파이썬 프로그래밍 기초1

전체 목차

1. python 기초
    - 입출력
    - 자료형(숫자형, 문자형, 불, 리스트, 튜플, 딕셔너리)
    - 조건문, 반복문
    - 함수
2. 크롤링

3. 머신러닝(딥러닝 포함)

목차

1. 강의 소개
2. 프로그램이란
3. 프로그래밍을 하는 이유
4. 파이썬을 사용하는 이유
5. 프로그래밍 언어 선택
6. 파이썬 프로그램 시작해보기
   (입출력, 변수, 자료형)

강의소개

- 파이썬을 한 번도 배워본 적 없고, 프로그래밍이 처음이라고 생각하고 진행
- 초반에 배우는 개념들을 잘 이해하시면 후반 코딩에도 무리가 없습니다
- 너무 쉬운 걸 배우는 것은 아닐까 걱정하지마세요 금방 말도 안되게 어려워집니다.
- 머리가 상당히 좋으시면 쉬우실 수도 있습니다.
- 이론을 하나씩 배울 때마다 실습을 한 번씩 수행
- 백견이 불여일타라는 말이 있습니다.
- 스스로 사고하고 구현해보는 과정이 중요하다고 생각합니다.
- 각 실습은 3~5분정도의 시간을 드리고 각자 수행해봅니다
- 추가적으로 궁금한 것들은 구글링을 하시면 프로그래밍 실력에 큰 도움이 됩니다.

프로그램이란

- 컴퓨터에게 내리는 명령의 집합 (명세서)
- A 라는 작업을 하고 그게 끝났으면 B 라는 작업을 하라...
- 컴퓨터가 알아듣는 여러 언어로 작성될 수 있음
  (C, C#, C++ Java, Javascript, Python ...)

프로그래밍을 하는 이유

- 일상 속의 특정한 문제를 해결하기 위해
- 프로그램의 대부분 해결하고자 하는 문제가 존재해서 작성 됨
- 사람이 하기 귀찮은 일이 있을 때
- 사람은 반복되는 작업에 지루함을 느끼고, 반복되는 일들은 '자동화' 될 수 있음
- 사람이 실수하기 쉬운 일이 있을 때
- 컴퓨터는 언제나 정해진대로 순차적인 명령을 따르는 것이 보장 됨
- 프로그램이 해결해줄 수 있는 일상속의 문제들
- 알람 (컴퓨터는 늦잠을 자지 않으며, 깨우는 귀찮음도 느끼지 않는다)
- 계산기 (컴퓨터는 실수를 하지 않는다)
- 인공지능?
- 현재의 인공지능은 내부적인 복잡한 수학적 계산을 자동화

파이썬을 사용하는 이유 #1

- 인간의 언어에 가까운 프로그래밍 언어 (If you are fluent in English!)
- 문법이 매우 간단명료하여, 다른 프로그래밍 언어에 비해 접근성이 좋음
- 인공지능 라이브러리들이 잘 구현되어 있다
- 라이브러리란 성능과 안정성이 보장된 미리 작성된 코드들임

```
public static void main(String args[]) {
System.out.println("안녕하세요");
}
```

