PYTHON

Day6

RECAP

- function
- File I/O
- List Comprehension

MiraeN

INDEX

- List Comprehension again!
- handle excel with python
- Error Exception
- Class

MiraeN

PYTHON

List Comprehension

MiraeN

LIST COMPREHENSION

존재하는 리스트를 활용하여 새로운 리스트를 생성하는 방법

MiraeN

LIST COMPREHENSION

```
old_list = [1, 2, 3, 4, 5]



doubled_list = []

for i in old_list:

    doubled_list.append(i * 2)
```


MiraeN

LIST COMPREHENSION

```
old_list = [1, 2, 3, 4, 5]
```

```
doubled_list = []

for i in old_list:

    doubled_list.append(i * 2)
```

```
doubled_list = []
```


MiraeN

LIST COMPREHENSION

```
old_list = [1, 2, 3, 4, 5]
```

```
doubled_list = []

for i in old_list:

    doubled_list.append(i * 2)
```

```
doubled_list = [i * 2]
```


MiraeN

LIST COMPREHENSION

```
old_list = [1, 2, 3, 4, 5]
```

```
doubled_list = []

for i in old_list:

    doubled_list.append(i * 2)
```

```
doubled_list = [i * 2 for i in old_list]
```


MiraeN

LIST COMPREHENSION

```
old_list = [1, 2, 3, 4, 5]

doubled_list = []

for i in old_list:
    if i % 2 == 0:
        doubled_list.append(i * 2)
```


MiraeN

LIST COMPREHENSION

```
old_list = [1, 2, 3, 4, 5]

doubled_list = []

for i in old_list:
    if i % 2 == 0:
        doubled_list.append(i * 2)
```

```
doubled_list = [i * 2]
```


MiraeN

LIST COMPREHENSION

```
old_list = [1, 2, 3, 4, 5]

doubled_list = []

for i in old_list:
    if i % 2 == 0:
        doubled_list.append(i * 2)
```

```
doubled_list = [i * 2 for i in old_list]
```


MiraeN

LIST COMPREHENSION

```
old_list = [1, 2, 3, 4, 5]

doubled_list = []

for i in old_list:
    if i % 2 == 0:
        doubled_list.append(i * 2)
```

```
doubled_list = [i * 2 for i in old_list if i % 2 == 0]
```


MiraeN

MINI PROJECT

with pair programming

MINI PROJECT

1. 1부터 100까지 리스트를 생성합니다.

2. 조건문을 활용하여
3의 배수일 경우, "fizz"
5의 배수일 경우, "buzz"
15의 배수일 경우, "fizzbuzz"
나머지 경우, 그 숫자를 저장하는 프로그램을 작성하세요.

MiraeN

MINI PROJECT

* Sample output

[1, 2, fizz, 4, buzz, …, fizzbuzz, …, buzz]

MiraeN

MINI PROJECT - ANSWER

```
num_list = []
for i in range(1, 100+1):
    num_list.append(i)

fizz_list = ["fizzbuzz" if i%15==0 else "fizz" if i%3==0 else "buzz" if
i%5==0 else i for i in num_list]
```


