



# 데이터 관리를 위한 파이썬 PYTHON 기초 과정

## 1일차 : Python 기초 (1)

- Python 특징
- 기초 자료형(1)
- 제어문

## 2일차 : Python 기초 (2)

- 함수 (입출력)
- 파일 읽고 쓰기
- 클래스

## 3일차 : 데이터 분석 툴

- Numpy
- Pandas
- matplotlib
- 응용 예제

## 4일차 : 데이터 분석 실습

- 데이터 분석 예제1
- 데이터 분석 예제2

# 1. 파이썬 특징, 파이썬 설치

---

- 파이썬 설치하기
- 파이썬 소개
- 파이썬 개발환경 설치 (IDLE)



파이썬(Python)은 1990년 암스테르담의 귀도 반 로섬(Guido Van Rossum)이 개발한 **인터프리터 언어**이다. 귀도는 파이썬이라는 이름을 자신이 좋아하는 코미디 쇼인 "몬티 파이썬의 날아다니는 서커스(Monty Python's Flying Circus)"에서 따왔다고 한다.

파이썬은 문법이 쉬워 빠르게 배울 수 있다

파이썬은 무료이지만 강력하다

파이썬은 간결하다

파이썬은 개발 속도가 빠르다



"Life is too short, You need python." (인생은 너무 짧으니 파이썬이 필요해.)

# 파이썬 설치

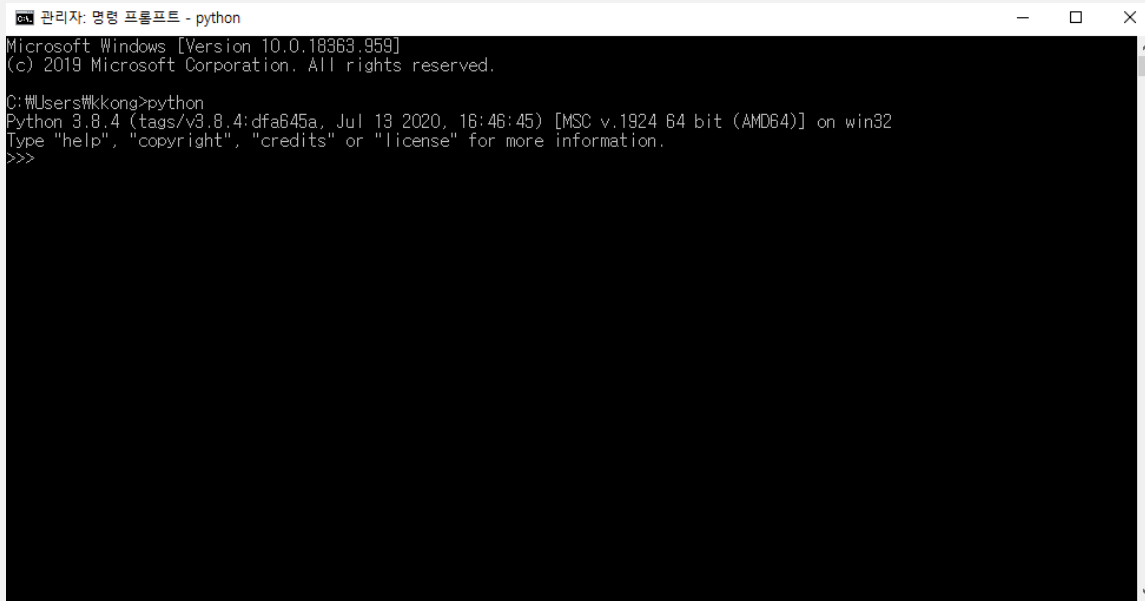
The image shows the Python 3.8.3 (32-bit) Setup window. The background is the Python.org website, which includes a navigation bar with links to Python, PSF, Docs, PyPI, Jobs, and Community. The main content area of the website displays the Python logo, a search bar, and a 'Download for Windows' section. The setup window is titled 'Python 3.8.3 (32-bit) Setup' and contains the following elements:

- Install Python 3.8.3 (32-bit)**: Select Install Now to install Python with default settings, or choose Customize to enable or disable features.
- Install Now**: A button with a red border, labeled 'Install Now' and 'C:\Users\kkong\AppData\Local\Programs\Python\Python38-32'. Below it, it states 'Includes IDLE, pip and documentation' and 'Creates shortcuts and file associations'.
- Customize installation**: A button labeled 'Customize installation' and 'Choose location and features'.
- Install launcher for all users (recommended)**: A checkbox that is checked.
- Add Python 3.8 to PATH**: A checkbox that is checked and highlighted with a red border.
- Cancel**: A button in the bottom right corner.