=

```
print("안녕하세요")
```

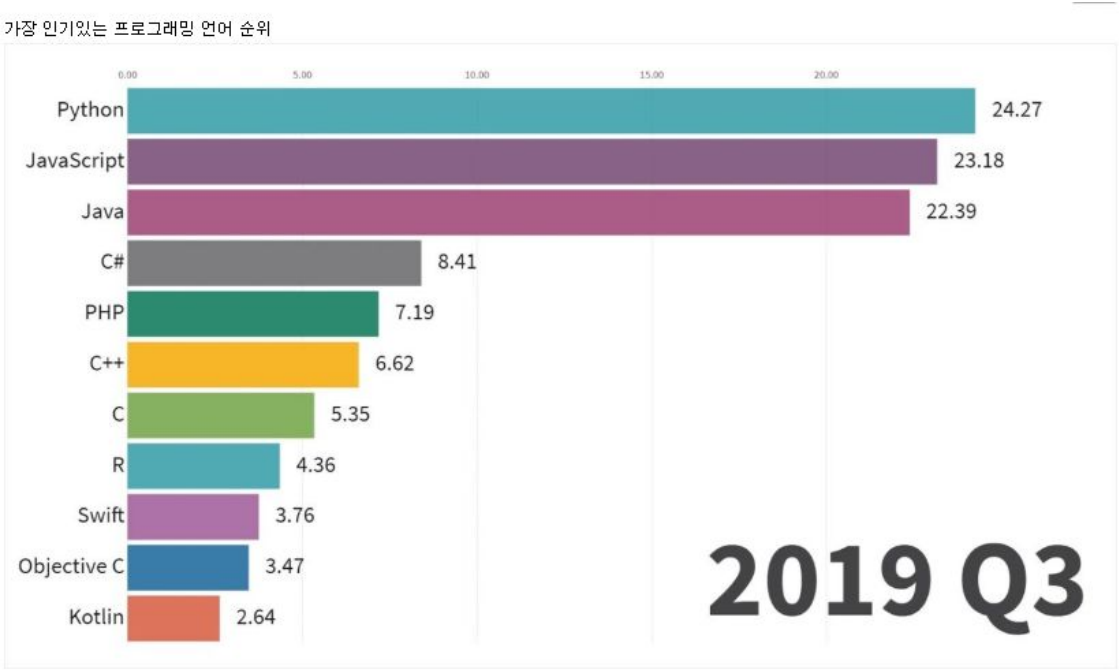
하지만 유의해야 할 점! 짧다고 무조건 좋은 것은 아님.. 긴 데는 긴 이유가 있음!
하지만 프로그래밍 입문 용도로 파이썬을 선택하는 것은 매우 좋은 선택!

파이썬을 사용하는 이유 #2

- 프로그래밍 언어의 선택은 사실 무엇을 만드냐에 따라 결정되는 경우가 많음
- 파이썬을 사용하면 만들기 좋은 것들?
- 인공지능 모듈, 간단한 웹 서버

  - 다른 언어들은 어디에 쓰일까?
    - C언어 (퍼포먼스가 중요한 프로그램들 ...)
    - 자바스크립트 (웹페이지 만들기, 웹서버, 클라이언트 프로그램)
    - 자바 (엔터프라이즈 앱, 모바일 어플리케이션 등...)

가장 인기있는 프로그래밍 언어 순위



https://www.fmkorea.com/2334959386

파이썬 환경 구축하기 #1

- https://www.python.org/ 에 접속하면 Downloads라는 항목에서 다운로드 가능 (오른쪽 하단 참조)

- 파이썬은 대부분의 운영체제에서 돌릴 수 있음
- Windows, Linux, Mac OS …
- 파이썬 설치는 파이썬 공식 홈페이지에서 가능
- 윈도우에서 파이썬을 설치하면 기본적으로 IDLE이라는 도구를 제공함
- 하지만 우리는 IDE를 설치해서 IDE로 코딩을 수행할 것
- IDE는 Integrated Development Environment의 약자로 개발을 도와주는 환경을 말함
- 대표적 IDE로는 Pycharm, Vscode 등이 있음
- 우리는 Vscode를 사용해서 코딩해볼 예정!

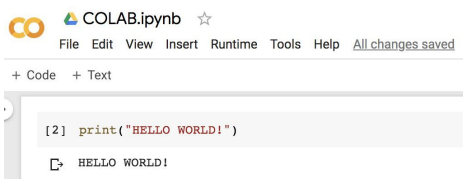
파이썬 환경 구축하기 #2 (Pycharm VS Vscode)

1. Pycharm
- Jetbrain 개발
- 사실 python을 사용하는 유저들이 가장 많이 사용하는 IDE
- Community 버전과 Pro 버전이 있으며, Pro버전은 유료임

2. Vscode
- Microsoft 개발
- 요즘 핫한 IDE, 이 IDE하나로 여러가지 개발언어 개발 가능 (Pycharm은 파이썬만 가능)
- (사실 자바스크립트 개발쪽에서 핫함)
- 매우 가벼워서 용량도 적으며 완전히 무료임