MiraeN

PYTHON

handle excel with python

MiraeN

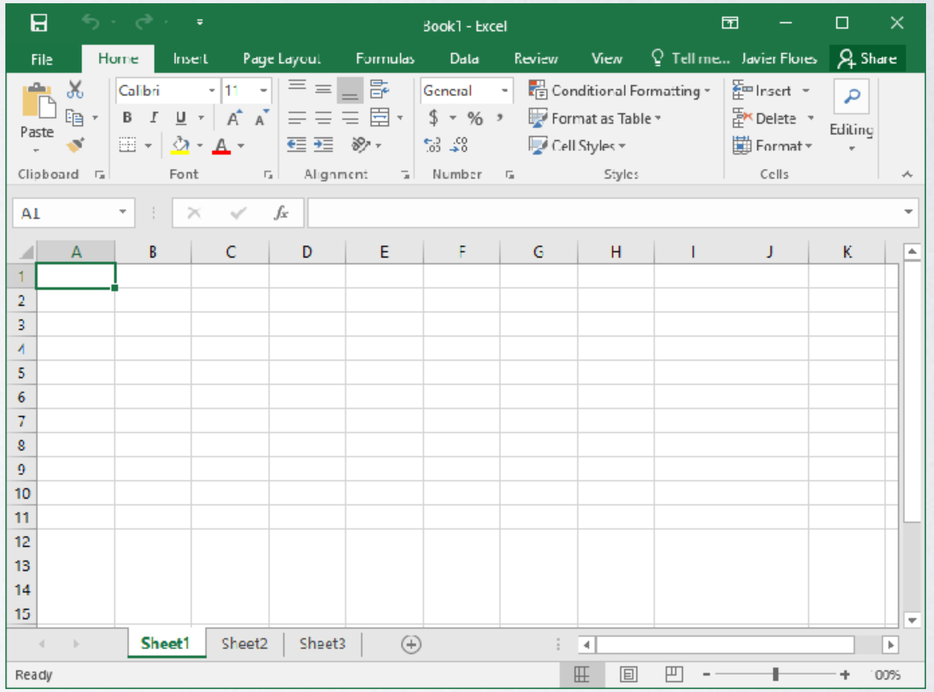
EXCEL

- Microsoft에서 발표한 스프레드 시트 프로그램

EXCEL WITH PYTHON??

- excel은 row와 column으로 구성!

- excel = [
    [1,2,3],
    [4,5,6],
    [7,8,9],
  ]



WE CAN HANDLE WITH PYTHON

pip install pandas

MiraeN

IMPORT PANDAS

```
import pandas as pd
```

READ .XLSX

```
df = pd.read_excel(“blahblah.xlsx”, sheet_name=“Sheet1”)
df
```


MiraeN

SELECT COLUMNS

df[“email”]

df[“phone”]

df[“firstname”]

..

MiraeN

DROP COLUMN

```
df.drop("firstname", axis=1)
```


MiraeN

DROP ROW

```
df.drop(1, axis=0)
```


MiraeN

SHOW INFORMATION

df.shape

df.index

df.columns

df.loc[0]

MiraeN

SAVE TO .XLSX

```
df.to_excel("user.xlsx", sheet_name = "Sheet2")
```


MiraeN

PYTHON

Error Exception

MiraeN

ERROR

- error
  - a mistake
  - the state or condition of being wrong in conduct or judgment
  - a measure of the estimated difference between the observed or calculated value of a quantity and its true value

MiraeN

ERROR COULD BE ..

- 프로그래밍에서 에러는 오동작으로 끝나지 않습니다.
  - 오동작을 역이용한 예외처리
  - 에러를 통한 효율적인 알고리즘 처리

MiraeN

ERROR EXCEPTION

try:

    정상적인 알고리즘

except:

    알고리즘 수행 중 실패했을 경우에 대한 처리

MiraeN

ERROR EXCEPTION - VALUE ERROR

```
try:
    some_input = int(input("Type the number: "))
except ValueError as e:
    print("I said type the NUMBER!!!!!")
    print(e)
```


MiraeN

ERROR EXCEPTION - FILE NOT FOUND ERROR

```
try:
    f = open("no_existance.txt","r")
except FileNotFoundError as e:
    print(e)
else:
    text = f.read()
```


MiraeN

ERROR EXCEPTION - PASS ERROR

```
try:
    f = open("no_existance.txt","r")
except FileNotFoundError as e:
    pass
else:
    text = f.read()
```


MiraeN

ERROR EXCEPTION - MULTIPLE ERRORS

```
try:
    f = open(“no_existance.txt”,“r”)
except FileNotFoundError as e:
    print(e)
except NameError as e:
    print(e)
except ValueError as e:
    print(e)
else:
    text = f.read()
```


MiraeN

PYTHON

Class

MiraeN

OOP

Object-Oriented Programming

- 객체 지향적으로 프로그래밍을 하는 것

객체의 흐름과 행위를 먼저 정의한 후, 객체의 타입에 따라 수행시키는 것

C++, Java, ..

MiraeN

CLASS

Python 도 당연히 객체 지향적으로 프로그래밍 가능합니다.

```
class Bread():   # ==> 빵이라는 객체를 정의
    def bake(self):    # ==> 메소드: 빵이라는 객체에 대한 설명
        return “Bake complete”
```


MiraeN

LET'S BAKE BREAD

```
class Bread():
    def bake(self):
        return "Bake complete"
```


MiraeN

LET'S BAKE BREAD WITH DETAILS

```
class Bread():
    def __init__(self): # ==> 클래스 객체가 만들어지면 자동으로 하는 일
        print("Let's make bread!! select dough and fill with recipe()")
    def recipe(self, dough, fill):  # ==> 클래스 내에서 쓰일 재료를 세팅
        self.dough = dough
        self.fill = fill

    def bake(self):
        return "%s and %s bake complete" % (self.dough, self.fill)
```


MiraeN

LET'S BAKE BREAD AGAIN

```
boongeo = Bread()
boongeo.recipe("wheat","redbean")
boongeo.bake()
```


MiraeN

NEXT CLASS

- Chronicles of Web
- Web Scraping with python