

제 1 장. Python의 소개

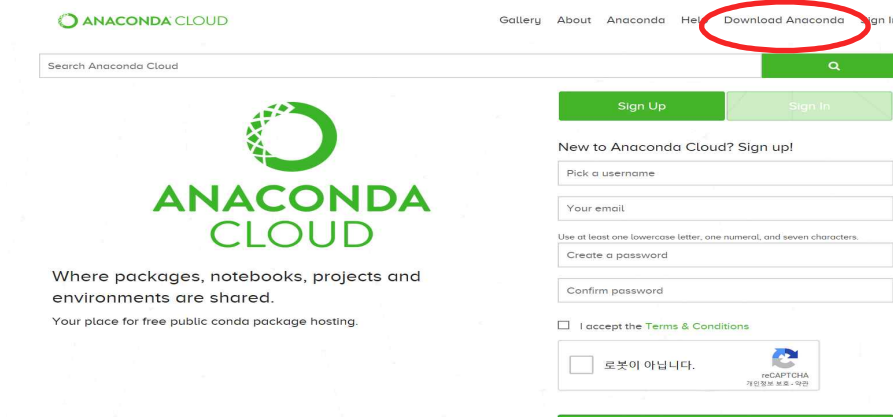
1.1 Python의 소개

- 파이썬은 인터프리터 방식의 객체 지향적 프로그래밍 언어이다
- 간결한 문법으로 입문자가 이해하기 쉽고, 다양한 분야에 활용 가능하다
- 파이썬은 컴퓨터 프로그래밍 교육을 위해 많이 사용하지만, 기업의 실무를 위해서도 많이 사용하는 언어로써 활용범위가 점점 넓어지고 있는 추세이다
- 무료로 설치하고 사용 가능하며, 머신러닝 및 딥러닝 등의 분야에서 특히 많이 사용된다

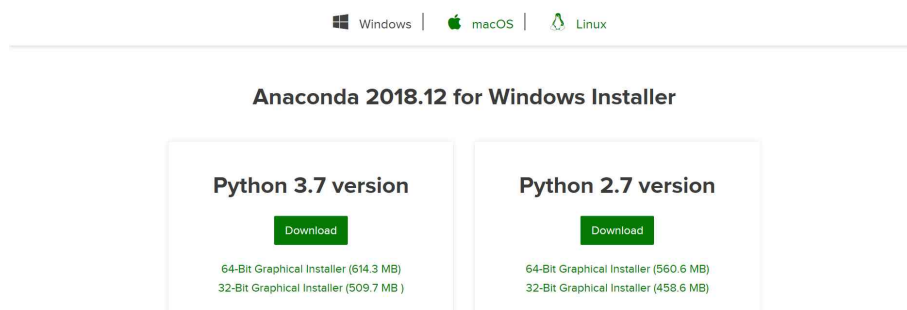
1.2 Python 설치하기

- Anaconda는 R과 파이썬 등을 이용한 데이터 분석에 필요한 오픈소스를 모아놓은 데이터 과학 플랫폼이다. 파이썬 공식 홈페이지(<https://www.python.org/>) 등의 다른 방법들로도 파이썬을 설치 할수 있지만, Anaconda를 이용하면 손쉽게 파이썬의 모든 데이터 과학 패키지를 설치, 관리할 수 있다.

- <https://anaconda.org/> 접속 후 우측 상단의 Download Anaconda를 클릭한다.



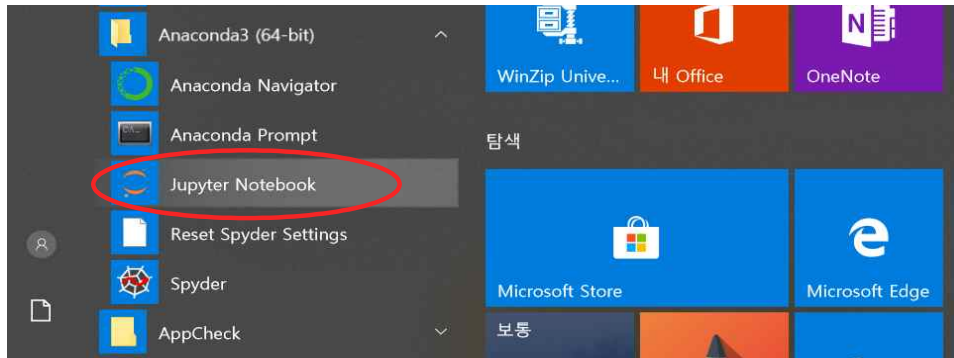
- 최신 버전의 Anaconda 다운로드 (2019년 2월 현재 3.7)



