Python 환경설정

김경민

Python

"약 6년 전인 1989년 12월, 크리스마스를 전후하여 취미로 만들어 볼 프로그래밍 프로젝트를 찾고 있었죠. 그 때 사무실은 잠겨있어지만, 집에 컴퓨터가 있었고, 뭐 특별히 할 일도 없었죠. 그래서 그 때 당시 한동안 생각하고 있었던 새 스크립트 언어에 대한 인터프리터를 만들어 보기로 했죠. 유닉스/C 해커들에게 어필할 수 있는, ABC 언어로부터 파생된 언어말이죠. 나는 그 프로젝트명으로 Python이라는 이름을 선택했는데, 그 당시 약간은 불손한 기분이 들어서이기도 했고, 또한 당시 Monty Python's Flying Circus(BBC 코메디)에 열성팬이기도 하여..."

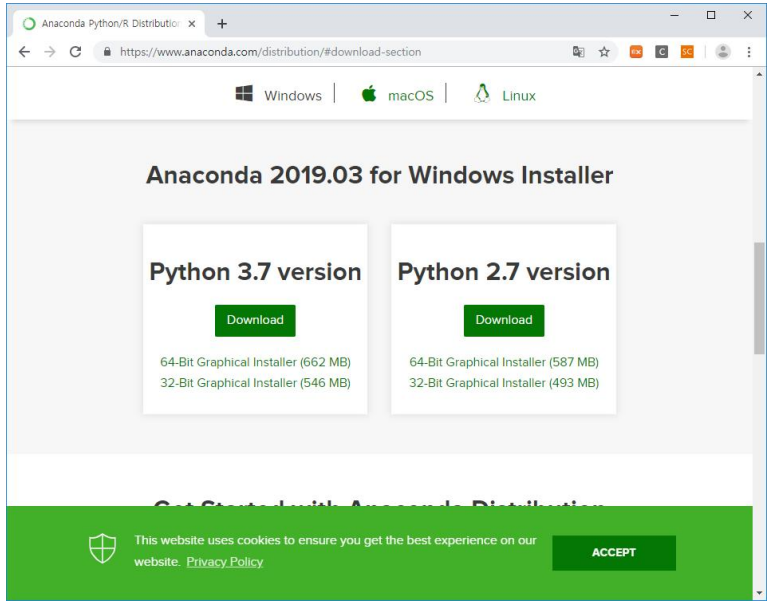
- 1996, Guido

Python 특징

- **오픈 소스로 무료로 제공된다.**
  - 다양한 라이브러리를 지원한다.
- **매우 간결하며 명시적이다.**
  - 높은 생산성을 가진다.
  - 가독성이 좋다.
  - 문법이 쉬워 빠르게 배울 수 있다.
  - 문법이 매우 엄격하다.
- **플랫폼 독립적인 언어**
  - 운영체제에 종속되지 않는다.
  - Python 바이트 코드를 생성하여 소스코드 없이도 다른 컴퓨터에서 수행된다.

Python 환경 설정

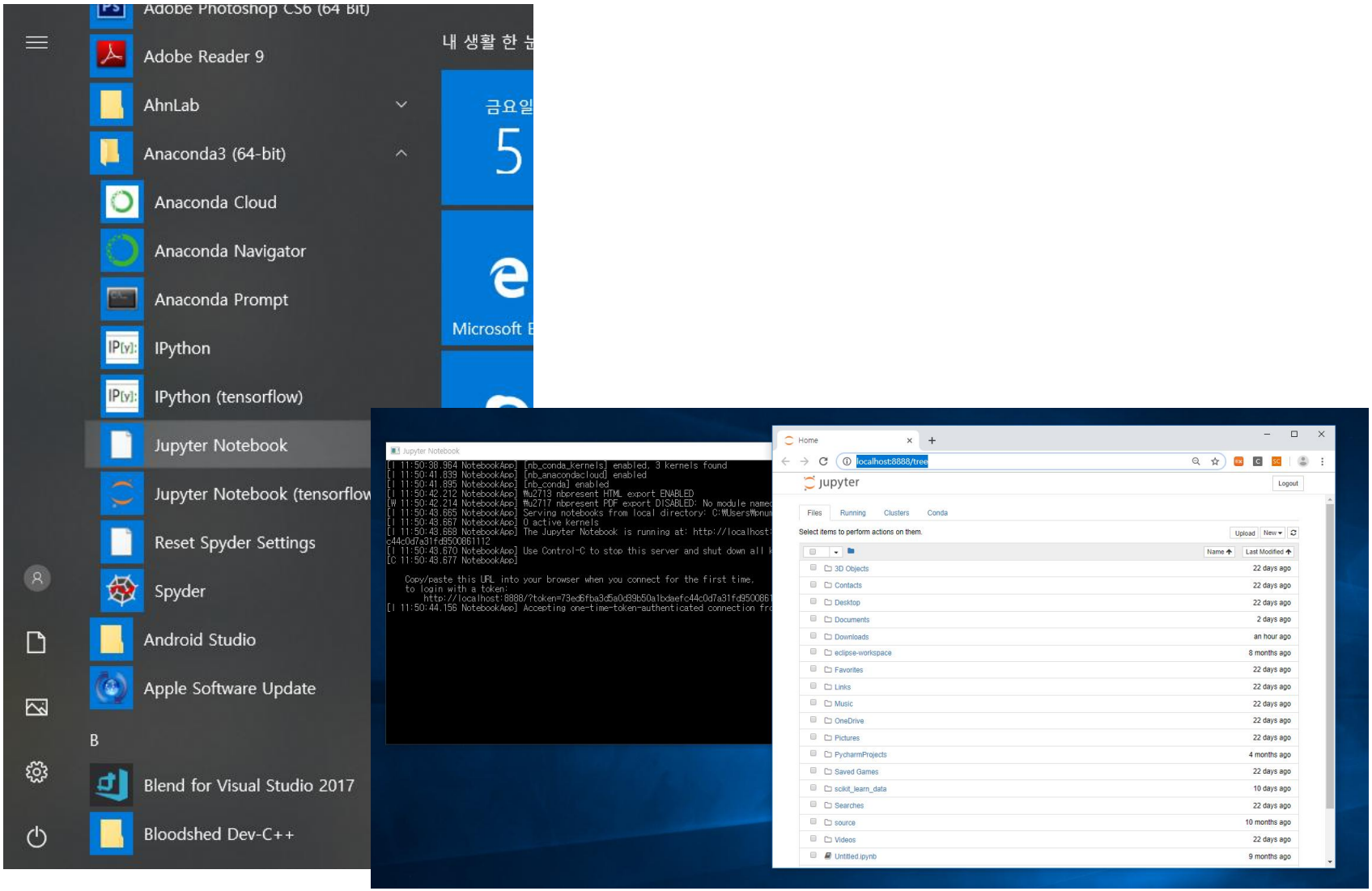
- 아나콘다
  - Python 기본 패키지에 각종 수학/과학 라이브러리들을 같이 패키징해서 배포하는 버전
- 아나콘다 설치
  - https://www.anaconda.com



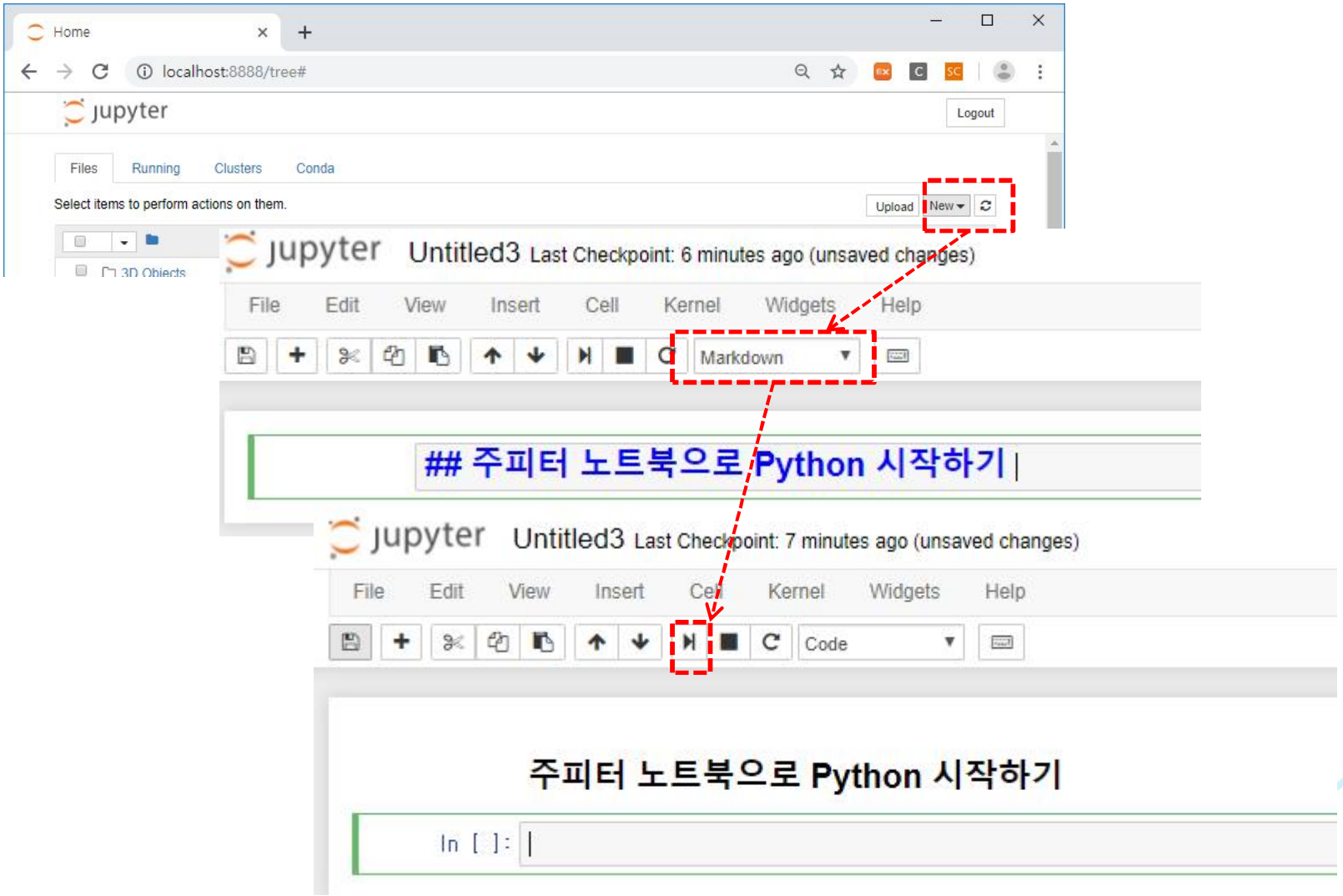
주피터 노트북(Jupyter Notebook)

- **웹 브라우저에서 코드를 작성하고 실행 가능한 툴**
  - **데이터 시각화**
    - 시각화를 작성하고 공유할 수 있으며 공유된 코드 및 데이터 모음에 대화형 수정도 가능
  - **코드 공유**
    - 웹 브라우저에서 직접 코드를 확인, 실행하고 결과를 표시할 수 있음
  - **코드와의 실시간 대화.**
    - 브라우저에서 직접 제공되는 피드백을 반영해 실시간으로 조금씩 편집하여 다시 실행
  - **코드 샘플 기록**
    - 설명과 함께 대화 기능을 추가할 수 있음

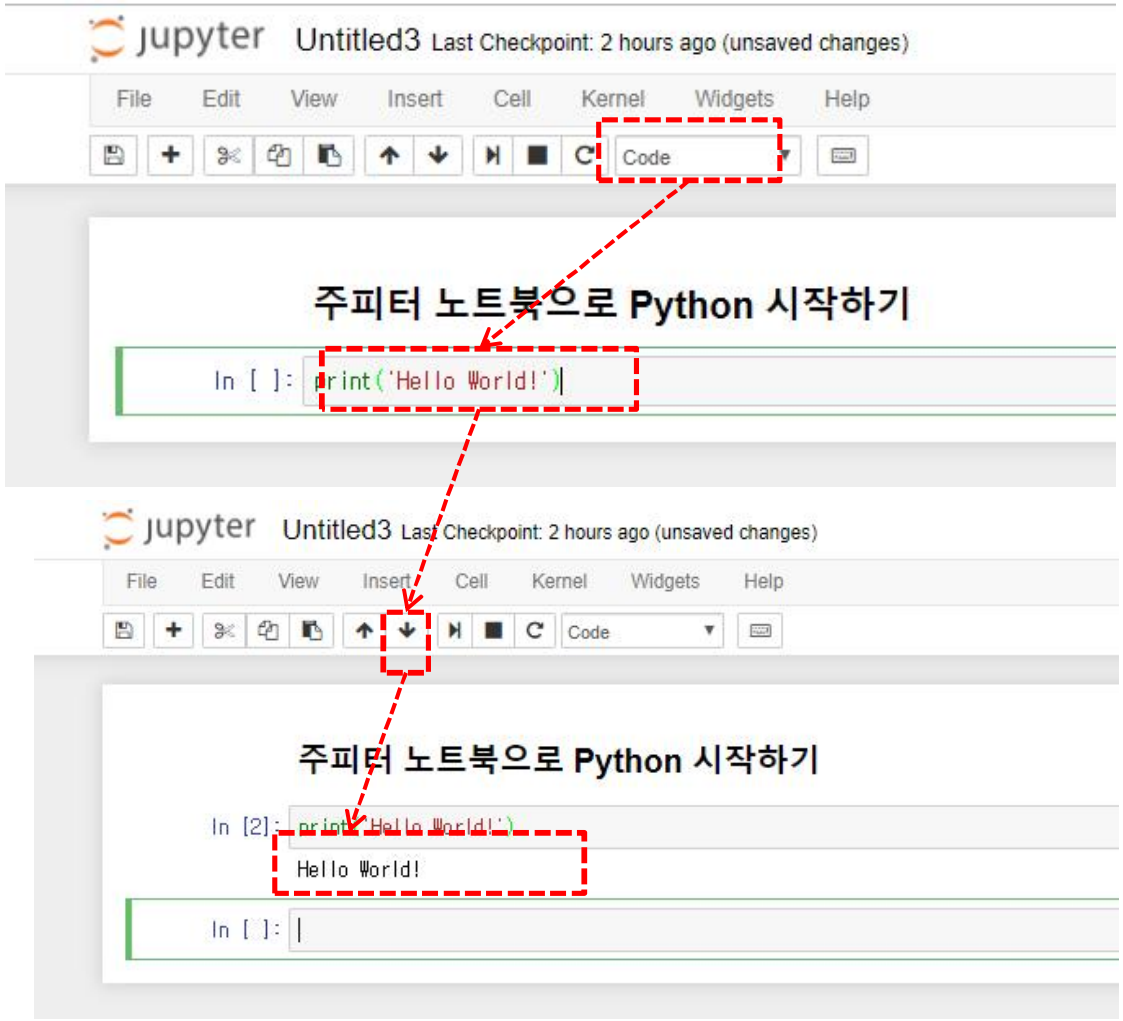
주피터 노트북 실행


Adobe Photoshop CS6 (64 Bit)
Adobe Reader 9
AhnLab
Anaconda3 (64-bit)
Anaconda Cloud
Anaconda Navigator
Anaconda Prompt
IPython
IPython (tensorflow)
Jupyter Notebook
Jupyter Notebook (tensorflow)
Reset Spyder Settings
Spyder
Android Studio
Apple Software Update
B
Blend for Visual Studio 2017
Bloodshed Dev-C++
Jupyter Notebook
Home
localhost:8888/tree
jupyter
Logout
Files
Running
Clusters
Conda
Select items to perform actions on them.
Upload
New
Name
Last Modified
3D Objects
Contacts
Desktop
Documents
Downloads
eclipse-workspace
Favorites
Links
Music
OneDrive
Pictures
PycharmProjects
Saved Games
scikit_learn_data
Searches
source
Videos
Untitled.ipynb

Python 노트북 만들기-Markdown



Python 노트북 만들기-Code



Python 기본 문법(1)

Python 표준출력

- **출력함수 : print()**

```
#표준 출력
print('Hello')
print('World!')

#한줄 연결 출력
print('Hello', end='')
print('World!')

print('Hello', 'World!')
print('Hello' + 'World!')
```

한 줄 주석처리 : #
여러 줄 주석처리 : ''' ... '''

```
Hello
World!
HelloWorld!
Hello World!
HelloWorld!
```

Python 표준입력

- **입력 함수 : input()**
  - 사용자가 어떤 값을 입력하게 하고, 그 값을 변수에 저장
  - **input("메시지")**
    - 문자열 입력

```
#표준 입력
x = input()

x = input('x 입력 :')
y = input('y 입력 :')

#표준 출력
print(x , '+', y, '=', x+y)
```

```
4
x 입력 :10
y 입력 :5
10 + 5 = 105
```

+ 연산자 (결합연산자)

Python 표준입력

- **입력 함수 : input()**
  - 정수 입력 : int(input("메시지"))
  - 실수 입력 : float(input("메시지"))

변수

```
#표준 입력
x = int(input('x 입력 :'))
y = int(input('y 입력 :'))

#표준 출력
print(x , '+', y, '=', x+y)
```

```
x 입력 :10
y 입력 :5
10 + 5 = 15
```

해결문제

- 명함을 작성하시오.

```
소속을 입력하세요부산대학교
성명을 입력하세요김경민
연락처를 입력하세요010-1234-1234
이메일 주소를 입력하세요.pnu@pusan.ac.kr
+--------------------------------+
    부산대학교
    김경민
    010-1234-1234
    pnu@pusan.ac.kr
+--------------------------------+
```

변수

- 어떤 값을 저장하는 공간
  - 어떤 데이터가 있을 때 그 데이터가 메모리 상에 위치하는 주소를 변수라는 곳에 저장해두고, 나중에 변수에 저장된 메모리 상의 주소에 가서 실제 값을 읽을 수 있는 것
- Python은 기본적으로 인터프리터(interpreter)를 통해 실행되는 스크립트 언어
  - 변수를 선언할 때 타입을 지정하지 않음
  - 값을 할당하면 그때 동적으로 타입이 정해짐
  - 타입이 다를 경우 형 변환을 해줘야 연산에 사용할 수 있음

변수

- **변수 명 규칙**
  - 영문 문자와 숫자 사용
    - 대소문자 구분
  - 문자나 _(밑줄 문자)로 시작
    - 숫자부터 시작하면 안 됨
  - 특수 문자(+, -, *, /, $, @, &, % 등)는 사용할 수 없음
  - python 키워드(if, for, while, and, or 등)는 사용할 수 없음

- **변수 만들기**

```
변수명 = 변수에 저장할 값
```

자료형

```
#정수형
x = 10
print('x=', x, ':' , type(x))

#실수형
y = 20.5
print('y=', y, ':' , type(y))

#부울형
b = True
print('b=', b, ':' , type(b))

#문자열
s = 'Python'
print('s=', s, ':' , type(s))

#리스트
lst = [1, 'a']
print('lst=', lst, ':' , type(lst))

#튜플
t = (1, 'a')
print('t=', t, ':' , type(t))

#딕셔너리
d = { 'a': 1, 'b':2}
print('d=', d, ':' , type(d))
```

```
x= 10 : <class 'int'>
y= 20.5 : <class 'float'>
b= True : <class 'bool'>
s= Python : <class 'str'>
lst= [1, 'a'] : <class 'list'>
t= (1, 'a') : <class 'tuple'>
d= {'a': 1, 'b': 2} : <class 'dict'>
```

```
#형변환
x = 10
print('x=', x, ':' , type(x))

x = float(x)
print('x=', x, ':' , type(x))

x = str(x)
print('x=', x, ':' , type(x))

# +연산자가 결합 연산자로 사용될 경우 숫자와 문자를 더할 수 없음
y = 20
print('y = ' + str(y))
print('y = ' + y)
```

```
x= 10 : <class 'int'>
x= 10.0 : <class 'float'>
x= 10.0 : <class 'str'>
y = 20
```

```
TypeError                                 Traceback (most recent call last)
<ipython-input-17-7b60e179f3a4> in <module>()
     12 y = 20
     13 print('y = ' + str(y))
---> 14 print('y = ' + y)

TypeError: Can't convert 'int' object to str implicitly
```

