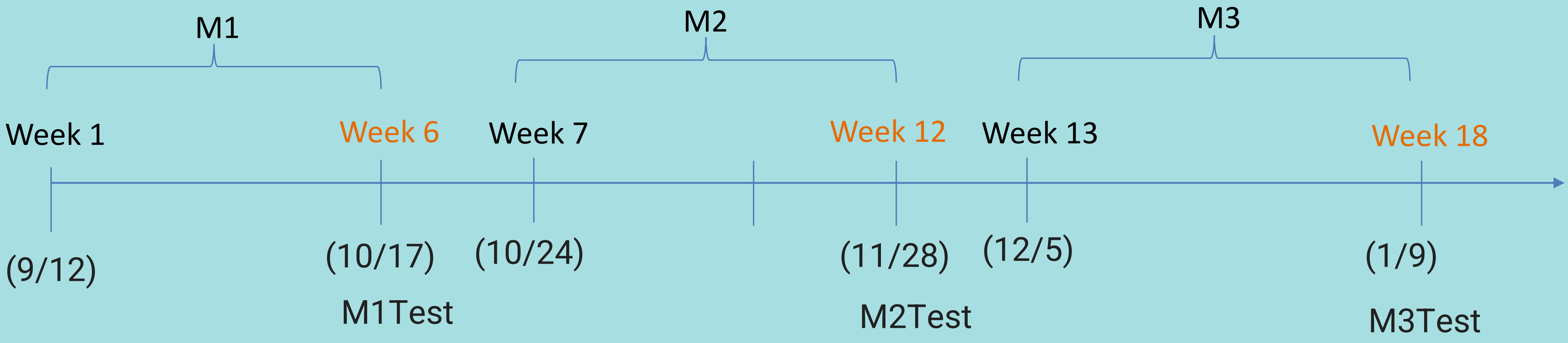
Fundamental Programming Course
Week 13

Huseh-Ting Chu@Asia University, 2022

Schedule



M1

M2

M3

Week 1

Week 6

Week 7

Week 12

Week 13

Week 18

(9/12)

(10/17)

M1Test

(10/24)

(11/28)

M2Test

(12/5)

(1/9)

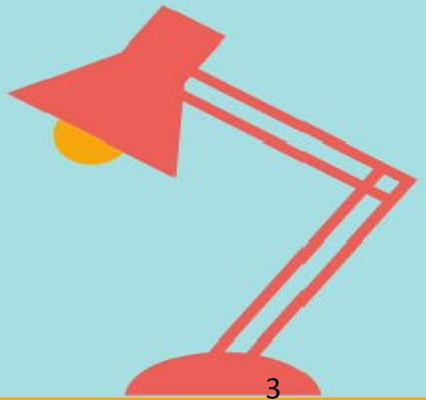
M3Test

Syllabus

- W1-Python Introduction and Programming Tools
- W2-Variables and Operations
- W3-Loop and formatted output
- W4-Condition and Containers
- W5-String and built-in functions
- W6-M1 test

- W07-Dictionary Container
- W08-File I/O
- W09-Function
- W10-Advanced flow control
- W11-Advanced operations and generators
- W12-M2 test

- W13-Advanced functions (definitions and calls, Recursion)
- W14-Class fundamentals (classes, objects, properties, constructors, methods)
- W15-Advanced Classes (Static methods, class Methods and class decorators)
- W16-Modules and Packages
- W17-Advanced programming(Argparse and Venv)
- W18-M3 test

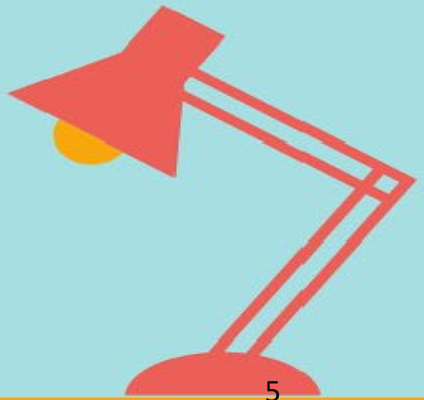


Content

- Week13-Advanced functions
  - Topic 1 - review of function definitions and calls
  - Topic 2 - data types of parameters and return values
  - Topic 3 - recursive function (Recursion)
  - Topic 4 - Lambda function (Anonymous Function)
  - Topic 5 - JSON format
  - Topic 6 - random number (random)
  - Topic 7 - charts

Topic 1- function definition and function call

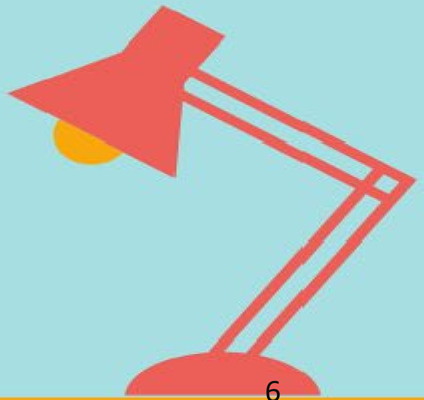
- function definition and function call
- function call
- Variable-Length Arguments *
- Variable-Length Arguments **



