

Today's agenda.

1. Python data collection structure: **list**, **tuple**, **dictionary**, etc.
2. Use of **loop** in handling data collection.
3. What is a '**function**'? Why is it so important to programming?
4. Use of **module** in building program libraries for reusability.
5. Introducing the **Pandas** program library for data analysis and visualisation.

Exercises for Quick Recap

- Addition +
- Subtraction -
- Multiplication *
- Division /
- Rounded division //
- Exponent **
- Modulus % (division returns remainder)

$$x = 17, y = -21, z = 5$$

1) Print $5x(7+y) = -1190$

2) Print $-3y - \left(\frac{x+23}{2z}\right) = 59$

Implement the above equation using Python codes.

```
s = "Chen Zi"  
v1 = "studies"  
v2 = "now lives"  
p = "atin"  
o = "CUHK"  
b = " "
```

- 1) Print "Chen Zi studies at CUHK."
- 2) Print "Chen Zi now lives in HK."

Implement the above equation using Python codes.

Useful String Functions

1. `type(), str(), int(), float()`
2. `len()`
3. `name.upper()`
4. `name.replace(destination,source)`
5. `name.find(destination)`
6. `name.split()`
7. `name.count(target)`

```
y = "cuhk"  
z = "abc"  
z = z.replace(z,y)  
position = z.find("c")  
sentence = "I am a CUHK student."
```

- 1) Print z
- 2) Print position
- 3) Print number of blank characters in the variable "sentence".
- 4) Print number of characters in the variable "sentence".

Input Function

```
print('Enter your name:')  
x = input()  
print('Hello, ' + x)
```

- 1) Print 'Input a sentence'
- 2) Input the value
- 3) Print 'Input a string within the sentence.'
- 4) Print the starting position of the string within the sentence.

Conditional & Loop

1. == (equal)
2. != (not equal)
3. >= (greater than and equal)
4. <= (less than and equal)
5. not
6. if ... elif ... else

```
a = 5
b = 2
if a > b:
    print("a is greater than b")
else:
    print("a is not greater than b")
```

Conditional

```
string = 'This is a string.'
for i in string:
    print(i)
```

Loop

- 1) Enter “Age”
- 2) Enter “Gender”
- 3) Use conditional statement to test entries to see if the age is greater than 18 and gender is “male”.
- 4) Enter a number and determine whether it is greater than 100 or not.
*** Hint: Use number conversation to apply computation.

LISTS AND DICTIONARIES

```
students = ["Jane","Mary","Esther"]  
print(students)  
print(students[2])  
students[1] = "John"  
print(students)  
print(len(students))  
misc = [23,"Bernard",True]  
print(misc)
```

```
students = {  
    "Mei Li": 80,  
    "John Chen": 90  
}  
student1 = {  
    "name": "Mei Li",  
    "major": "English"  
}  
student2 = {  
    "name": "John Chen", "major": "Physics"  
}
```

**A dictionary is a
collection of Value Pairs**
"Key": "Value"

Useful List Functions

```
list_name.append(value)
list_name.insert(pos, value)
list_name.pop()
sorted(list_name, reverse=True/False))
```

```
list1 = [1,2,3]
list1.append(4)
print(list1)
list1.insert(0, 5)
print(list1)
list1.pop()
print(sorted(list1,reverse=True))
```

Useful Dictionary Functions

```
print(students)
print(student1, len(student1))
print(student2, len(student2))
print(student1.keys())
print(student1.values())
print(students["John Chen"])
print(student1["name"], student1["major"])
for key in students:          #Loop through a
    print(key, students[key]) # dictionary to print
```

Add Dictionary Item

```
dict = {'Peter':80,"David":90,"Mary":100}  
dict['John'] = 30 #Add {"John",30} to dict or  
dict.update({'John':30})
```

Remove Dictionary Item

```
del dict['John']
```

Insert Dictionary Item

```
dict.insert(dict.index('Peter')+1, 'William')  
dict['William'] = 65
```


Looping through **List** and **Dictionary**

```
list = [19,21,32,45,57]
```

```
for i in list:
```

```
    print(i)
```

```
dict = {'John':70,'David': 60, 'Peter': 80}
```

```
for i in dict:
```

```
    print(k, dict[k])
```

Combining List & Dictionary

```
student1 = {  
    'id': '1111',  
    'name': 'David Chan',  
    'email': 'david@cuhk.edu.hk'  
}  
student2 = {  
    'id': '1112',  
    'name': 'Peter Lee',  
    'email': 'peter@cuhk.edu.hk'  
}  
student3 = {  
    'id': '1113',  
    'name': 'Mary Fung',  
    'email': 'mary@cuhk.edu.hk'  
}
```

```
students = []  
students.append(student1)  
students.append(student2)  
students.append(student3)  
print(students)  
for i in students:  
    print(i)  
key = 0  
for i in students:  
    print("Individual entry:", i)  
    for key in i:  
        print("Key:", key, "Value:", i[key])
```

Python Function

```
def name_of_function(parameter1, parameter2...):
```

[Python variable assignment statements,
mathematical and logical operations]

```
    return result
```

Example

```
def calc_avg(x,y):  
    result = x + y  
    return result
```

```
print("Result = ", calc_avg(10,20))
```

Output

Result=30

Exercise

Formula for converting Celcius into Fahrenheit: $(23^{\circ}\text{C} \times 9/5) + 32 = 73.4^{\circ}\text{F}$.

