

산업 인공지능 - 실습 1

Python 프로그래밍 환경

2021 Spring

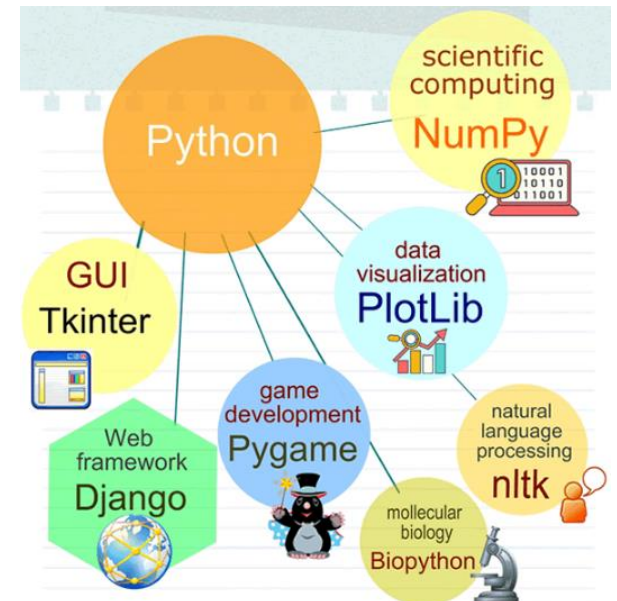
Python

❖ Python 언어

- 1991년 귀도 반 로섬(Guido van Rossum) 개발
- 대화형 프로그래밍 언어



- 높은 생산성
- 초보자한테 쉬운 언어 – 인터프리터 언어
- 다양한 플랫폼 지원
- 풍부한 라이브러리
- 애니메이션, 그래픽 지원 용이