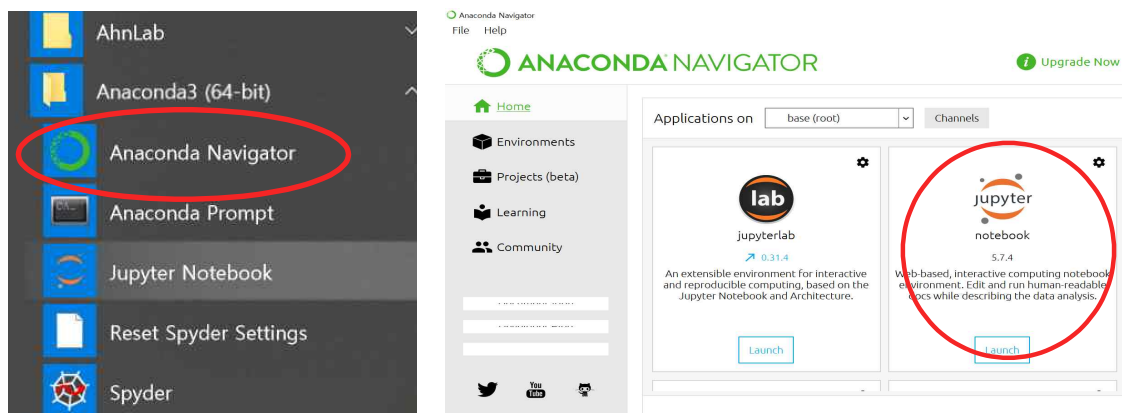
1.3 editor 실행하기

- Jupyter Notebook은 파이썬 에디터 중 하나로, 파이썬의 사용을 용이하게 한다. Jupyter Notebook 외에도 PyCharm, Spyder 등의 editor가 존재한다. 본 실습에서는 jupyter notebook을 사용하도록 한다.
- Jupyter Notebook 실행하는 여러 가지 방법은 다음과 같다.

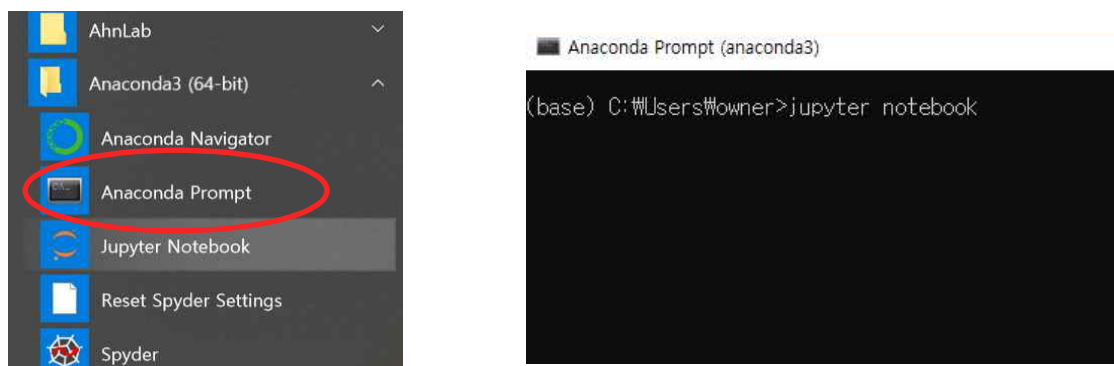
1) 윈도우 시작 화면에서 Jupyter Notebook 바로가기 클릭



2) Anaconda Navigator를 실행하여 Jupyter Notebook 바로가기 클릭



3) Anaconda Prompt에서 jupyter notebook 을 입력

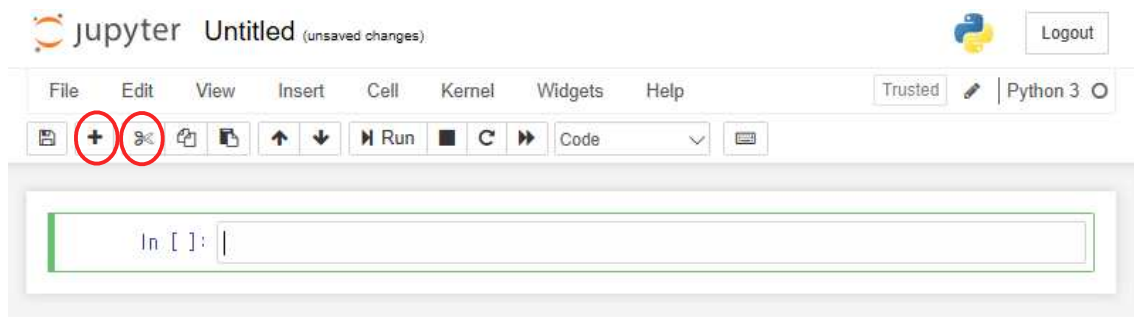


1.4 Jupyter Notebook 사용방법

Jupyter Notebook을 실행하면 나타나는 첫 화면에서 파일 저장을 원하는 폴더를 선택 할 수 있다. 폴더 선택 후, 오른쪽 위의 [New] - [Python 3] 클릭하면 새 파이썬 노트북이 실행된다.