문자열

```
#문자열 선언
str = "No pain no gain."
print(str)

#이스케이프 문자를 활용한 줄바꿈
str2 = "₩nNo pain ₩nno gain.₩n"
print(str2)

#문자열 연산
str3 = str + "₩n" + str ;
print('더하기 연산 : ', str3)
str4 = str* 3 ;
print('곱하기 연산: ', str4)


print(str, '₩n')

#문자열 길이
print("문자열 길이 : ", len(str), '₩n')

#문자열 인덱싱
print("문자열 인덱싱(첫번째 문자) :" , str[0] )
print("문자열 인덱싱(마지막 문자) :" , str[-1],'₩n')

#문자열 슬라이싱 : 끝번호 포함하지 않음
print('문자열 슬라이싱 [3:4] : ', str[3:4])
#문자열 슬라이싱: 시작번호 생략하면 처음부터 추출
print('문자열 슬라이싱 [:2] : ', str[:2])
#문자열 슬라이싱: 끝번호 생략하면 끝까지 추출
print('문자열 슬라이싱 [2:] : ', str[2:])
#문자열 슬라이싱: 시작번호 끝번호 생략하면 전체 추출
print('문자열 슬라이싱 [:] : ', str[:],'₩n')

#format함수를 이용한 문자열 포매팅
str5 = '{0} pain {1} gain'
str5 = str5.format('No', 'no')
print(str5)
```

```
No pain no gain.

No pain
no gain.

더하기 연산 :  No pain no gain.
No pain no gain.
곱하기 연산:  No pain no gain.No pain no gain.No pain no gain.
No pain no gain.

문자열 길이 :  16

문자열 인덱싱(첫번째 문자) : N
문자열 인덱싱(마지막 문자) : .

문자열 슬라이싱 [3:4] :  p
문자열 슬라이싱 [:2] :  No
문자열 슬라이싱 [2:] :   pain no gain.
문자열 슬라이싱 [:] :  No pain no gain.

No pain no gain
```

문자열 메소드

```
str = "No pain no gain."
print('문자열 = ' , str)


#지정 문자 개수
print('ain 개수 : ', str.count('ain') )

#문자 위치 찾기
print('a 위치 :', str.find('a'))
print('a 위치 :', str.index('a'),'₩n') #없으면 오류

#대소문자 변경
str = str.upper()
print("대문자 변경(upper) :" , str)

str = str.lower()
print("소문자 변경(lower) :" , str)

str = str.capitalize()
print("첫문자만 대문자(capitalize) :" , str)

str = str.title()
print("단어 첫글자 대문자 변경(title) :" , str)

str = str.swapcase()
print("대문자는 소문자 소문자는 대문자(swapcase) :" , str,'₩n')

str = "No pain no gain."
#문자열 편집 치환
str2 = str.strip(str[2:])
print("양쪽 공백 제거 : :", str2)

str2 = str.strip('.')
print(". 제거 : :", str2)

str2 = str.replace(' ','')
print("문장내 공백 제거 : :", str2)

str2 = ",".join(str)
print("단어사이에 ,삽입  :", str2)

str2 = str.split(' ')
print("단어 분리 :" , str2)
```

```
문자열 =  No pain no gain.
ain 개수 :  2
a 위치 : 4
a 위치 : 4

대문자 변경(upper) : NO PAIN NO GAIN.
소문자 변경(lower) : no pain no gain.
첫문자만 대문자(capitalize) : No pain no gain.
단어 첫글자 대문자 변경(title) : No Pain No Gain.
대문자는 소문자 소문자는 대문자(swapcase) : nO pAIN nO gAIN.

양쪽 공백 제거 : : N
. 제거 : : No pain no gain
문장내 공백 제거 : : Nopainnogain.
단어사이에 ,삽입  : N,o, ,p,a,i,n, ,n,o, ,g,a,i,n,.
단어 분리 : ['No', 'pain', 'no', 'gain.']
```

해결문제

- 다음은 문재인 대통령의 연설문 일부 입니다.
  - 찾고자 하는 단어를 입력 받아서 해당하는 단어가 몇 번 사용되었는지 확인해 보세요.

국민 여러분의 위대한 선택에 머리 숙여 깊이 감사드립니다. 저는 오늘 대한민국 제19대 대통령으로서 새로운 대한민국을 향해 첫걸음을 내딛습니다. 지금 제 두 어깨 는 국민 여러분으로부터 부여받은 막중한 소명감으로 무겁습니다. 지금 제 가슴은 한번도 경험하지 못한 나라를 만들겠다는 열정으로 뜨겁습니다. 그리고 지금 제 머리는 통합과 공존의 새로운 세상을 열어갈 청사진으로 가득 차 있습니다.