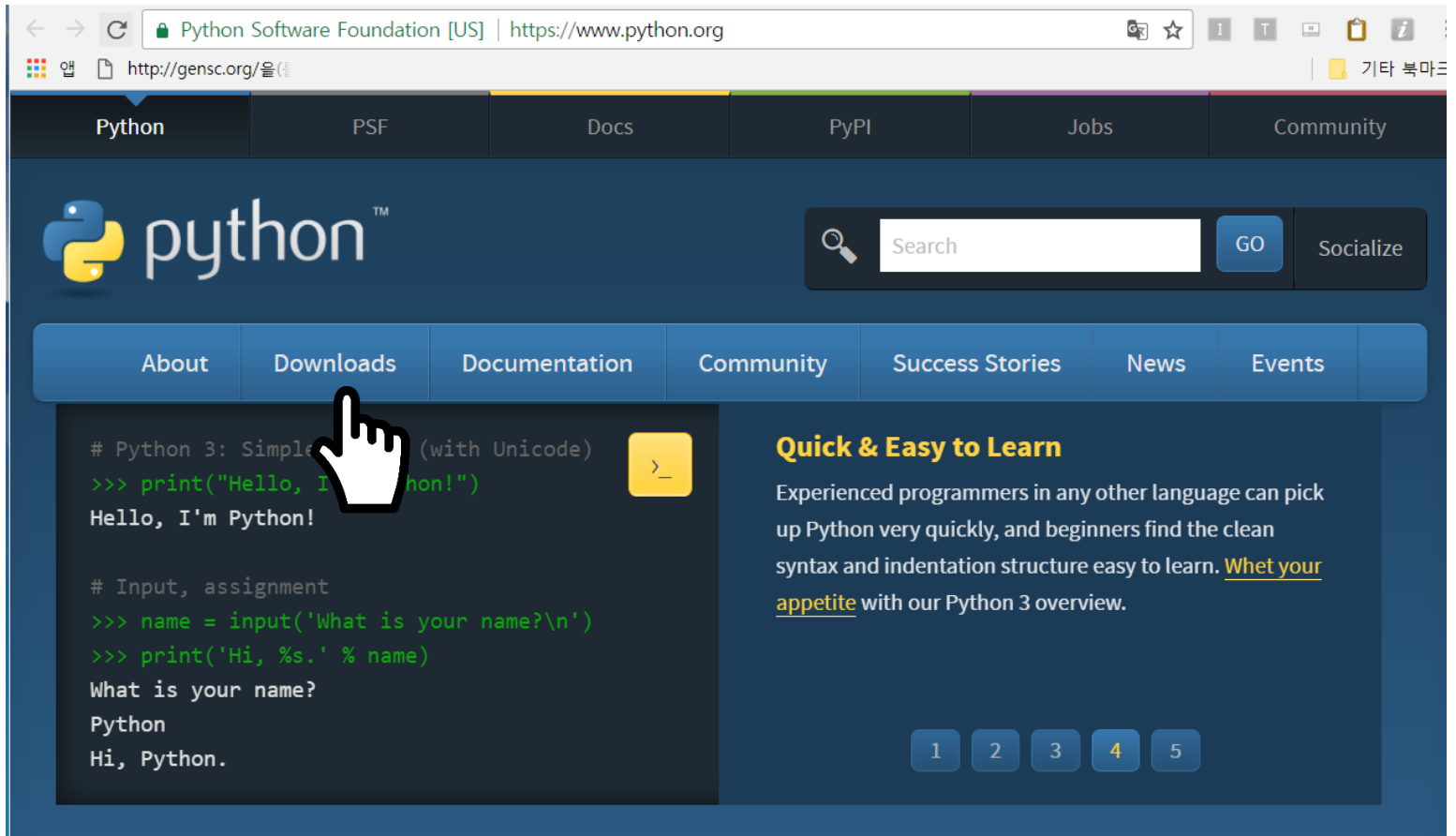


Python 프로그램 개발환경

- ❖ Standalone Python 개발환경
- ❖ Anaconda 개발환경
- ❖ Colab 개발환경

Python 설치

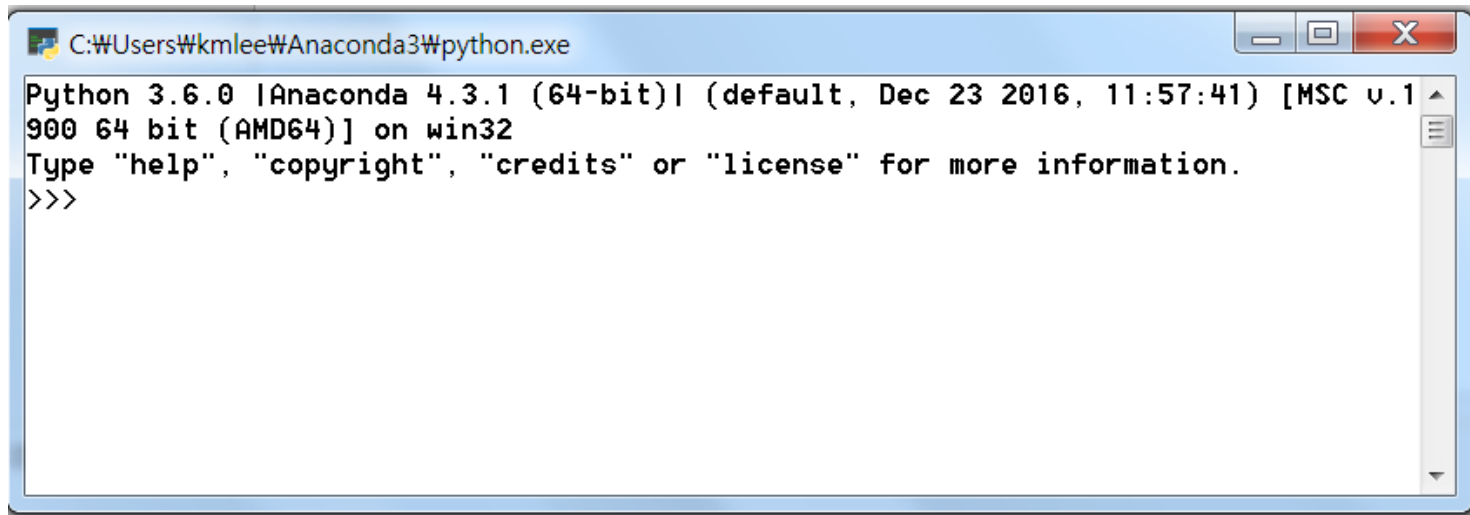
- ❖ <https://www.python.org/>
 - 최신 버전(Python3.9.2) 설치



