScene` to the appropriate label from `sceneData['nextScene']`. E.g., if the user selected the first choice, set `currentScene` to the first label in the list.

In the event that the scene you're displaying does not have any choices, set `currentScene = ""`. This should end your loop, and complete your story.

-The End.-