Topic 2- Data types for parameters and return values (new in version 3.5)

```
def gcd1(x, y):
  r=x%y
  while r!=0:
    x=y;y=r
    r=x%y
  return y

a=240; b=96
print(f"{a}, {b}, {gcd1(a,b)}")
```

```
def gcd2(x:int, y:int)->int:
  r=x%y
  while r!=0:
    x=y;y=r
    r=x%y
  return y

a=240; b=96
print(f"{a}, {b}, {gcd2(a,b)}")
```



Topic 3- Recursion

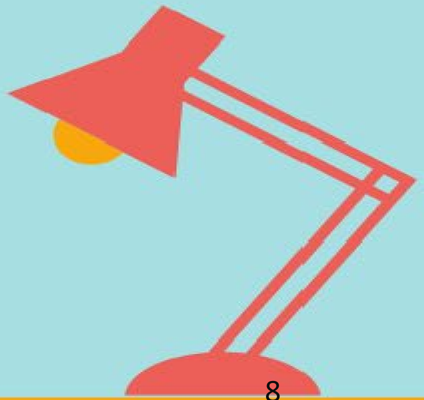
```
def gcd1(x, y):
  r=x%y
  while r!=0:
    x=y;y=r
    r=x%y
  return y

a=240; b=96
print(f"{a}, {b}, {gcd1(a,b)}")
```

```
def gcd2(x, y):
    if y == 0:
        return x
    else:
        return gcd2(y, x%y)


a=240; b=96
print(f"{a}, {b}, {gcd2(a,b)}")
```

Topic 4-Lambda function

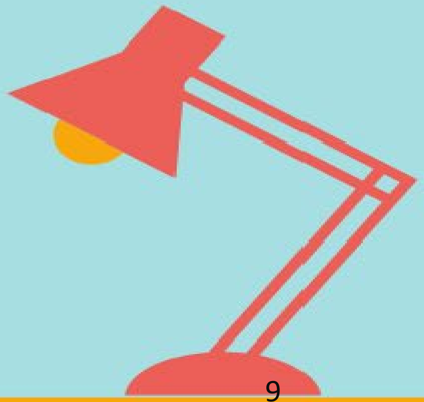
```
def gcd1(x, y):
  r=x%y
  while r!=0:
    x=y;y=r
    r=x%y
  return y

a=240; b=96
print(f"{a}, {b}, {gcd1(a,b)}")
```

```
gcd = lambda x,y : (gcd(y,x%y)) if x%y else y

a=240; b=96
print(f"{a}, {b}, {gcd(a,b)}")
```

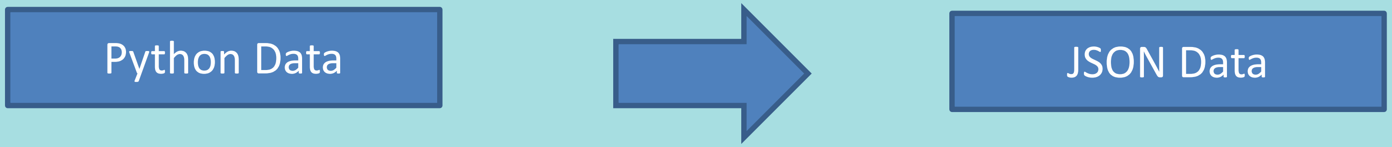


Using Lambda function in map() and filter()

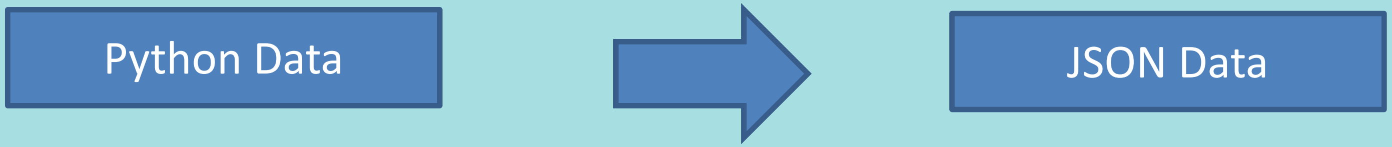
```
base=[1, 2, 3, 4]

square=list(map(lambda x: x**2, base))
print(square)




data=[0,1,2,3,4,7,8,14,19,31,34]

odds =list(filter(lambda x: x%2 > 0, data))
print(odds)
evens=list(filter(lambda x: x%2 ==0, data))
print(evens)
```