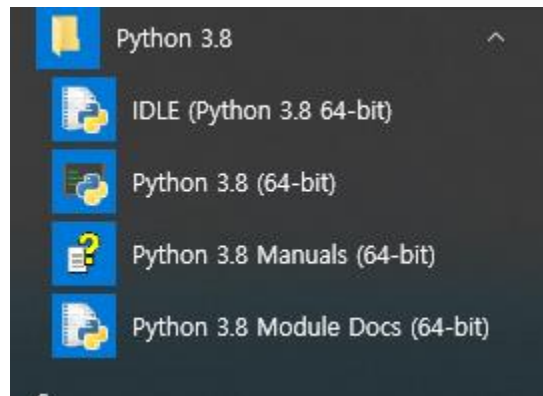
Python 설치

❖ Python shell

- >>> 뒤에 우리가 명령어를 입력하고 엔터키를 누르면 명령어가 실행되고 실행 결과가 화면에 출력



A screenshot of a Windows command prompt window titled "C:\Users\Wkmlee\Anaconda3\python.exe". The window shows the following text: "Python 3.6.0 |Anaconda 4.3.1 (64-bit)| (default, Dec 23 2016, 11:57:41) [MSC v.1900 64 bit (AMD64)] on win32", "Type 'help', 'copyright', 'credits' or 'license' for more information.", and the prompt ">>>".



Python IDE

❖ IDE (Integrated Development Environment, 통합 개발 환경)

```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
```

```
Untitled
File Edit Format Run Options Window Help
|
Ln: 1 Col: 0
```

```
ex1.py - C:/Users/CBNU/Google 드라이브/2021 I-AI/프로그램/프로그램/2단원_0822/2-1/ex1.py (3....
File Edit Format Run Options Window Help
a = [242, 256, 237, 223, 263, 81, 46]

# -----
# 그래프를 그리기 위한 외부 모듈 선언
import matplotlib.pyplot as plt
from matplotlib import font_manager, rc

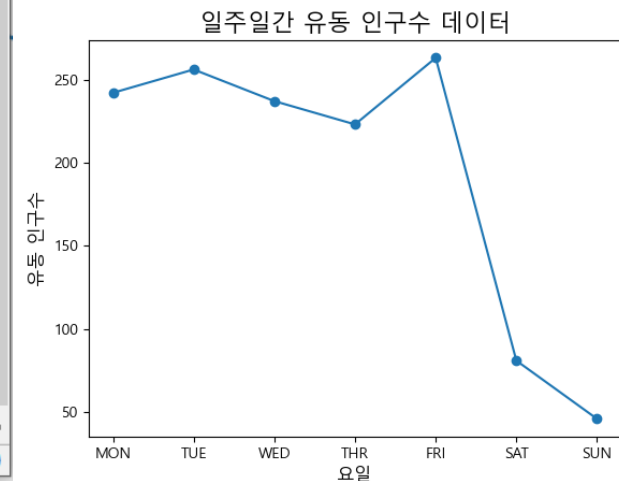
#한글을 출력하기 위한 폰트 로딩
font_name = font_manager.FontProperties(fname="c:/Windows/Fonts/malgun.ttf").get_name()
rc('font', family=font_name)

x_data = ['MON', 'TUE', 'WED', 'THR', 'FRI', 'SAT', 'SUN']

# 그래프의 제목 붙이기
plt.title("일주일간 유동 인구수 데이터", fontsize = 16)
plt.xlabel("요일", fontsize=12)
plt.ylabel("유동 인구수", fontsize=12)

plt.scatter(x_data, a)
plt.plot(x_data, a)
plt.show()

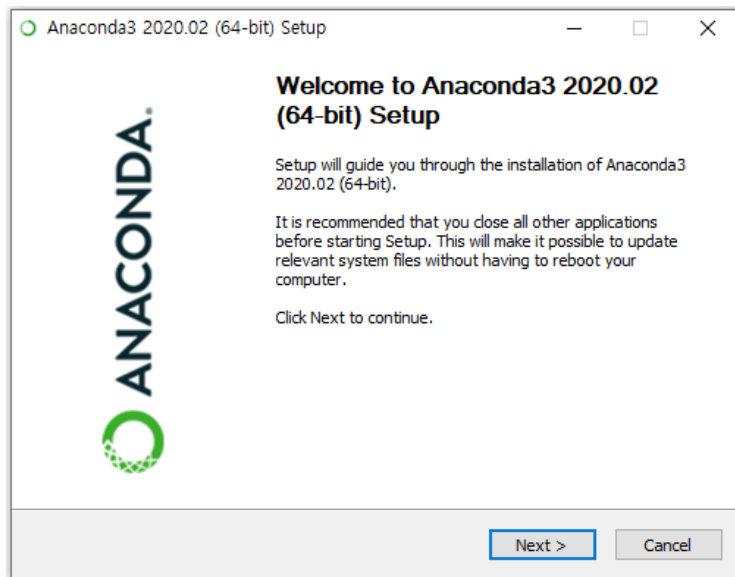
Ln: 23 Col: 0
```