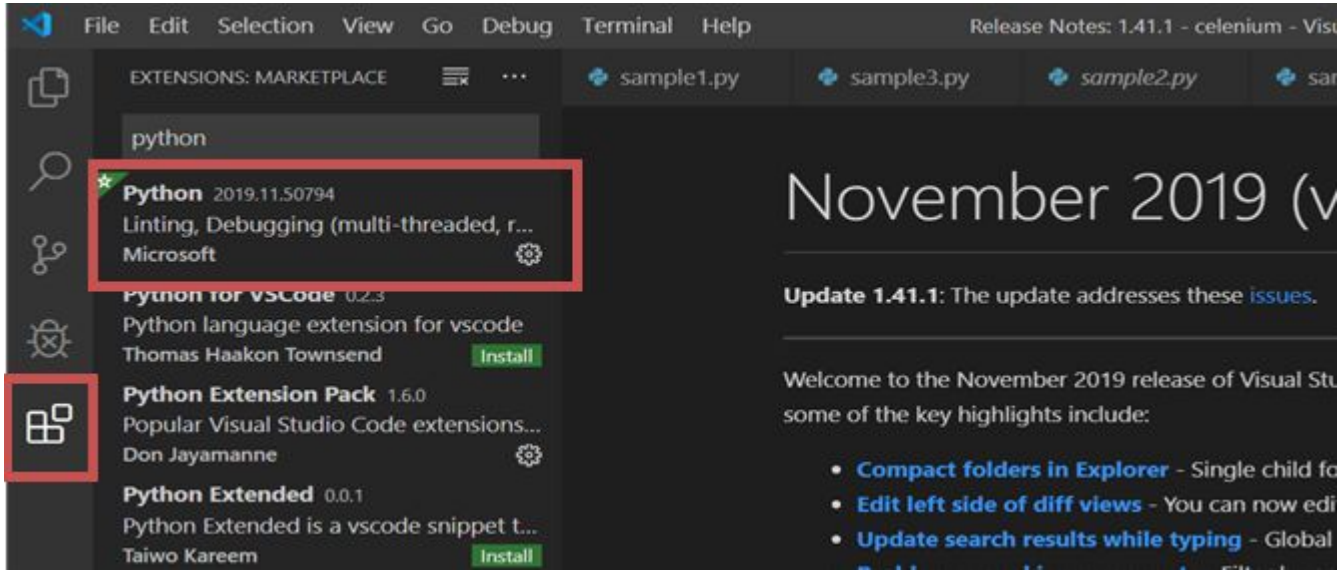
파이썬 환경 구축하기 #3 (Colab)

- 코랩이란? 구글에서 무료로 제공하는 웹상의 파이썬 개발환경
- 따로 파이썬 코딩 툴을 설치하지 않고도 파이썬 코딩을 시작할 수 있는 방법
- 사실, 실무자에게도 가장 시간 소모가 많이 되는 부분이 환경 구축이지만, 코랩을 이용하면
파이썬 개발 시에 별달리 다른 세팅을 할 필요가 없음
- 또한 최신형 GPU 환경도 무료로 제공하여, 실습할 GPU가 없는 사람들에게 매우 좋음
- 이용 방법은 간단히 사이트에 접속하여 구글 아이디를 로그인하기만 하면 됨
- http://colab.research.google.com



파이썬으로 Hello World 프린트해보기

간단하게, print(“Hello World”) 로 프린트가 가능
- Print, pritn 등 한 글자만 오타가 있어도 프로그램이 구동되지 않으니 주의해야 함



파이썬으로 Hello World 프린트해보기

```
print(“Hello World”)
```

- VS Code에서 파이썬 코드를 작성하기 위해서는 확장자 .py 를 가진 임의의 파이썬 파일을 생성하면 된다. [File] 메뉴에서 [New File]을 눌러 새 파일을 만들어도 되지만, VS Code의 기능을 100% 활용하기 위해서는 [File] 메뉴에서 [Open Folder]를 사용하여 작업 폴더를 지정하고 사용하는 것이 좋다.