The website background also features a 'Get Started' section with a power icon, a 'Download' section with a download icon, and a 'Docs' section with a document icon. The 'Get Started' section includes a link to 'Start with our Beginner's Guide'. The 'Download' section mentions 'Latest: Python 3.8.3'. The 'Docs' section mentions 'docs.python.org'.

# 통합 개발 환경 (IDE) 설치

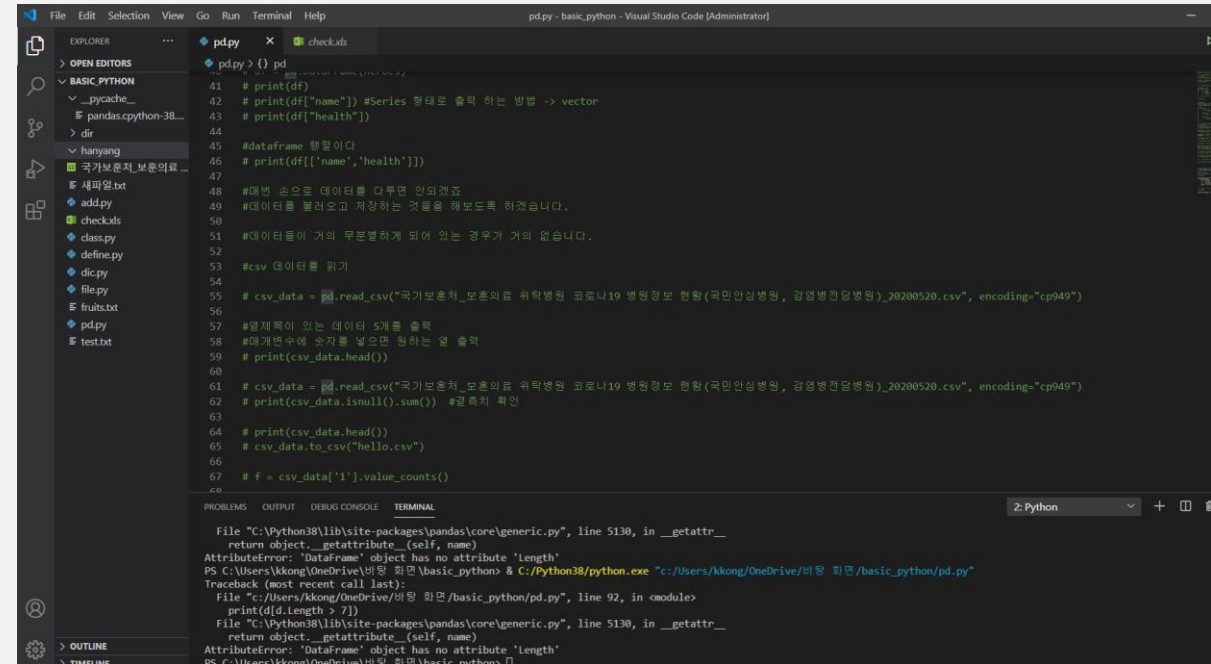
## 파이썬 IDE (Integrated Development Environment, 통합개발환경툴)



```
관리자: 명령 프롬프트 - python
Microsoft Windows [Version 10.0.18363.959]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\kkong>python
Python 3.8.4 (tags/v3.8.4:dfa645a, Jul 13 2020, 16:46:45) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

Shell



```
File Edit Selection View Go Run Terminal Help
pd.py - basic_python - Visual Studio Code [Administrator]

EXPLORER
  OPEN EDITORS
  BASIC_PYTHON
    __pycache__
    pandas.cpython-38...
    > dir
    hanyang
      국가보훈처 보훈이력...
      새파일.txt
      add.py
      checks
      class.py
      define.py
      dicpy
      file.py
      fruits.txt
      pd.py
      test.txt