Anaconda

❖ Anaconda 설치

- <https://www.anaconda.com/distribution/#download-section>



Windows 

Python 3.7

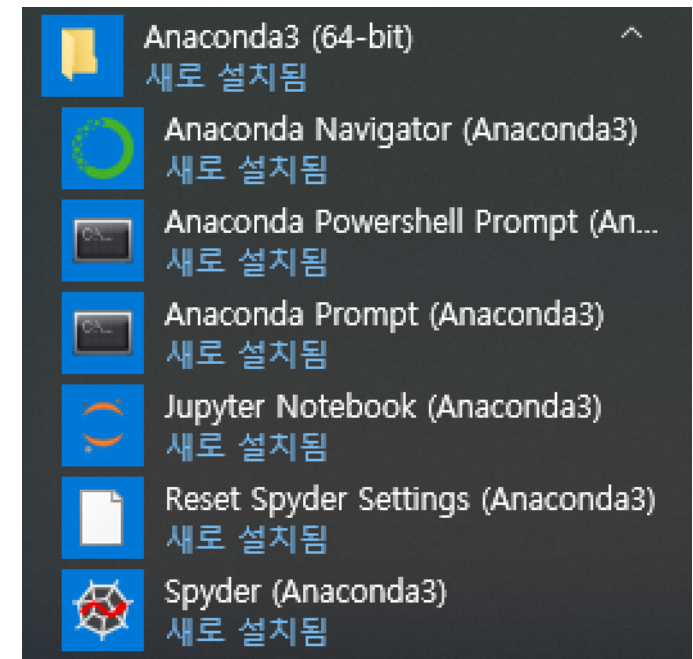
64-Bit Graphical Installer (466 MB)

32-Bit Graphical Installer (423 MB)

Python 2.7

64-Bit Graphical Installer (413 MB)

32-Bit Graphical Installer (356 MB)