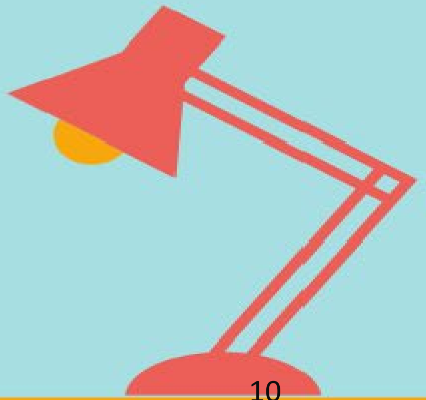
Topic 5-JSON



```
import json
data = [ { 'a' : 1, 'b' : 2, 'c' : 3, 'd' : 4, 'e' : 5 } ]
file = open("test.json", "w")
json.dump(data, file)
file.close()
```



```
import json
file = open ('test.json')
json_data = json.load(file)
file.close()
```

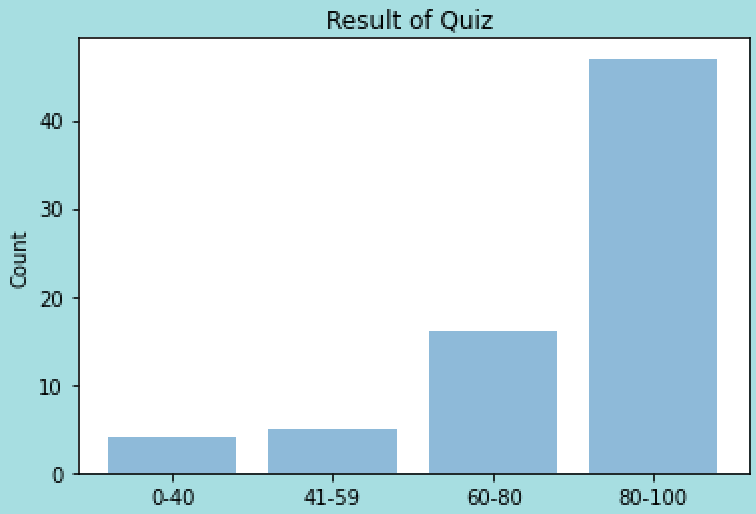


Topic 6 – random numbers

```
rd.seed(12345)
a=rd.random()                             # Random float:  0.0 <= x < 1.0
b=rd.uniform(2.5, 10.0)                   # Random float:  2.5 <= x < 10.0
c=rd.randrange(10)                        # Integer from 0 to 9 inclusive
d=rd.choice(['win', 'lose', 'draw'])      # Single random element from a seq
uence
print(a,b,c,d)
```

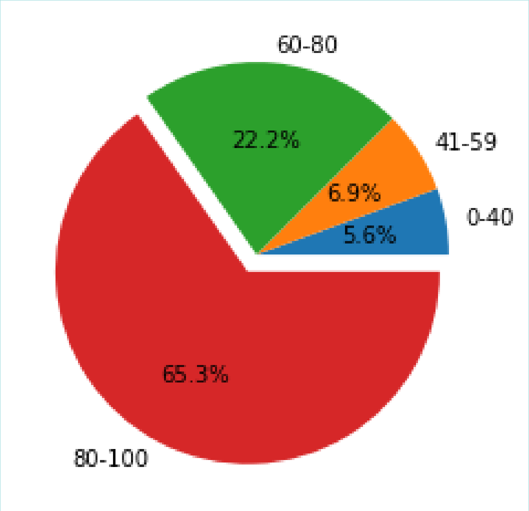
Topic 7-plotting a chart

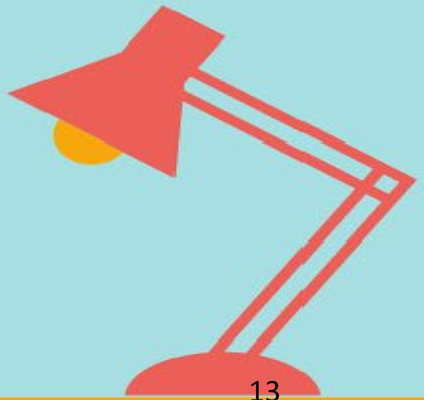
```
#Draw a bar chart
import matplotlib.pyplot as plt
import numpy as np
stats = [4, 5 ,16 , 47]
labels = ('0-40', '41-59', '60-80', '80-100')
y_pos = np.arange(len(stats))
plt.bar(y_pos, stats, align='center', alpha=0.5)
plt.xticks(y_pos, labels)
plt.ylabel('Count')
plt.title('Result of Quiz ')
plt.show()
```



Topic 7- Presenting data with pie charts

```
import matplotlib.pyplot as plt

# Pie chart

stats = [4, 5 ,16 , 47]
labels = ('0-40', '41-59', '60-80', '80-100')
explode = (0, 0, 0,  0.1)  # slicing the number of '80-100'
fig1, ax1 = plt.subplots()
ax1.pie(stats, labels=labels, explode=explode, autopct='%.1f%%')
plt.show()
```





Thanks!

Q&A