Lab01: Revision 2

- Indentation
- File IO
- Lists
- Dictionaries
- List Comprehension

Indentation

- You can NOT mix indentation levels within the same block! The above code will result in a syntax error!


```
hard_working = 1
careless = 0

        print('Hello World')
        print('Can you fix the error?')
        print('You must never mix spaces and tabs.')
if hard_working and not careless:
    print("You should get a good grade")
elif hard_working and careless:
        print("You may be able to pass this course")
else:
    print("You may need to try again in 2020 S1")
```

- If you're using IDLE, you can use Ctrl+] to indent and Ctrl+[to unindent.

File Reading

```
hello world

life is a long journey


well done
```

- When should we use file.read() and file.readlines()?
  - file.read() reads the file as an individual string.
    - Use .split() to separate all words
  - file.readlines() reads the file by line by line iteration. It returns a list of strings. Note: remember to remove the '\n' character

```
print(input_file.read()._____())
```

```
['hello', 'world',
'life', 'is', 'a',
'long', 'journey',
'well', 'done']
```

```
print(input_file.readlines())
```

```
['hello world\n', '\n', 'life is a long
journey\n', '\n', '\n', 'well done\n',
'\n']
```

```
['hello world', '', 'life is a long
journey', '', '', 'well done', '']
```

Dictionaries

- Python's dictionaries allow you to connect pieces of related information. Each piece of information in a dictionary is stored as a key-value pair. When you provide a key, Python returns the value associated with that key. You can loop through all the key-value pairs, all the keys, or all the values.
- Loop through …

```
for name in languages.keys():
  print(name)
```

```
for name, language in languages.items():
  print(name + ": " + language)
```

- Print in sorted/reverse keys order

```
for name in sorted(languages.keys()):
  print(name + ": " + languages[name])
```

```
for name in sorted(languages.keys(), reverse=          ):
  ...
```

List comprehension

- Very Useful
- Basic:
- With if/else

```
[expression for variable in sequence]
```

```
[expression for variable in sequence if condition]
```

```
squares = [x**2 for x in range(1, 6)]
```

```
[1, 4, 9, 16, 25]
```

SAME

```
names = ['kai', 'abe', 'ada', 'gus', 'zoe']
upper_names = []
for name in names:
    upper_names.append(name.upper())
```

```
upper_names = [name.upper() for name in names]
```

```
['kai', 'abe', 'ada']
```

```
names_with_a = [name for name in names if 'a' in name]
```

Tips for coderunner

- Q5: List comprehension will be useful here
- Q6: To check if a letter is a consonant you can use:
  letter not in "AEIOU"
- Q7: my_dictionary.values() will get all of the values in a dictionary
- Q9: The split() method will be helpful here. you will have to use it twice, once to split the file by newlines and once to split the two words appart.