Fastcampus Sprint - Programming

Day 2. 파이썬 핵심문법 이해하기

Index

- 정리한 요소 저장하기(xlsx, json)
- Python 핵심문법
  - 연산과 연산자
  - 변수
  - 자료형
  - 문자열
  - 리스트
  - 조건문
  - 반복문

정리한 요소 저장하기

정리한 요소를 저장하기 위해 저장할 포맷을 먼저 정합니다.

다양한 파일 포맷

- **TXT(TeXT)**
- **CSV(Comma Spread Values)**
- TSV(Tab Spread Values)
- XML(eXtensible Markup Language)
- **XLS, XLSX(eXceL Spreadsheet (XML based))**
- **json(javascript object notation)**

TXT

```
with open('nvquery.txt', 'a') as f:
    for kw in nv_keywords:
        f.write(kw[0] + "의" +
        str(kw[1]) + "위는 " + kw[2] + "입니다.\n")
```

파일 읽기

```
with open('nvquery.txt', 'r') as f:
    text = f.readlines()
    for item in text:
        print(item)
```

CSV

```
with open('nv_query.csv', 'a') as f:
    for kw in nv_keywords:
        f.write(kw[0]+","+str(kw[1])+","+kw[2]+"\n")
```

XLSX

```
import openpyxl


workbook_name = 'nv_query.xlsx'
try:
    workbook = load_workbook(workbook_name)
except FileNotFoundError as e:
    workbook = Workbook()
worksheet = workbook.active
for keyword in nv_keywords:
    worksheet.append(keyword)

workbook.save('nv_query.xlsx')
```

write and update json

```
import json


try:
    with open('nv_query.json', 'r+') as f:
        data = json.load(f)

        data["data"].append(nv_object)
        f.seek(0)
        json.dump(data,f)
        f.truncate()
except FileNotFoundError:
    with open('nv_query.json', 'w') as f:
        json.dump({"data":[nv_object]}, f)
```

read json

```
with open('nv_query.json', 'r') as f:
    data = json.load(f)
data
```

Python

특징

- 인터프리터
- 객체지향
- 동적타이핑
- 엄격한 문법

C vs Python

```c
int main(){
int num;
for(i=0;i<=10;i++){
if (i % 2 == 0){
printf(i);
}
}
}
```

```python
for i in range(1,10+1):
    if i % 2 == 0:
        print(i)
```

Numbers & Math

`<object> <operator> <object>`

```
print(3 + 7)
print(10 - 3)
print(15 / 7)
print(34 * 100)
```

Numbers & Math

```
print(15 / 7)
print(15 / 5)
type(15 / 5)

print(15 // 5)
type(15 // 5)

print(7 % 3)

print(15 ** 3)

print(34 * 100)
print(3 * 2.5)
type(3 * 2.5)
```

Boolean

```
print(3 < 7)
print(10 < 3)
print(15 > 7)
print(3 >= 3)
print(3 <= 10)
print(34 == 100)
print(34 != 100)
```

Variable

```
print("hello python!")
hello = "hello"
python = "python!"
print(hello, python)
```

```
num1 = 14
num2 = 5

print(num1+num2)
print(num1-num2)
print(num1*num2)
print(num1/num2)
```

Let's Code PYTHONIC

Variables

- `_variable` : 내부적으로 사용되는 변수
- `print_` : 파이썬 키워드와 충돌 방지
- 클래스 이름은 `CamelCase`
- 함수, 변수, 메소드 이름은 `snake_case`

Data type

- int
- float
- long(0b, 0o, 0x)
- string
- boolean
- list, tuple, range
- set
- dictionary

type casting

float(3) --> int to float
int(3.6) --> float to int
str(1) --> int to string
int("12") --> string to int

input

```
name = input("What is your name? ")
print("Hi, ", name)
```

input with evaluation

```
input("How old are you? ")
eval(input("How old are you? "))
```

Small Project Again!