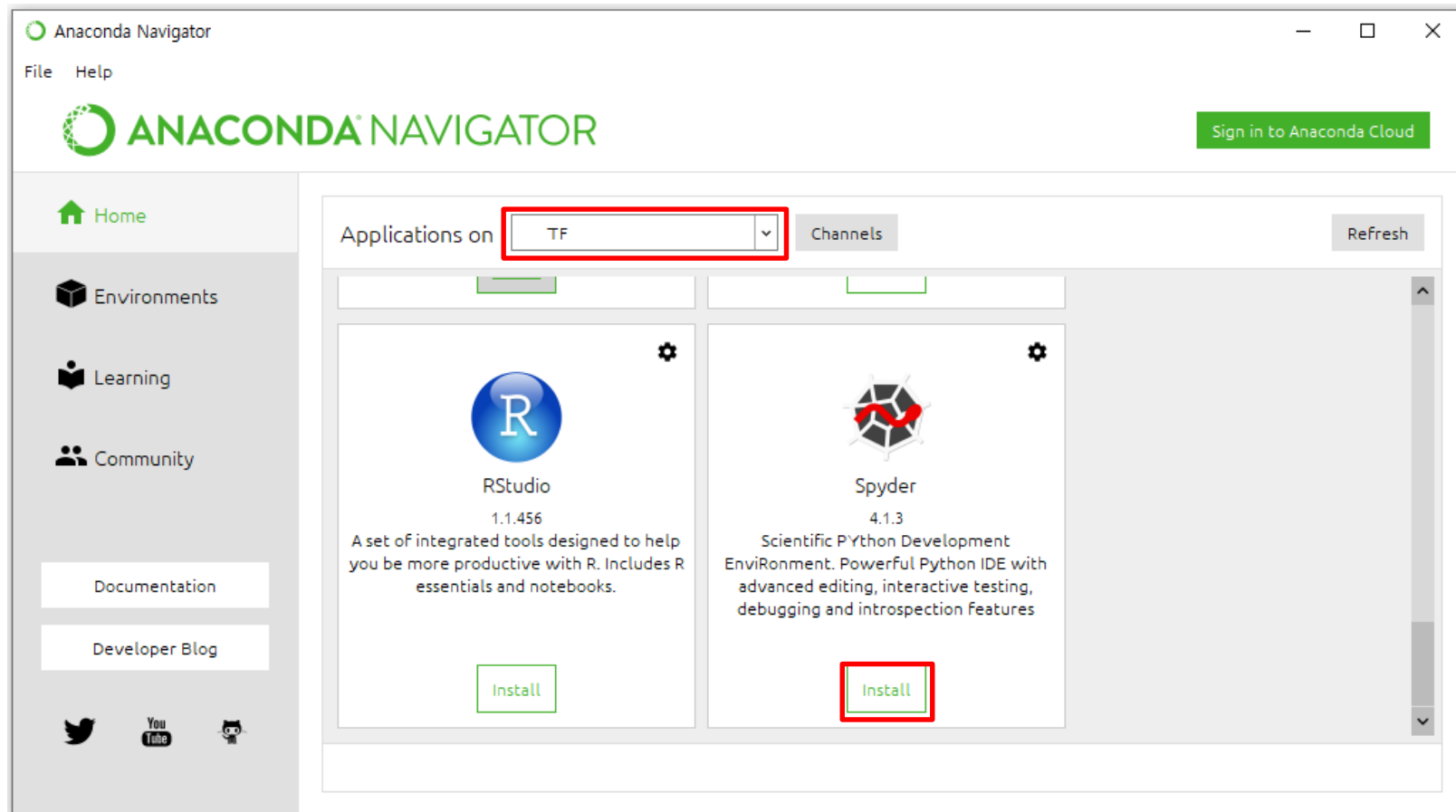


Anaconda

❖ Anaconda 설정

- 가상 환경 생성

```
C:\> 선택 명령 프롬프트
C:\Users\kml>conda create -n TF pip python=3.7
Collecting package metadata (current_repodata.json): done
Solving environment: done
```



Jupyter (Anaconda)

jupyter Untitled Last Checkpoint: 4분 전 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Code

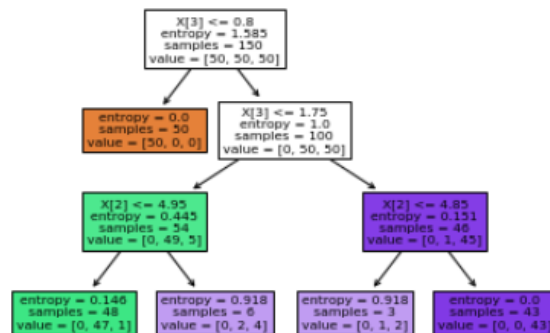
```
In [1]: import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier, plot_tree
```

```
In [2]: iris = load_iris()
```

```
In [3]: decision_tree = DecisionTreeClassifier(criterion="entropy", random_state=0, max_depth=3)
```

```
In [4]: decision_tree = decision_tree.fit(iris.data, iris.target)
```

```
In [5]: plt.figure()
plot_tree(decision_tree, filled=True)
plt.show()
```



<https://colab.research.google.com/>

❖ Google account 필요

- 무료 가입 -> 로그인 후 해당 웹페이지 접속

← → ↻ colab.research.google.com/drive/13MMEFa62Lbr8L52Rkfz4SB1tdspj5C4#scrollTo=c9mYLIGFpYLK