코드 작성 완료 후 프로그램을 실행하기 위해서는 [Debug] 메뉴의 [Start Debugging (F5)] 나 [Start without Debugging (Ctrl+F5)]를 누르면 된다.

파이썬으로 변수생성하기 #1

- 파이썬의 변수 선언은 그 어느 언어보다 간단함
- A = B라고 정의하면 A라는 변수에 B가 들어감
    - 프로그래밍에서 '=' 은 수학기호 '=' 과 다르다는 것에 유의하면 좋음
    - 프로그래밍에서 '=' 은 왼쪽 피연산자(A)에 오른쪽 피연산자(B)를 넣는다는 의미
    - 수학기호 '='에 상응하는 기호는 프로그래밍에서는 '==' 임

- 'a' 라는 변수에 10을 넣고 출력해보기

- 변수에는 아래 예제와 같이 숫자도 들어갈 수 있지만, 문자나 기타 여러가지 값도 들어갈 수 있음

파이썬으로 변수생성하기 #2

- 파이썬 변수의 특징
    - 변수에 할당된 값은 언제든 변할 수 있음 (재할당이 가능)
    - 변수의 이름은 그 값이 의미하는 뜻으로 작성하는 것이 좋음
        - ex) year = 2019, month = 11 과 같이 작성하면 코드 작성 이후 다시 코드를 볼 때 이해가 쉬움
        - 협업에서는 매우 중요한 Rule중 하나임
        - 모두가 한눈에 내 변수명을 이해하는 것이 중요
- 숫자로 시작하는 이름은 사용할 수 없음
- 띄어쓰기를 포함할 수 없음
    - 단, 띄어쓰기 대신에 '_'를 많이 이용함
- 한글로도 변수명을 설정 가능하지만, 거의 필수적으로 영어로 작성해야 함

파이썬 입출력 #1

- 주석
  - 코드에 대한 설명을 달아 놓는 것
  - '#' 기호와, '' '' 기호를 이용하여 작성

- 출력 : print()

print_example1.py

– 하나만 출력하기

```
>>> print("Hello! Python Progrmming...")
Hello! Python Progrmming...
>>> print(52)
52
>>> print(273)
273
```

파이썬 입출력 #1

- 출력 : print()

print_example1.py

– 여러개 출력하기

```
>>> print(52,273,"Hello")
52 273 Hello
>>> print("안녕하세요","저의","이름은","홍길동입니다.")
안녕하세요 저의 이름은 홍길동입니다.
```

– 줄바꿈하기

```
print()

#아무것도 출력되지 않고 단순히 줄바꿈
```

파이썬 입출력 #2

- 출력 : print()

print_example2.py

– 문자열 포맷팅1

```
>>> this_year = 2020
>>> this_month = 1
>>> print("this year is %d and this month is %d" %(this_year, this_month))
this year is 2020 and this month is 1
>>> py = 3.14
>>>print("py is %f" %py)
py is 3.140000
```

파이썬 입출력 #2

- 출력 : print()

print_example2.py

– 문자열 포맷팅2 : { }기호와 .format()메소드를 이용하여 포맷팅

```
>>>py = 3.141529
>>>print("py is {:.3f}".format(py))
py is 3.14
>>> my_name = 'park'
>>> your_name = 'lee'
>>>print("my name is { } and your name is { }".format(my_name, your_name))
my name is park and your name is lee
```

파이썬 입출력 #3

- 입력 : input()

input_example1.py

```
>>> string = input("인사말을 입력하세요>")
인사말을 입력하세요>안녕하세요
>>> print(string)
안녕하세요
```

```
>>> number = input("숫자를 입력하세요>")
숫자를 입력하세요>1234
```