`사용자의 입력을 받아` 반지름(`r`)을 선언한 뒤, 이를 이용하여 원의 지름, 둘레, 넓이, 구의 겉넓이, 부피를 각각 출력하는 파이썬 파일을 만들어보세요.
(`pi=3.1415`)

type casting with input

```
int(input("How old are you? "))
```

String Functions

```
func = "python is easy programming language"
func.count('p')

func.find('p')

comma = ","
func = comma.join('python')

func.split(',')

python_is_easy = "python is easy"
python_is_easy.split()

python_is_easy.replace("python", "golang")
```

String Functions

```
some_string = "   computer     "
some_string.strip()
```

```
some_string = ",,,Fastcampus..."
some_string.strip(",")
some_string.strip(".")
```

String Formatting

```
print("I have a {}, I have an {}.".format("pen", "apple"))
```

```
print("I have a {0}, I have an {1}.".format("pen", "apple"))
```

```
print("I have a {0}, I have an {0}.".format("pen", "apple"))
```

padding and align

- `{:10}`
- `{:>10}`
- `{:^10}`
- `{:_^10}`

List

List

```
animals = ['','','']
```

List

빈 list를 선언합니다. 선언과 동시에 값을 채워넣을 수 있습니다.

```
lang = ["python", "c", "java", "golang"]
```

lang = []

list에 요소를 추가합니다.

lang.append("python")
lang.append("java")
lang.append("golang")
print(lang)

혹은 특정한 위치에 원하는 값을 추가할 수 있습니다.

```
lang.insert(1, "c")
print(lang)
```

특정 요소를 삭제할 수도 있습니다.

```
lang.remove("golang")
print(lang)
```

혹은 리스트에 있던 값을 빼낼 수도 있습니다.

```
java = lang.pop(2)
print(lang)
print(java)
```

리스트를 정렬하는 법을 알아봅니다.

```
numbers = [2, 1, 4, 3]
print(numbers)

numbers.sort()
print(numbers)
```

리스트를 역순으로 출력하고 싶을땐 이렇게 한답니다.

```
numbers = [2, 1, 4, 3]
numbers.reverse()
print(numbers)
```

리스트를 내림차순으로 정렬하려면??

1. sort -> reverse

```
numbers.sort()
numbers.reverse()
```

2. sort(reverse=True)

```
numbers.sort(reverse=True)
```

특정 값의 위치를 출력할땐 이렇게 합니다.

```
index_of_two = numbers.index(2)
print(index_of_two)
```

리스트끼리 더할 땐 extend를 활용합니다.

```
numbers += [5, 6]
print(numbers)
numbers.extend([7, 8])
print(numbers)
```

조건문

Conditional Statements

`배가 고프다!!!`

- case 1: 집이라면
  - 밥이 있다면
  - 밥이 없다면
- case 2: 밖이라면
  - 현금이 10만원 초과라면
  - 현금이 5만원 초과라면
  - 현금이 없다면

If

```
if 조건:
        실행문

if 조건1 and 조건2:
        실행문

if 조건1 or 조건2:
        실행문
if not 조건:
        실행문
```

Comparison Operators

```
x == n
x != n

x < n
x > n
x <= n
x >= n
```

if

```
if 현금 > 100000:
        레스토랑으로 간다
```

```
cash = 120000
if cash > 100000:
        print("go to restaurant")
```

else

```
if 조건:
        실행문1
else:
        실행문2
```

```
cash = 120000
if cash > 100000:
    print("go to restaurant")
else:
    print("go to cvs")
```

else if

```
if 조건1:
        실행문1
else:
        if 조건2:
                실행문2
        else:
                실행문3
```

```
cash = 120000
if cash > 100000:
    print("go to restaurant")
else:
        if cash > 50000:
            print("go to bobjib")
        else:
            print("go to cvs")
```

if in else in if in else in ..

```
cash = 120000
if cash > 100000:
    print("go to restaurant")
else:
    if cash > 50000:
        print("go to bobjib")
    else:
        if cash > 30000:
            print("go to buffet")
        else:
            if cash > 20000:
                print("go to ramen store")
            else:
                if cash > 10000:
                    print("go to chinese restaurant")
                else:
                    print("go to cvs")
```

elif

```
if 조건1:
        실행문1
elif 조건2:
        실행문2
elif 조건3:
        실행문3
...
else:
        실행문n
```

elif

```
cash = 120000
if cash > 100000:
    print("go to restaurant")
elif cash > 50000:
    print("go to bobjib")
elif cash > 30000:
    print("go to buffet")
elif cash > 20000:
    print("go to ramen store")
elif cash > 10000:
    print("go to chinese restaurant")
else:
    print("go to cvs")
```