Untitled58.ipynb ☆

파일 수정 보기 삽입 런타임 도구 도움말 모든 변경사항이 저장됨

+ 코드 + 텍스트

```
[1] 1 import matplotlib.pyplot as plt
    2 from sklearn.datasets import load_iris
    3 from sklearn.tree import DecisionTreeClassifier, plot_tree
```

```
[2] 1 iris = load_iris()
    2 decision_tree = DecisionTreeClassifier(criterion="entropy", random_state=0, max_depth=3)
    3 decision_tree = decision_tree.fit(iris.data, iris.target)
```

```
▶ 1 plt.figure()
   2 plot_tree(decision_tree, filled=True)
   3 plt.show()
```

```
graph TD
    Root["X[3] <= 0.8  
entropy = 1.585  
samples = 150  
value = [50, 50, 50]"]
    Root --> Left["entropy = 0.0  
samples = 50  
value = [50, 0, 0]"]
    Root --> Right["X[3] <= 1.75  
entropy = 1.0  
samples = 100  
value = [0, 50, 50]"]
    Left --> LeftLeft["X[2] <= 4.95  
entropy = 0.445  
samples = 54  
value = [0, 49, 5]"]
    LeftLeft --> LeftLeftLeft["entropy = 0.146  
samples = 48  
value = [0, 47, 1]"]
    LeftLeftLeft --> LeftLeftLeftLeft["entropy = 0.918  
samples = 6  
value = [0, 2, 4]"]
    Right --> RightLeft["X[2] <= 4.85  
entropy = 0.151  
samples = 46  
value = [0, 1, 45]"]
    RightLeft --> RightLeftLeft["entropy = 0.918  
samples = 3  
value = [0, 1, 2]"]
    RightLeftLeft --> RightLeftLeftLeft["entropy = 0.0  
samples = 43  
value = [0, 0, 43]"]
```

예

최근 사용

Google Drive

GitHub

업로드

노트 필터링



제목

처음 연 시간

마지막 연 시간



Colaboratory에 오신 것을 환영합니다

2020년 5월 17일 0분 전



Untitled58.ipynb

1일 전

1일 전



graphneuralnets.ipynb

2021년 2월 5일

2021년 2월 5일



Untitled57.ipynb

2021년 1월 29일

2021년 1월 29일



Untitled56.ipynb

2021년 1월 10일

2021년 1월 10일



새 노트

취소

예제 : 파티 준비

- ❖ 참석자에 맞추어서 치킨(1인당 1마리), 맥주(1인당 2캔), 케익(1인당 4개)를 출력하는 프로그램을 작성해보자.

참석자의 수를 입력하시오:25

치킨의 수: 25

맥주의 수: 50

케익의 수: 100



```
number = int(input("참석자의 수를 입력하시오:"))
chickens = number
beers = number*2
cakes = number*4
print("치킨의 수: ", chickens)
print("맥주의 수: ", beers)
print("케익의 수: ", cakes)
```



Untitled59.ipynb ☆

댓글

공유



파일 수정 보기 삽입 런타임 도구 도움말 저장 중...

+ 코드 + 텍스트

✓ RAM 디스크

수정 가능

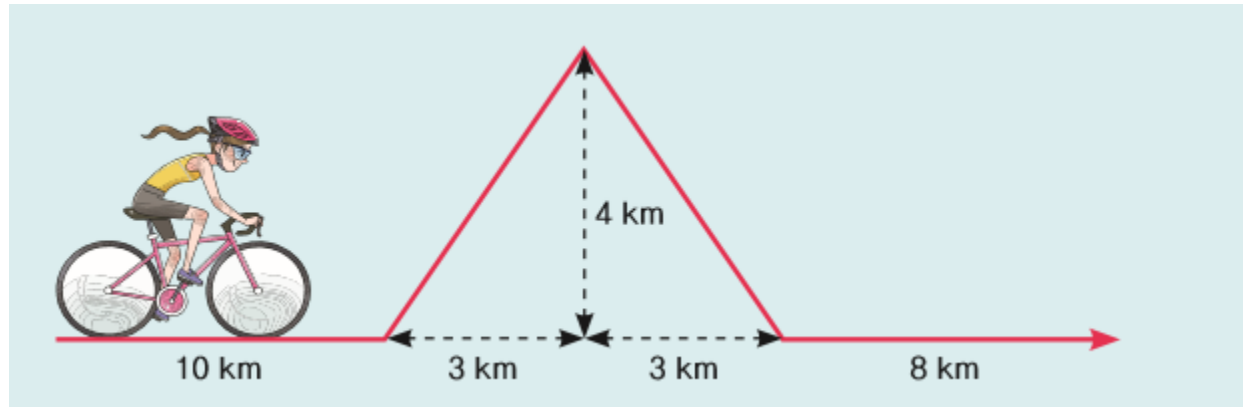


```
1 number = int(input("참석자의 수를 입력하시오:"))
2 chickens = number
3 beers = number*2
4 cakes = number*4
5 print("치킨의 수: ", chickens)
6 print("맥주의 수: ", beers)
7 print("케익의 수: ", cakes)
```