파이썬 입출력 #4

input_example2.py

- 입력 : input()

  input은 항상 문자열

```
>>> number = input("숫자를 입력하세요>")
숫자를 입력하세요>1234
>>> number = number + 1 #error
>>> print(number)
TypeError: must be str, not int
```

```
>>> number = input("숫자를 입력하세요>")
숫자를 입력하세요>1234
>>> number = number + str(1) #casting
>>> print(number)
12341
```

```
>>> number = int(input("숫자를 입력하세요>")) #eval함수를 이용해도 됨
숫자를 입력하세요>1234
>>> number = number + 1
>>> print(number)
1234
```

파이썬 입출력 #5

- 문자열과 이스케이프
  -" " 과 ' '둘다 사용가능하다.
  - '로 시작했으면 '로 끝나야 한다

-이스케이프(escape)문자

input_example3.py

```
>>> print('안녕하세요')
안녕하세요
>>> print('"안녕하세요"라고 말했습니다.)
"안녕하세요"라고 말했습니다.

>>> year = input("This year: ")
This year: 2020
>>> year = eval(year) # eval은 문자열을 숫자로 바꿈
>>> year = year + 1
>>> print("Next year:", year)
Next year: 2021
```

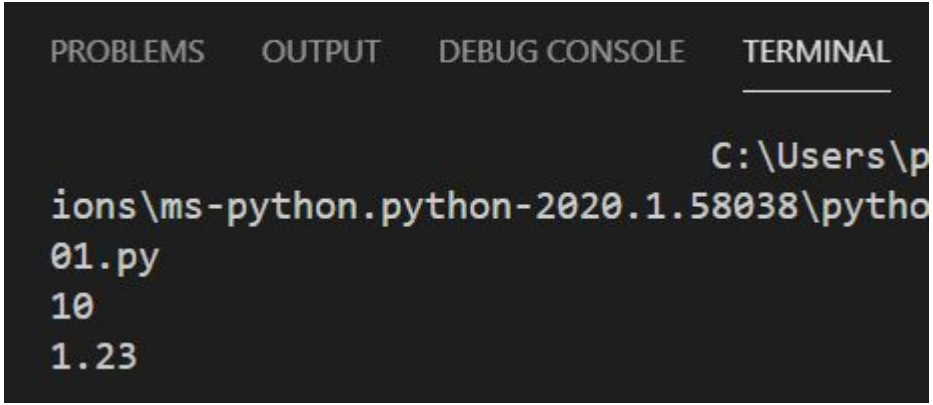
```
# 이스케이프 문자
>>> print("\"안녕하세요\"라고 말했습니다.")
"안녕하세요"라고 말했습니다.
>>> print('"안녕하세요"라고 말했습니다.')
"안녕하세요"라고 말했습니다.
```

파이썬의 자료형 알아보기 #1

numeric_example.py

- 숫자형(Numeric)
  - 정수, 실수 등이 포함

```
1    a=10
2    b=1.23
3    print(a)
4    print(b)
```



파이썬의 자료형 알아보기 #2

string_example.py

- 문자열(String)
  - 문자 1글자 혹은 여러문자, 큰따옴표나 작은 따옴표로 구분한다.
    문자열의 +(덧셈)은 숫자의 덧셈과는 다르다.

```
string.py > ...
1   #문자열
2   num1 = "20" + "20"
3   print(num1)
4   num2 = 20 + 20
5   print(num2)
6
```

```
2020
40
```

파이썬의 자료형 알아보기 #2

string_example.py

- 문자열(String)

“Py is 3.14”라는 문자를 출력해 보자

- py_str = “py”
- py_num = 3.14
- print(...)
- 결과 : “py is 3.14”

파이썬의 자료형 알아보기 #2

string_example.py

- 정답

```
>>> py_str = 'py'
>>> py_num = 3.14
>>> print(py_str + " is " + str(py_num))
py is 3.14
```

- py_str = 'py' ,py_num = 3.14로 결과 py is 3.14를 출력하려면 다음과 같이 출력해야 한다.

- 문자열 여러개는 '+'기호로 묶을 수 있다.