If with Web scraper

```
box_office_list = []
for tr in tr_list:
    a_list = tr.find_all("a")
    score = int(a_list[0].find("span", attrs={"class":"tMeterSco
    url = base_uri + a_list[0]["href"]
    movie_name = a_list[1].text
    revenue = a_list[2].text[1:]

    # convert scale to zeros
    if revenue[-1] == 'M':
        digits = 6
    elif revenue[-1] == 'B':
        digits = 9
    elif revenue[-1] == 'K':
        digits = 3

    if revenue.find('.') == -1:
        revenue = int(revenue[:-1] + '0'*digits)
    elif revenue[0] == '0':
        revenue = int(revenue[1:-1].replace(".","") + '0'*(digit
    else:
        revenue = int(revenue[:-1].replace(".","") + '0'*(digits

    box_office_list.append((executed_time, url, score, movie_nam
box_office_list
```

Do it your self!

Numguess

- 1부터 100까지 정수 중 하나를 `answer` 라는 변수에 할당
- 사용자로 부터 임의의 값 하나를 받아 `guess` 라는 변수에 할당
- `answer` 와 `guess` 를 비교하여 정답여부를 출력

numguess

```
import random


answer = random.randint(1,100)
print(answer)
```

numguess

```
username = input("Hi there, What's your name?? ")
guess = eval(input("Hi, "+ username + "guess the number: "))

if guess == answer:
        print("Correct! The answer was ", str(answer))
else:
        print("That's not what I wanted!! The answer was ", str(
```

Iteration

For, while

```
for 변수 in (리스트 or 문자열):
        실행문1
    ...
```

```
for i in ["python", "java", "golang"]:
        print(i)
```

For, while

```
sum = 0
for i in range(1,11):
        sum += i
    sum = sum + i
        print(sum)
```

For, while

```
while 조건:
        실행문1
        ...
```

```
while name != "foo bar":
        name = input("What's your name? ")
        print("Hi, " + name + "So, where is foo bar?")
```

```
while 1:
        print("Hello world!")
```

Iterations with Conditional Statements

Fizzbuzz

1부터 100까지 **반복하면서,**

3의 배수 = "Fizz"
5의 배수 = "Buzz"
15의 배수 = "FizzBuzz"
나머지 = 그 숫자

Fizzbuzz

```
num = eval(input("type the number: "))

for i in range(1, num + 1):
        if i % 15 == 0:
                print("fizzbuzz")
        elif i % 3 == 0:
                print("fizz")
        elif i % 5 == 0:
                print("buzz")
        else:
                print(i)
```

Refactoring numguess

```
import random


answer = random.randint(1,100)
username = input("Hi there, What's your name?? ")

while True:
        guess = eval(input("Hi "+ username + ", guess the number
        if guess == answer:
                print("Correct! The answer was ", str(answer))
                break
        else:
                print("That's not what I wanted!! Try again!!")
```

give a hint!!

```
import random


answer = random.randint(1,100)
username = input("Hi there, What's your name?? ")

while True:
    guess = eval(input("Hi, "+ username + "guess the number: "))

    if guess == answer:
        print("Correct! The answer was ", str(answer))
        break
    elif guess > answer:
        print("Too high!! Try again!!")
    elif guess < answer:
        print("Too Low!! Try again!!")
```

limit trial

```
import random


answer = random.randint(1,100)
username = input("Hi there, What's your name?? ")
trial = 5
while trial:
    guess = eval(input("Hi, "+ username + ". guess the number: "

    if guess == answer:
        print("Correct! The answer was ", str(answer))
        break
    elif guess > answer:
        trial -= 1
        print("Too high!! Try again!!(%d times left)" % (trial))
    elif guess < answer:
        trial -= 1
        print("Too Low!! Try again!!(%d times left)" % (trial))

if trial == 0:
    print("You are Wrong! The answer was ", str(answer))
```