참석자의 수를 입력하시오:5
치킨의 수: 5
맥주의 수: 10
케익의 수: 20

예제: 등산 시간 계산

- ❖ 어떤 사람이 산악 자전거로 등산을 계획하고 있다. 평지에서는 시속 **20km/h**가 가능 하고 오르막에서는 **10km/h**, 내리막에서는 **30km/h**가 가능하다고 하자. 아래 경로를 자전거로 주행한다면 시간이 얼마나 걸릴까?



```
from math import *  
  
time1 = 10/20  
hill = sqrt(3**2+4**2)  
time2 = hill/10  
time3 = hill/30  
time4 = 8/20  
total = time1+time2+time3+time4  
print(total)
```

리스트

❖ 리스트(list)

- 여러 개의 값을 모아서 하나의 변수에 저장
- 인덱스는 0부터 시작

shopping_list = ['milk', 'eggs', 'cheese', 'butter', 'cream']

항목 #0 항목 #1 항목 #2 항목 #3 항목 #4

0	milk
1	eggs
2	cheese
3	butter
4	cream

리스트의 인덱스는 항상 0부터 시작합니다. 인덱스를 시작위치에서의 오프셋으로 생각하세요!

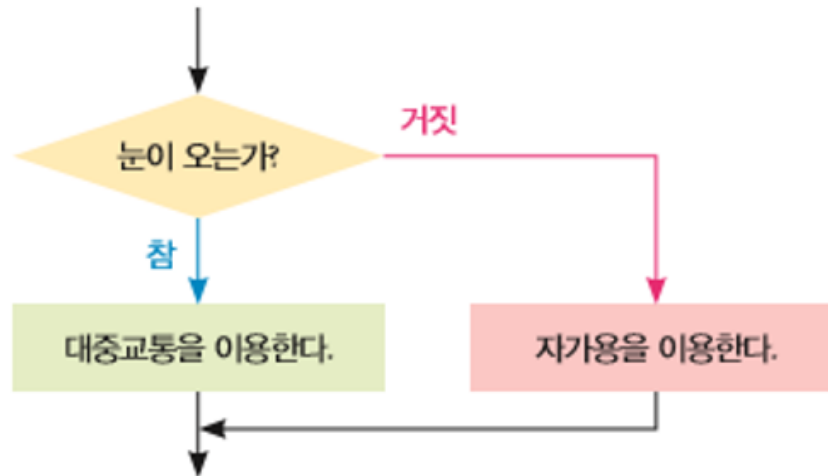


```
shopping_list = ['milk', 'eggs', 'cheese', 'butter', 'cream']
print(shopping_list)      # ['milk', 'eggs', 'cheese', 'butter', 'cream']
print(shopping_list[2])   # cheese
shopping_list[2]='apple'
print(shopping_list)      # ['milk', 'eggs', 'apple', 'butter', 'cream']
```

조건문

❖ 조건문 (conditional statement)

- 조건의 만족 여부에 따라 실행되는 부분이 결정되는 문장



```
if (눈이 내린다) :  
    대중교통을 이용한다  
else :  
    자가용을 이용한다
```


조건문

❖ 조건문의 구조

- 콜론(:) : 문장이 아직 끝나지 않음 표시
- 실행문장 들어쓰기



예제: 수하물 비용 계산

- ❖ 항공사에서는 짐을 부칠 때, **20kg이 넘어가면** 20,000원을 내야한다. **20kg이하 이면** 수수료는 없다. 사용자로부터 **짐의 무게를 입력**받고 사용자가 **지불하여야 할 금액을 계산**하는 프로그램을 작성해보자.

짐의 무게는 얼마입니까? 18
짐에 대한 수수료는 없습니다.
감사합니다.

짐의 무게는 얼마입니까? 30
무거운 짐은 20,000원을 내셔야 합니다.
감사합니다.

```
weight = float(input("짐의 무게는 얼마입니까? "))  
  
if weight > 20:  
    print("무거운 짐은 20,000원을 내셔야 합니다. ")  
else:  
    print("짐에 대한 수수료는 없습니다. ")  
print("감사합니다. ")
```

예제: 정수들의 합

- ❖ 1부터 사용자가 입력한 수 n까지 더해서 출력하는 프로그램 (for 문 사용)

어디까지 계산할까요: 10
1부터 10 까지의 정수의 합= 55

```
# 반복을 이용한 정수합 프로그램
sum = 0

limit = int(input("어디까지 계산할까요: "))
for i in range(1, limit+1):
    sum += i

print("1부터 ", limit, "까지의 정수의 합= ", sum)
```