DSE5002 MODULE 5 LAB

Peter Gyorda April 19, 2025

From Think Python, Chapter 10,

10.11.2, 10.11.4

In [3]:
```
10.11.2. Exercise
Use get to write a more concise version of value_counts. You should be able to elim
```

In [3]:
```
def value_counts_concise(s):
    counts = {}
    for char in s:
        counts[char] = counts.get(char, 0) + 1
    return counts

# Let's test it out
result = value_counts_concise('brontosaurus')
print(result)
```

```
{'b': 1, 'r': 2, 'o': 2, 'n': 1, 't': 1, 's': 2, 'a': 1, 'u': 2}
```

In [ ]:
```
Exercise 10.11.4
10.11.4. Exercise
Write function called find_repeats that takes a dictionary that maps from each key
```

In [5]:
```
def find_repeats(counter):
    """Makes a list of keys with values greater than 1.
        counter: dictionary that maps from keys to counts
        returns: list of keys
    """
    repeated_keys = []
    for key, count in counter.items():
        if count > 1:
            repeated_keys.append(key)
    return repeated_keys

# Let's test it with the output from our value_counts function
word_counts = value_counts_concise('brontosaurus')
repeated = find_repeats(word_counts)
print(f"Counts of each letter: {word_counts}")
print(f"Letters that appear more than once: {repeated}")
```

```
Counts of each letter: {'b': 1, 'r': 2, 'o': 2, 'n': 1, 't': 1, 's': 2, 'a': 1, 'u':
2}
Letters that appear more than once: ['r', 'o', 's', 'u']
```

In [ ]: