Week 3

Dates	April 11
Topics	CH3

Tuple

- A tuple is like a list that uses parentheses

```
>>> fibs = (0, 1, 1, 2, 3)
>>> print(fibs[3])
2
```

- The main difference between a tuple and a list is that a tuple cannot change once you've created it

Replace

```
>>> fibs[0] = 4
Traceback (most recent call last):
  File "<pyshell>", line 1, in <module>
    fibs[0] = 4
TypeError: 'tuple' object does not support item assignment
```

- Use tuple if you know can never change

Map

- A map or dictionary is a collection of things, like lists and tuples.

```
>>> favorite_sports = ['Ralph Williams, Football',
                       'Michael Tippett, Basketball',
                       'Edward Elgar, Baseball',
                       'Rebecca Clarke, Netball',
                       'Ethel Smyth, Badminton',
                       'Frank Bridge, Rugby']
```

If I asked you what Rebecca Clarke's favorite sport is, you could skim through that list and find the answer is netball. But what if the list included 100 (or many more) people?

```
>>> favorite_sports = {'Ralph Williams' : 'Football',
                       'Michael Tippett' : 'Basketball',
                       'Edward Elgar' : 'Baseball',
                       'Rebecca Clarke' : 'Netball',
                       'Ethel Smyth' : 'Badminton',
                       'Frank Bridge' : 'Rugby'}
```

- The difference between maps and lists or tuples is that each item in a map has a key and a
  corresponding value

Table 3-1: Keys Pointing to Values in a Map of Favorite Sports

Key	Value
Ralph Williams	Football
Michael Tippett	Basketball
Edward Elgar	Baseball
Rebecca Clarke	Netball
Ethel Smyth	Badminton
Frank Bridge	Rugby

- This is a table representation of the favorite_sports

Access

```
>>> print(favorite_sports['Rebecca Clarke'])
Netball
```

- We access our map favorite_sports using her name as the key

> 💡 Did you realize the index is different in a map?

Delete

- To delete a value in a map, use its key

```
>>> del favorite_sports['Ethel Smyth']
>>> print(favorite_sports)
{'Rebecca Clarke': 'Netball', 'Michael Tippett': 'Basketball', 'Ralph
Williams': 'Football', 'Edward Elgar': 'Baseball', 'Frank Bridge':
'Rugby'}
```

Replace

- To replace a value in a map, we also use its key

```
>>> favorite_sports['Ralph Williams'] = 'Ice Hockey'
>>> print(favorite_sports)
{'Rebecca Clarke': 'Netball', 'Michael Tippett': 'Basketball', 'Ralph
Williams': 'Ice Hockey', 'Edward Elgar': 'Baseball', 'Frank Bridge':
'Rugby'}
```

Join

- You can't join maps with the plus operator (+) like we did in list

```
>>> favorite_sports = {'Rebecca Clarke': 'Netball',
                       'Michael Tippett': 'Basketball',
                       'Ralph Williams': 'Ice Hockey',
                       'Edward Elgar': 'Baseball',
                       'Frank Bridge': 'Rugby'}
>>> favorite_colors = {'Malcolm Warner' : 'Pink polka dots',
                       'James Baxter' : 'Orange stripes',
                       'Sue Lee' : 'Purple paisley'}
>>> favorite_sports + favorite_colors
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for +: 'dict' and 'dict'
```

Challenge 1

Create a list and map that contain this week's date and day

Ex. '4.12 : Friday'

If Statements

- An if statement might be written in Python like this

```
>>> age = 13
>>> if age > 20:
        print('You are too old!')
```