pd.py
41 # print(df)
42 # print(df[["name"]]) #Series 형태로 출력 하는 방법 -> vector
43 # print(df[["health"]])
44
45 #dataframe 형식이다
46 # print(df[["name", "health"]])
47
48 #대형 손으로 데이터를 다루면 안되겠다
49 #데이터를 불러오고 저장하는 것들을 해보도록 하겠습니다.
50
51 #데이터들이 거의 무분별하게 되어 있는 경우가 거의 없습니다.
52
53 #csv 데이터를 읽기
54
55 # csv_data = pd.read_csv("국가보훈처_보훈이력_위탁병원_코로나19 병원정보 현황(국민안심병원, 감염병전담병원)_20200520.csv", encoding="cp949")
56
57 #알짜책이 있는 데이터 5개를 출력
58 #대개 변수에 숫자를 넣으면 원하는 열 출력
59 # print(csv_data.head())
60
61 # csv_data = pd.read_csv("국가보훈처_보훈이력_위탁병원_코로나19 병원정보 현황(국민안심병원, 감염병전담병원)_20200520.csv", encoding="cp949")
62 # print(csv_data.isnull().sum()) #공백치 확인
63
64 # print(csv_data.head())
65 # csv_data.to_csv("hello.csv")
66
67 # f = csv_data['1'].value_counts()
68

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
2: Python
File "C:\Python38\lib\site-packages\pandas\core\generic.py", line 5130, in __getattr__
    return object.__getattr__(self, name)
AttributeError: 'DataFrame' object has no attribute 'length'
PS C:\Users\kkong\OneDrive\바탕 화면\basic_python> & C:\Python38\python.exe "c:/Users/kkong/OneDrive/바탕 화면/basic_python/pd.py"
Traceback (most recent call last):
  File "c:/Users/kkong/OneDrive/바탕 화면/basic_python/pd.py", line 92, in <module>
    print(df.length > 7)
  File "C:\Python38\lib\site-packages\pandas\core\generic.py", line 5130, in __getattr__
    return object.__getattr__(self, name)
AttributeError: 'DataFrame' object has no attribute 'length'
PS C:\Users\kkong\OneDrive\바탕 화면\basic_python>
```

Code Editor

# Python Code Editor



Visual Studio code



# Visual Studio Code (VSCode)

Google

Visual studio code

전체 이미지 동영상 쇼핑 도서 더보기 설정 도구

검색결과 약 625,000,000개 (0.51초)

code.visualstudio.com ▾ 이 페이지 번역하기

## Visual Studio Code - Code Editing. Redefined

**Visual Studio Code** is a code editor redefined and optimized for building web and cloud applications. **Visual Studio Code** is free and ...

이 페이지를 3번 방문했습니다. 최근 방문 날짜: 20. 6. 24

### Download Visual Studio Code

Visual Studio Code is free and available on your favorite ...

### Docs

Intro Videos - Windows - GCC on Windows - Basic Editing - Java

### Updates

May 2020 - April 2020 - November 2019 - HexEditor - ...

visualstudio.com 검색결과 더보기 »

### Visual Studio Code

Visual Studio Code contain a Con...

### Extension

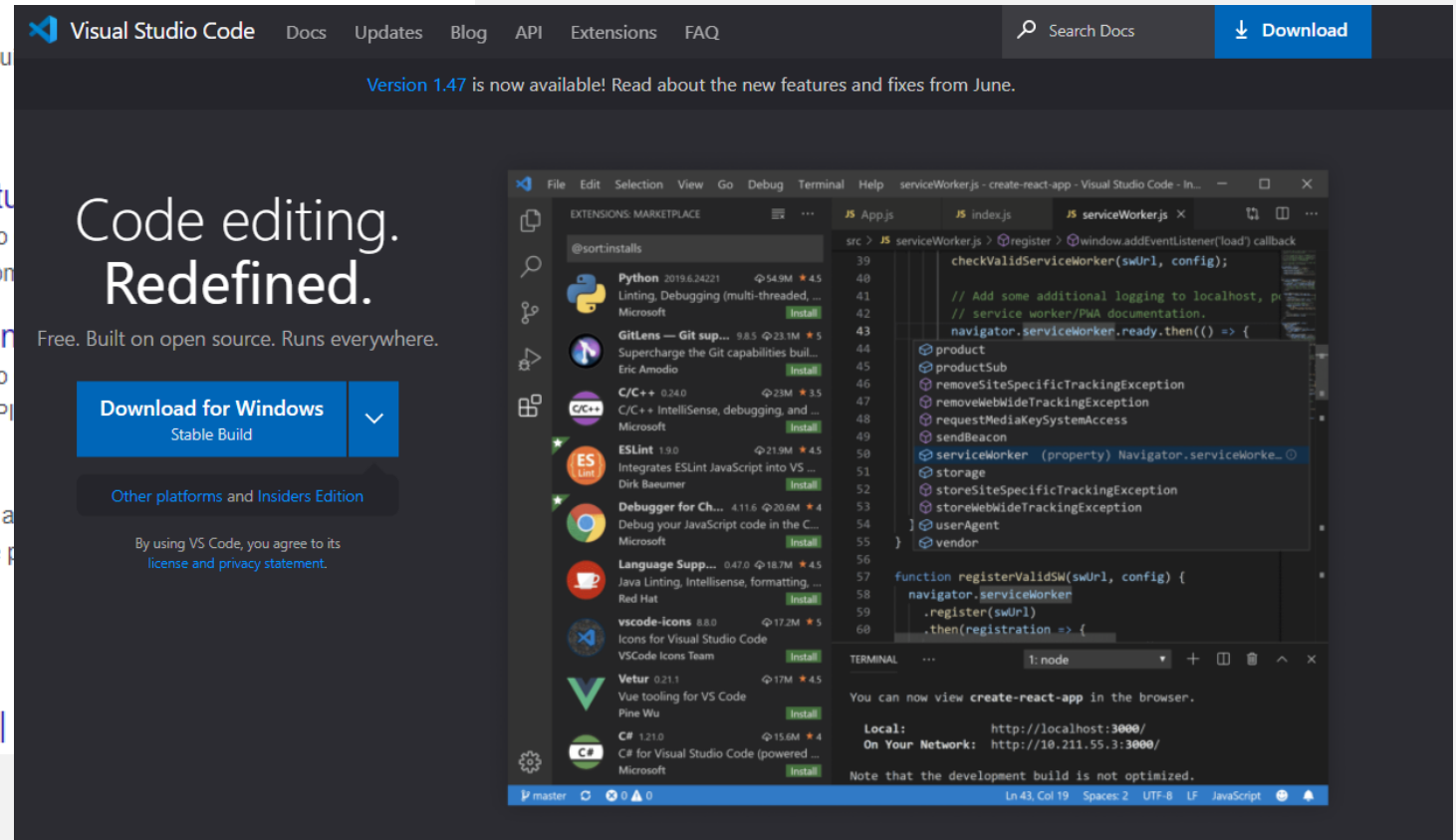
Visual Studio Code extension API

### License

This license a... Studio Code p...

www.44bits.io > keyword > visual-studio-code

## 비주얼 스튜디오 코드(Visual Studio Code)란? |





## 2. 파이썬 기초 자료형

---

- 숫자형
- 문자열 자료형
- 리스트, 튜플 자료형
- 딕셔너리 자료형
- 불(Bool) 자료형
- 연습문제

# 파이썬 기본 문법 : 숫자형

## 정수형