- 숫자는 문자열과 함께 '+'기호로 묶을 수 없으므로 string으로 변환해 줘야 한다.

파이썬의 자료형 알아보기 #3

- 불린(Boolean)

  - 참(True) 또는 거짓(False)을 표현하기 위한 자료형

  - 참을 1, 거짓을 0으로 표현하는 언어도 많음

  - bool1 = True

  - bool2 = False

  - bool3 = 10 > 5

  - 참(True)

  - bool4 = 3 == 4

- 거짓(False), 할당 연산자와 헷갈리지 않기

파이썬의 자료형 알아보기 #3

- 불린(Boolean)

  - 불린은 if 조건문에서 많이 쓰인다.

boolean_example.py

```
>>> bool1 = True
>>> bool2 = 10<5
>>> print(bool1)
>>> print(bool2)
True
False
```

파이썬의 자료형 알아보기 #3

- 불린(Boolean)

boolean_example.py

```
1   boolean_ex_1 = "1" == 1
2   boolean_ex_2 = 1 == (2 - 1)
3   boolean_ex_3 = "1" == str(1)
4   boolean_ex_4 = 1 == str(1)
5
6   print(boolean_ex_1)
7   print(boolean_ex_2)
8   print(boolean_ex_3)
9   print(boolean_ex_4)
10
```

파이썬의 자료형 알아보기 #4

- 리스트(List)
  - 여러 가지 자료를 저장할 수 있는 자료 [] 대괄호 내부에 여러 종류의 자료를 넣어서 선언한다.

list_example1.py

```
>>> array = [273, "문자열", True]
>>> print(array)
[273, '문자열', True]
>>> array[0]
273
>>> array[1:2]
['문자열']
>>> array[:]
[273, '문자열', True]
>>> array[-1]
True
>>> array[1][0]
'문'
```

파이썬의 자료형 알아보기 #4

- 리스트(List)
  - 리스트의 연산

list_example1.py

```
>>> list1 = [1.,2.,3.]
>>> list2 =[4.,5.,6.]
>>> print(list1 + list2)
[1.0, 2.0, 3.0, 4.0, 5.0, 6.0]
>>> print(type(list1))
<class 'list'>
>>> print(type(list1[0]))
<class 'float'>
>>> list3 = [1,2,3]*2
>>> print(list3)
[1, 2, 3, 1, 2, 3]
```

파이썬의 자료형 알아보기 #4

- 리스트(List)
  - 리스트의 값 삽입(리스트명.append(요소)

list_example2.py

```
>>> list_a =[[1,2,3],[4,5,6],[7,8,9]]
>>> list_a[1]
[4, 5, 6]
>>> list_a[1][0]
4
>>> list_a.append([10,11,12])
>>> print(list_a)
[[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]]
```

파이썬의 자료형 알아보기 #4

- 리스트(List)

list_example3.py

[문제]

- fruit_list라는 빈 리스트를 하나 만든다.

-append()함수를 이용하여 “Banana”,”Pineapple”,”Apple”3개의 원소를 추가해본다.

-출력해본다.

```
['Banana', 'Pineapple', 'Apple']
```

파이썬의 자료형 알아보기 #4

- 리스트(List)

[정답]

list_example3.py

```
1   fruit_list = []
2   fruit_list.append('Banana')
3   fruit_list.append('Pineapple')
4   fruit_list.append('Apple')
5   print(fruit_list)
```

파이썬의 자료형 알아보기 #5

tuple_example1.py

- 튜플(tuple)

  -튜플 : 함수와 함께 많이 사용되는 리스트와 비슷한 자료형으로, 리스트와 다른 점은 한번 결정된 요소는 바꿀 수 없다

  -값들을 묶어서 저장할 때 용이(값의 그룹화)

  -여러 값 자체를 하나의 그룹(묶음)으로 보는 것

  -()괄호를 사용한다

파이썬의 자료형 알아보기 #5

- 튜플(tuple)

  -값을 변경할 수 없다.