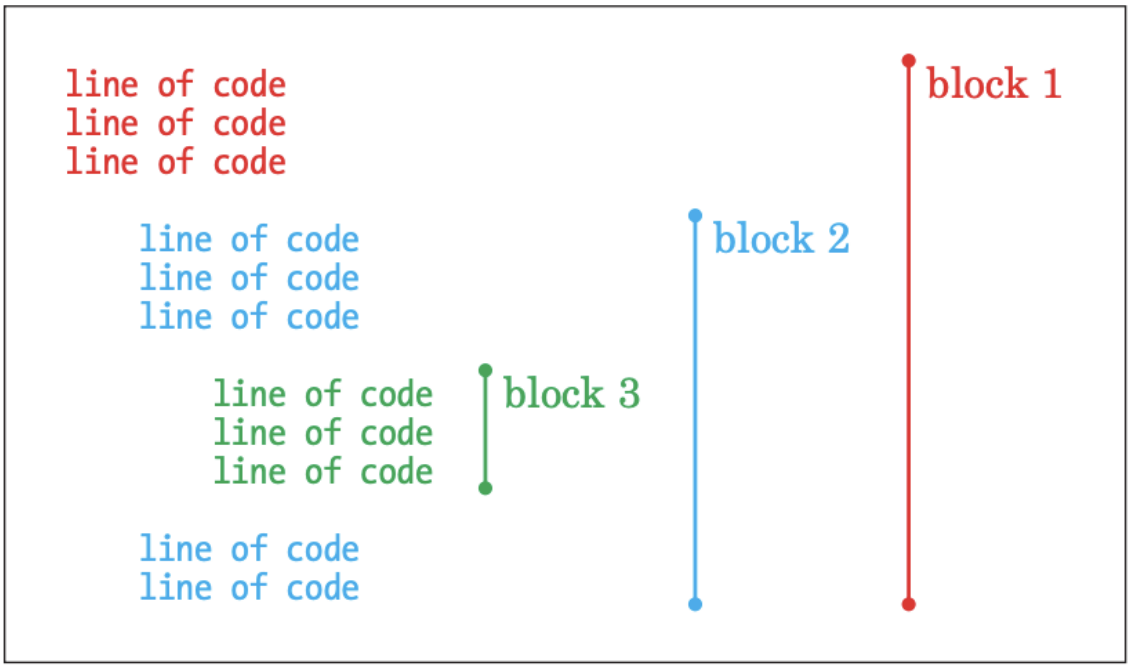
- An if statement is made up of the if keyword, followed by a condition and a colon (:)

Block

- A block of code is a grouped set of programming statements
- For example, when if age > 20: is true, you might want to do more

```
>>> age = 25
>>> if age > 20:
        print('You are too old!')
        print('Why are you here?')
        print('Why aren\'t you mowing a lawn or sorting papers?')
```

- Whitespace, such as a tab (inserted when you press the tab key) or a space (inserted when you press the spacebar), is meaningful.



Quiz1

Will it work?

```
>>> if age > 20:
    □□□□print('You are too old!')
    □□□□□□print('Why are you here?')
```

▼ Answer

No! It is not part of a block. Use consistent spacing!

```
>>> age = 25
>>> if age > 20:
        print('You are too old!')
        ■print('Why are you here?')
SyntaxError: unexpected indent
```

Conditions

Symbol	Definition
==	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

- Be sure to use a double equal sign (==) when defining an equal-to condition
- equal sign (=) is an assignment operator!

```
>>> age = 10
>>> if age == 10:
        print('What\'s brown and sticky? A stick!!')
```

If and Else

- “If something is true, then do this”

```
>>> age = 8
>>> if age == 12:
        print("A pig fell in the mud!")
else:
        print("Shh. It's a secret.")
```

If and Elif

```
>>> age = 12
>>> if age == 10:
        print("What do you call an unhappy cranberry?")
        print("A blueberry!")


elif age == 11:
        print("What did the green grape say to the blue grape?")
        print("Breathe! Breathe!")
elif age == 12:
        print("What did 0 say to 8?")
        print("Hi guys!")
elif age == 13:
        print("Why wasn't 10 afraid of 7?")
        print("Because rather than eating 9, 7 8 pi.")
else:
        print("Huh?")
```

- Elif means else if
- You can add more conditions using elif

Challenge 2

Suppose we need to assign different grades to students based on their scores.

1. If a student scores above **90**, assign grade **A**
2. If a student scores above **75**, assign grade **B**
3. If a student scores above **65**, assign grade **C**
4. Below, assign grade **D**