```
>>> a = 123
>>> a = -178
>>> a = 0
```

## 실수형

```
>>> a = 1.2
>>> a = -3.45
```

## 사칙연산

```
>>> a = 3
>>> b = 4
>>> a + b 7
>>> a * b 12
>>> a / b 0.75
```

## 제곱 연산

```
>>> a = 3
>>> b = 4
>>> a ** b 81
```

## 나머지 연산

```
>>> 7 % 3
1
>>> 3 % 7
3
```

## 몫

```
>>> 7 // 3
2
>>> 3 // 7
0
```

## 파이썬 기본 문법 : 문자열

- 문자는 `'''` 또는 `''` 안에 문자를 작성
- 다음 줄로 쓰는 것은 `\n` 을 활용하여 작성
- 문자열끼리 + 연산이 가능함
- 문자열 \* (숫자) 로 연산이 가능
- 문자열 길이를 찾을 수 있음.

```
>>> food = "apple"
```

```
>>> food  
"apple"
```

```
>>> multiline = "Life is too short\nYou need python"
```

```
>>> head = "Python"
```

```
>>> tail = " is fun!"
```

```
>>> head + tail 'Python is fun!'
```

```
>>> a = "python"
```

```
>>> a * 2  
'pythonpython'
```

```
>>> a = "Life is too short"
```

```
>>> len(a) 17
```

## 파이썬 기본 문법 : 문자열 인덱싱

### 문자변수[숫자]

변수[ '시작' : '끝' : '간격' ] \* 단 빈칸은 모두를 나타낸다

```
>>> a = "Life is too short, You need Python"
>>> a[0] 'L' >>> a[12] 's' >>> a[-1] 'n'
```

```
>>> a[-2] 'o'
>>> a[-5] 'y'
```

```
>>> a = "Life is too short, You need Python"
>>> b = a[0] + a[1] + a[2] + a[3]
>>> b 'Life'
```