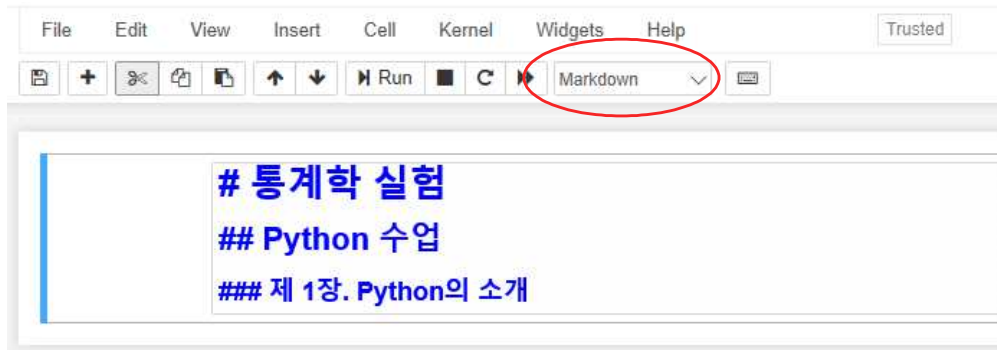
이제 아래 화면과 같이 상단에는 메뉴와 툴바들이 표시되고, 중간에 파이썬 코드를 넣을 수 있는 코드 셀이 표시된다. 새로운 코드 셀을 추가하기 위해서는 [+], 현재 선택된 코드 셀을 삭제하기 위해서는 가위표시 툴바를 누르면 된다.



코드 입력 방법은 다음과 같다. 첫 번째 코드 셀에 파이썬 코드를 넣고, [Run] 버튼을 누르면 코드가 실행되면서 결과가 코드 셀 하단에 출력되면서 In []:이 In [1]:로 바뀌게 된다. 이때의 1은 첫 번째 실행된 코드라는 뜻이며, 계속 다른 코드 셀을 실행할 때마다 이 번호가 증가하게 된다. 예를 들어, 다음과 같이 간단한 수식을 입력한 후 이를 실행해보자.



또한, 메뉴의 [Markdown]을 선택하면 코드에 대한 설명을 작성할 수 있다. 예를 들어 다음과 같이 입력한 후 실행 결과를 확인해보자.

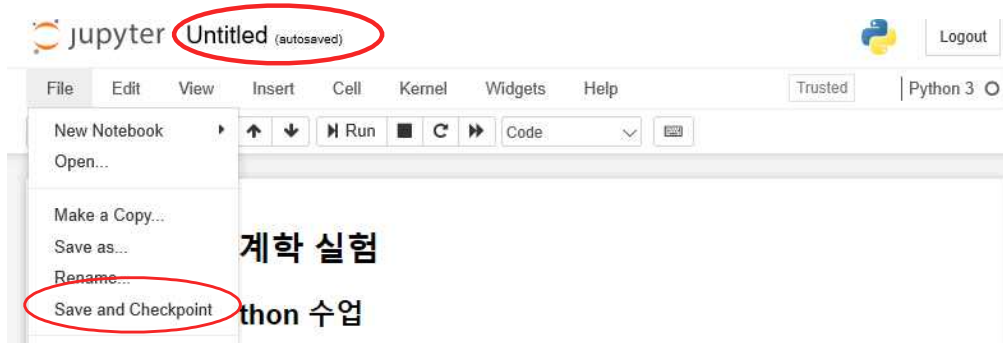


여기서 # 은 제목이라는 뜻이며 # 이 하나씩 늘어날수록 하위 제목이 된다. 결과는 다음과 같다.



이처럼 주피터 노트북은 코드, 코드의 실행 결과, 코드에 대한 설명을 한 번에 작성할 수 있어서 체계적인 기록이 가능하다. 특히 하나의 노트북 안에 들어있는 코드 셀은 모두 실행 상태가 연결되기 때문에 코드 셀에서 변수를 만들면 다른 코드 셀에서 해당 변수를 사용할 수 있다. 이때 코드 셀의 실행 순서는 In []:에 표시된 숫자 순서를 따르며 메뉴의 ↑↓ 버튼을 이용하여 코드 셀의 위치를 바꿀 수도 있다. 그러나 코드 셀의 위치를 바꾸더라도 실행 순서는 In []:의 숫자를 따르게 된다.