Create a list of temperature in Fahrenheit for the week based on the temperature of Celcius but implement the codes with Python functions.¶

Sun	Mon	Tue	Wed	Thu	Fri	Sat
23°C	24°C	28°C	19°C	17°C	21°C	25°C

Python Module

```
from convert_temp import *  
total_temp = 0  
total_temp = get_daily_celcius()  
display_avg(total_temp)
```

```
def convert_temp(temp):
```

```
    f_value = 0
```

```
    f_value = (float(temp) * 9/5) + 32
```

```
    return f_value
```

```
def compute_total_temp(temp, total_temp):
```

```
    total_temp = total_temp + float(temp)
```

```
    return total_temp
```

```
def get_daily_celcius():
```

```
    total_temp = 0
```

```
    week_of_day = ['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat']
```

```
    for i in week_of_day:
```

```
        temp = input("Enter " + i + "'s temperature in celcius:")
```

```
        f_temp = convert_temp(temp)
```

```
        print(i + "'s temperature in Fahrenheit:" + str(f_temp))
```

```
        total_temp = compute_total_temp(f_temp, total_temp)
```

```
    return total_temp
```

```
def display_avg(total_temp):
```

```
    avg_temp = round((total_temp/7),2)
```

```
    print("\n" + "Average temperature for the week:" + str(avg_temp))
```

```
    return
```

The Pandas Library

Save a Pandas Data Frame into CSV file

```
import pandas as pd
```

```
scores = [  
    {'Student': 'David Chan', 'Jan': 90, 'Feb': 85, 'Mar': 88},  
    {'Student': 'Peter Lee', 'Jan': 72, 'Feb': 75, 'Mar': 68},  
    {'Student': 'John Lui', 'Jan': 60, 'Feb': 80, 'Mar': 100 }]  
df = pd.DataFrame(scores)  
df.to_csv('weekly_hours.csv',mode='w',index=False)  
df  
list = df.to_dict('records')  
for entry in list:  
    print(entry['Student'],entry['Jan'],entry['Feb'],entry['Mar'])
```

```
import pandas as pd
```

```
scores = [
```

```
{ 'scores': 'David Chan', 'Jan': 90, 'Feb': 85, 'Mar': 88},
```

```
    { 'scores': 'Peter Lee', 'Jan': 72, 'Feb': 75, 'Mar': 68},
```

```
    { 'scores': 'John Wong', 'Jan': 60, 'Feb': 80, 'Mar': 2000 }]
```

```
df = pd.DataFrame(scores)
```

```
df
```

```
df.to_csv('students.csv',mode='w',index=False)
```

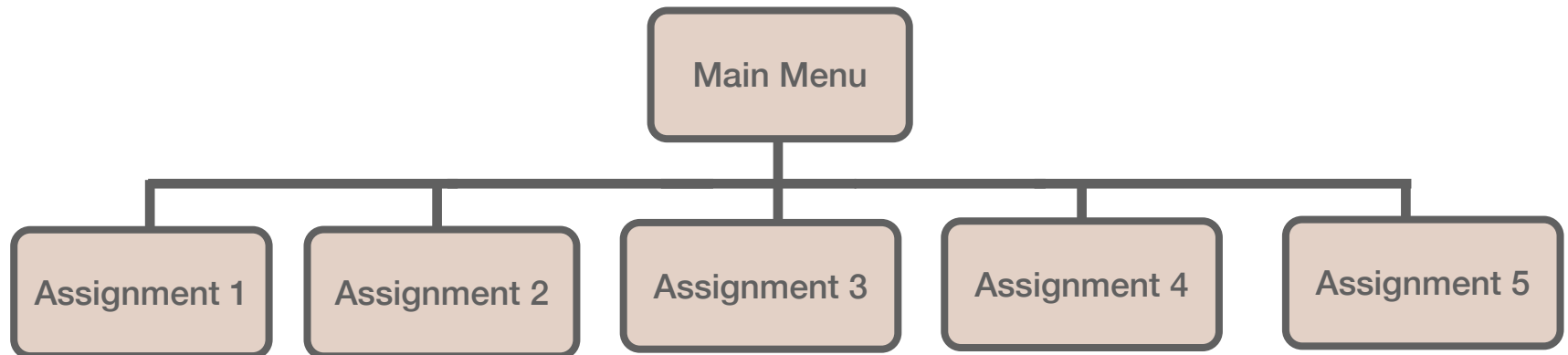
```
df
```

Read a CSV file into a Pandas Data Frame

```
import pandas as pd  
df = pd.read_csv("students.csv")  
df
```


Assignment #1:

Create a website with the follow structure and pages and upload to GitHub:



Thank you for your time!