```
>>> a = "Life is too short, You need Python"
>>> a[0:4] 'Life'
```

```
>>> a[19:] 'You need Python'
```

```
>>> a[:17] 'Life is too short'
```

```
>>> a[:] 'Life is too short, You need Python'
```

# 파이썬 기본 문법 : 문자열 포매팅

## 문자열 포맷 코드

### 숫자 출력

```
a = 10
b = 20
print('사과를 %d개 %d번 먹었습니다.' %(a,b))
```

### 문자 출력

```
a = 10
b = '엄청 많이'
print('사과를 %d개 %s 먹었습니다.' %(a,b))
```

## f문자열 포매팅 (파이썬 3.6버전 이상에서만 사용)

```
a = 10.5
b = '엄청 많이'
```

```
print(f'{a}개의 사과를 {b} 먹었습니다.')
```

## 파이썬 기본 문법 : 문자열 관련 함수

### 문자개수 세기

```
>>> a = "hobby"  
>>> a.count('b') 2
```

### 위치 알려주기 (find)

```
>>> a = "Python is the best choice"  
>>> a.find('b')  
14  
>>> a.find('k')  
-1
```

### 위치 알려주기2 (index)

```
>>> a = "Life is too short"  
>>> a.index('t') 8  
>>> a.index('k')  
Traceback (most recent call last): File  
"<stdin>", line 1, in <module> ValueError:  
substring not found
```

소문자 -> 대문자(upper)

대문자 -> 소문자 (lower)

왼쪽 공백 지우기(lstrip)

오른쪽 공백 지우기(rstrip)

양쪽 공백 지우기(strip)

문자열 바꾸기 (replace)

문자열 나누기(split)

## 파이썬 기본 문법 : 리스트

### 리스트의 생성

리스트는 배열과 같음, 리스트는 숫자, 정수, 문자 ,배열 모두 올 수 있음

```
>>> a = []  
>>> a = list()  
>>> b = [1, 2, 3]  
>>> c = ['Life', 'is', 'too', 'short']  
>>> d = [1, 2, 'Life', 'is']  
>>> e = [1, 2, ['Life', 'is']]  
  
>>> a = [1, 2, 3, ['a', 'b', 'c']]
```

# 파이썬 기본 문법 : 리스트

## 리스트의 슬라이싱

리스트 내부 값을 추출할 때 사용하는 방법

```
>>> a[0]
1
>>> a[-1]
['a', 'b', 'c']
>>> a[3]
['a', 'b', 'c']
>>> a[-1][0]
'a'

>>> a = [1, 2, 3, 4, 5]
>>> a[0:2]
[1, 2]
```

## 리스트 값 수정

```
>>> a = [1, 2, 3]
>>> a[2] = 4
>>> a [1, 2, 4]
```

## 리스트 값 삭제

```
>>> a = [1, 2, 3]
>>> del a[1]
>>> a [1, 3]
```



## 파이썬 기본 문법 : 리스트

### 리스트에 요소 추가(append)

```
>>> a = [1, 2, 3]
>>> a.append(4)
>>> a [1, 2, 3, 4]

>>> a.append([5,6])
>>> a [1, 2, 3, 4, [5, 6]]
```

### 리스트에 요소 추가(insert)

```
>>> a = [1, 2, 3]
>>> a.insert(0, 4)
>>> a [4, 1, 2, 3]
```

### 위치 반환(index)

```
>>> a = [1,2,3]
>>> a.index(3)
2
>>> a.index(1) 0
```

### 리스트 정렬 sort

```
>>> a = [1, 4, 3, 2]
>>> a.sort()
>>> a [1, 2, 3, 4]
```

### 리스트 뒤집기

```
>>> a = ['a', 'c', 'b']
>>> a.reverse()
>>> a ['b', 'c', 'a']
```

## 파이썬 기본 문법 : 리스트

### 리스트에 값 빼기(pop)

가장 마지막에 입력된 값을 빼내는 함수 pop()

```
>>> a = [1,2,3]
>>> a.pop()
3
>>> a [1, 2]
```

```
>>> a = [1,2,3]
>>> a.pop(1)
2
>>> a [1, 3]
```

## 파이썬 기본 문법 : 딕셔너리

### 딕셔너리 구조

{Key1:Value1, Key2:Value2, Key3:Value3, ...}