노트북의 Jupyter 로고 옆의 Untitled를 클릭하면 노트북의 제목을 변경할 수 있다. 그리고 메뉴의 [File] > [Save and Checkpoint]를 클릭하면 노트북이 *.ipynb 파일로 저장된다.



파이썬의 실행에 관련된 몇 개의 단축키를 소개한다.

Shift + Enter	해당 cell을 실행하고, 그 아래 cell을 선택한다.
Ctrl + Enter	해당 셀을 실행한다
Alt + Enter	해당 셀을 실행하고, 그 아래에 새로운 cell 하나를 추가한다.

작성된 코드를 실행할 때 주로 [Shift + Enter]을 사용하고, 자신이 코드를 작성할 경우 [Alt+Enter]을 주로 사용한다.

코드셀을 선택한 후 다음과 같은 단축키를 사용하여 좀 더 편리하게 노트북을 사용할 수 있다.

a	현재 cell 바로 위에 cell 하나를 추가한다
b	현재 cell 바로 아래에 cell 하나를 추가한다
x	cell 잘라내기
c	cell 복사하기
v	cell 현재 cell의 아래에 붙여넣기
d d	셀 삭제

1.5 Python의 변수 생성 및 자료의 형태

앞서 살펴본 바와 같이, 간단한 계산 및 출력 등을 위해서는 직접 표현식을 입력하고 실행할 수 있다. 하지만 이렇게 실행된 계산 결과는 결과물의 출력만 이루어 질 뿐 변수로써 저장되지 않는다. 따라서 자료를 변수에 입력하여 저장하는 작업이 필요하다.

파이썬의 변수명은 알파벳, 숫자, ‘_’로 구성되며 첫 글자는 문자로 시작해야 한다. 또한 이름 중간에는 공백을 가질 수 없으며 대소문자를 구별한다. 변수에 값을 할당할 때는 ‘=’를 사용한다.

```
In [1]: 3+5
Out[1]: 8

In [2]: a=3+5
        a+3
Out[2]: 11
```

1.5.1 숫자형 자료

숫자형 자료의 입력과 계산에 대한 간단한 예제를 보자. 숫자형 변수를 생성한 후, 두 변수의 연산 결과 역시 다른 변수로 저장이 가능하다. 하나의 명령어는 한 줄에 입력되는 것이 기본이지만 세미콜론(;)을 사용하여 한 줄에서 명령어를 나눌 수도 있다. print() 함수를 사용하여 변수의 값을 출력 할 수 있다.

```
In [1]: a=5
        b=3

        c=a+b

        print(a);print(b)
        print(c)

5
3
8
```

파이썬에서는 다음과 같이 여러 개의 변수를 동시에 만들 수도 있다. 단, 변수의 개수와 값의 개수가 동일해야 하며 나열한 순서대로 값이 할당 된다.

```
In [5]: a,b,c = 10,20,30
        print(a); print(b); print(c)

        print(a,b,c)

10
20
30
10 20 30
```

다양한 함수의 사용에 대해서는 3장에서 좀 더 자세히 다루도록 한다.

1.5.2 문자형 자료

문자형 자료의 입력은 작은 따옴표(' ') 또는 큰 따옴표(" ")를 사용한다.

```
In [2]: d = 'python' ; print(d)

        e = "python" ; print(e)

        f = " 'python' " ; print(f)

python
python
'python'
```

1.5.3. 논리형 자료

논리형 자료는 참(True) 또는 거짓(False)을 표현한다. 파이썬에서는 비교 연산자와 논리 연산자의 판단 결과로 참 또는 거짓을 사용한다. 3장에서 좀 더 자세히 다루도록 한다.

```
In [8]: a = 1
        b = a==1 # 비교를 위한 연산자

        print(b)
        print(a>2)

True
False
```

1.5.4 리스트

리스트는 여러 개의 값을 일렬로 늘어놓아 묶어 놓은 집합체이다. 변수에 값을 저장할 때, 여러 개의 값을 대괄호([])로 묶어주면 리스트가 되며 각 값은 comma(,) 로 구분할 수 있다. 리스트에 저장된 각각의 값은 요소(element)라고 부르며 리스트는 숫자형, 문자형, 논리형 등 모든 자료형을 저장할 수 있으며 자료형을 섞어서 저장할 수도 있다.

```
In [9]: age = [22, 23, 24, 25]
name = ['kim', 'lee', 'park']
TF = [True, False, True]
print(age, name, TF)

m = [1, 'ok', False]
print(m)

[22, 23, 24, 25] ['kim', 'lee', 'park'] [True, False, True]
[1, 'ok', False]
```