tuple_example1.py

```
>>> tuple_test=(10,20,30)
>>> tuple_test[0]
10
>>> tuple_test[1]
20
>>> tuple_test[0]=1
Traceback (most recent call last):
  File "<pyshell#118>", line 1, in <module>
    tuple_test[0]=1
TypeError: 'tuple' object does not support item assignment
```

파이썬의 자료형 알아보기 #5

- 튜플(tuple)

  -요소를 하나만 가지는 리스트

  [273]

  -요소를 하나만 가지는 튜플

  (273,)

tuple_example1.py

```
>>> [a,b] = [10,20]
>>> (c,d) = (30,40)
>>> print(a)
10
>>> print(b)
20
>>> print(c)
30
>>> print(d)
40
```

파이썬의 자료형 알아보기 #5

tuple_example2.py

- 튜플(tuple)
  - 괄호가 없는 튜플

```
>>> tuple_test = 10,20,30,40
>>> print(tuple_test)
(10, 20, 30, 40)
>>> a,b,c = 10,20,30
>>> print(a,b,c)
10 20 30
```

```
>>> movie = ("The Dark Knight",2008)
>>> print(movie)
('The Dark Knight', 2008)

>>> movie_name, movie_year = movie
>>> print(movie_name)
>>> print(movie_year)
The Dark Knight
2008
```

파이썬의 자료형 알아보기 #5

tuple_example3.py

- 튜플(tuple)

[문제]

-me라는 이름의 튜플을 만들어보자

-me는 나의 이름과 생일로 구성되어 있다.

-me를 이용하여 name변수에 자신의 이름을 넣고, birthday변수에 생일을 넣어보자

-각각을 2번에 걸쳐 프린트해 보자.

```
('park ji hoon', '2월 18일')
park ji hoon
2월 18일
```

파이썬의 자료형 알아보기 #5

- 튜플(tuple)

tuple_example3.py

[정답]

```
1  name=('park ji hoon','2월18일')
2  print(name)
3
4  name,birthday = name
5  print(name)
6  print(birthday)
```

파이썬의 자료형 알아보기 #6

dict_example1.py

- 딕셔너리(dictionary)
  - 관련된 정보를 쌍(Pair)으로 묶어 놓은 것
  - 값의 이름을 Key라고 하고, 값을 Value라고 함
  - 사전에서 어떤 페이지를 인덱싱하는 개념정도로 이해하면 쉬움
  - {} 중괄호로 표현한다

```
>>> dict_example={}
>>> dict_example['phone'] = 'galaxy 9'  #key는 phone, value는 galaxy 9
>>> dict_example['me'] = 'park'
>>> print(dict_example)
{'phone': 'galaxy 9', 'me': 'park'}
```

파이썬의 자료형 알아보기 #6

dict_example2.py

- 딕셔너리(dictionary)

  [문제]

  -dict_favorite라는 딕셔너리 자료형을 만들어보기

  - ‘color’라는 key에 자신이 좋아하는 색상을 list로 넣어보기

  -‘food’라는 key에 자신이 좋아하는 음식을 list로 넣기

  - print()를 이용하여 좋아하는 음식 출력하기

  - print()를 이용하여 좋아하는 색상의 첫번째 출력하기

파이썬의 자료형 알아보기 #6

- 딕셔너리(dictionary)

dict_example2.py

[정답]

```
dict_favorite={}
dict_favorite['color'] = ['red','blue']
dict_favorite['food'] = ['chicken','pizza']
print(dict_favorite)
```

배운것 정리해보기

- 파이썬에서 프린트문을 사용하는 방법
    - print(“something”)

- 파이썬에서 변수를 선언하는 방법
    - variable = “string”

- 파이썬의 각종 자료형
    - 문자열 / 숫자
        - “abc” / 123
    - 불리언
        - True or False
    - 리스트
        - list = [‘a’, ‘b’, ‘c’]
    - 튜플
        - tuple = (‘value1’, ‘value2’, ‘value3’)
    - 딕셔너리
        - dictionary = {‘a’ : ‘1’, ‘b’ : ‘2’}