```
>>> a = {1: 'hi'}
```

```
>>> a = { 'a': [1,2,3]}
```

```
>>> a = {1: 'a'}
```

```
>>> a[2] = 'b'
```

```
>>> a {1: 'a', 2: 'b'}
```

```
>>> del a[1]
```

```
>>> a {2: 'b', 'name': 'pey', 3: [1, 2, 3]}
```

### 딕셔너리

```
>>> grade = {'pey': 10, 'julliet': 99}
```

```
>>> grade['pey']
```

```
10
```

```
>>> grade['julliet']
```

```
99
```

```
>>> dic = {'name': 'pey', 'phone': '0119993323',  
'birth': '1118'}
```

```
>>> dic['name']
```

```
'pey'
```

```
>>> dic['phone']
```

```
'0119993323'
```

```
>>> dic['birth']
```

```
'1118'
```

```
>>> a = {'name': 'pey', 'phone': '0119993323',  
'birth': '1118'}
```

```
>>> 'name' in a True
```

```
>>> 'email' in a False
```

## 파이썬 기본 문법 : 딕셔너리

### 딕셔너리 keys 살펴보기

```
>>> a = {'name': 'pey', 'phone': '0119993323',  
        'birth': '1118'}  
>>> a.keys() dict_keys(['name', 'phone',  
                        'birth'])
```

### 딕셔너리 삭제

```
>>> a.clear()  
>>> a  
{}
```

### 딕셔너리 values 살펴보기

```
>>> a.values() dict_values(['pey',  
                          '0119993323', '1118'])
```

## 파이썬 기본 문법 : 불 자료형(True or False)

Bool 자료형이란 True(참) False(거짓)을 나타내는 자료형  
True / False 2가지만 존재

첫번째 문자는 반드시 대문자로 사용

Bool 경우에는 값이 있으면 True, 값이 없으면 False  
0이면 False, 1 이면 True

## 연습 문제 1

홍길동 씨의 주민등록번호는 “881120-1068234”

홍길동 씨의 주민등록번호를 연월일(YYMMDD) 부

분과 그 뒤의 숫자 부분으로 나누어 출력

(문자열 슬라이싱)

성별을 나타내는 숫자를 출력

(문자열 인덱싱 사용)

### 3. 제어문

---

- if문
- while 문
- for문
- 연습문제

### 3. 제어문

---

#### - if문 : 조건문

어떠한 조건이 충족 되었을 때에만 작동하는 제어문

*if 조건문:*

수행할 문장1

수행할 문장2

...

*else:*

수행할 문장1

수행할 문장2

...

들여쓰기!



# 파이썬 기본 문법 : 조건문

## If 조건문

```
>>> money = True
>>> if money:
...     print("택시를 타고 가라")
... else:
...     print("걸어 가라")

... 택시를 타고 가라
```

## elif 조건문

```
money = 2000

if money >= 3000:
    print('택시를 타고 가라')
elif money >= 1500:
    print('자전거를 빌리세요')
else:
    print('걸어서 가세요')
```

## 파이썬 기본 문법 : while문

### while 문

조건에 충족하지 않을 때 까지 반복

while 조건문:  
    수행하는 문장

```
treeHit = 0
```

```
while treeHit < 10:  
    treeHit += 1  
    print('나무를 %d번 찍었습니다.' %treeHit)  
    if treeHit == 10:  
        print("나무가 넘어갑니다.")
```

```
count = 0
```

```
while True:  
    count += 1  
    print(count)  
    if count == 10:  
        break  
    print('빠져나갑니다.')
```

```
while True:  
    count += 1  
    if count % 2 == 1: #홀수라면  
        continue  
    print(count)  
    if count == 10:  
        break
```

## 파이썬 기본 문법 : for문

### For문

```
>>> test_list = ['one', 'two', 'three']
>>> for i in test_list:
...     print(i)
... one two three
```

### For문 활용

```
# marks1.py
marks = [90, 25, 67, 45, 80]
number = 0
for mark in marks:
    number = number + 1
if mark >= 60:
    print("%d번 학생은 합격입니다." % number)
else:
    print("%d번 학생은 불합격입니다." % number)
```

## 2. 파이썬 기초 자료형

---

### 연습문제 1

while문을 사용해 1부터 1000까지의 자연수 중 3의 배수의 합을 구해 보자.

## 2. 파이썬 기초 자료형

---

### 딕셔너리 자료형 연습문제

리스트에 10개 이상의 과일을 적고 (중복허용)  
이를 분류하는 코드를 작성하시오.

예) fruits = ['사과', '딸기', '파인애플' ... '사과' ]

Hint 1

딕셔너리 추가 a[(추가할 키)] = Value

Hint 1

딕셔너리 내에 Key가 있는지 확인 'key' in 딕셔너리