

# 마크다운(Markdown)

---

일반 텍스트 형식 구문을 사용하는 마크업 언어의 일종으로 사용법이 쉽고 간결하여 빠르게 문서 정리를 할 수 있습니다. 단, 모든 HTML 마크업을 대체하지는 않습니다.

## 1. 문법

---

### 1.1 Header

헤더는 제목을 표현할 때 사용합니다. 단순히 글자의 크기를 표현하는 것이 아닌 의미론적인 중요도를 나타냅니다.

- `<h1>` 부터 `<h6>` 까지 표현 가능합니다.
- `#` 의 개수로 표현하거나 `<h1></h1>` 의 형태로 표현 가능합니다.

# h1 태그입니다.

---

## h2 태그입니다.

---

### h3 태그입니다.

#### h4 태그입니다.

##### h5태그입니다

###### h6 태그입니다.

### 1.2 List

목록을 나열할 때 사용합니다. 순서가 필요한 항목과 그렇지 않은 항목으로 구분할 수 있습니다. 순서가 있는 항목 아래 순서가 없는 항목을 지정할 수 있으며 그 반대도 가능합니다.

- 순서가 없는 목록
  - `1.` 을 누르고 스페이스바를 누르면 생성할 수 있습니다.
  - `tab` 키를 눌러서 하위 항목을 생성할 수 있고 `shift + tab` 키를 눌러서 상위 항목으로 이동 할 수 있습니다.
- 순서가 있는 목록
  - `-` (하이픈)을 쓰고 스페이스바를 누르면 생성할 수 있습니다
  - `tab` 키를 눌러서 하위 항목을 생성할 수 있고 `shift + tab` 키를 눌러서 상위 항목으로 이동 할 수 있

습니다.

1. 순서가 있는 항목
  2. 순서가 있는 항목
    1. 순서가 있는 하위 항목
    2. 순서가 있는 하위 항목
- 
- 순서가 없는 항목
  - 순서가 없는 항목
    - 순서가 없는 하위 항목
    - 순서가 없는 하위 항목

## 1.3 Code Block

코드 블록은 작성한 코드를 정리하거나 강조하고 싶은 부분을 나타낼 때 사용합니다. 인라인과 블록 단위로 구분할 수 있습니다.

- Inline
  - 인라인 블록으로 처리하고 싶은 부분을 ` (백틱) 으로 감싸줍니다.
- Block
  - ` (백틱) 을 3번 입력하고 `Enter` 를 눌러 생성합니다.

```
import random

numbers = range(1, 46)
lotto = random.sample(numbers, 6)
print(lotto)
```

## 1.4 Image

로컬에 있는 이미지를 삽입하거나 이미지 링크를 활용하여 이미지를 나타낼 때 사용합니다.

- `![]()` 을 작성하고 `()` 안에 이미지 주소를 입력합니다. `[]` 안에는 이미지 파일의 이름을 작성합니다.
- 로컬에 이미파일을 저장한 경우 절대 경로가 아닌 상대 경로를 사용하여 이미지를 저장합니다.



## 1.5 Link

특정 주소로 링크를 걸 때 사용합니다.

- `[ ]()` 을 작성하고 `()` 안에 링크 주소를 작성하고 `[ ]` 안에 어떤 링크 주소인지 작성합니다.

[git 공식문서](#)

[github 공식문서](#)

## 1.6 Table

표를 작성하여 요소를 구분할 수 있습니다.

- `|` (파이프) 사이에 컬럼을 작성하고 `enter` 를 입력합니다.
- 마지막 컬럼을 작성하고 뒤에 `|` 를 붙여줍니다.

Name	Age	Job
Alice	50	Developer
Peter	50	Developer

## 1.7 기타

### 인용문

- `>` 을 입력하고 `enter` 키를 누릅니다.

git은 컴퓨터 파일의 변경사항을 추적하고 여러 명의 사용자들 간에 해당 파일들의 작업을 조율하기 위한 분산 버전 관리 시스템이다.

- 인용문 안에 인용문을 작성하면 중첩해서 사용할 수 있습니다.

\$ pip

\$ 내게

\$ 빠져

### 수평선

- `---` , `***` , `___` 을 입력하여 작성합니다.

TITLE1

---

TITLE2

---

TITLE3

---

### 강조

- 이탤릭체는 해당 부분을 `*` 혹은 `_` (언더바) 로 감싸줍니다.
- बो드체는 해당 부분을 `**` 혹은 `__` (언더바 2개)로 감싸줍니다.
- 취소선은 `~~` 표시를 사용합니다.

이것은 *이탤릭체*입니다.

이것은 **볼드체**입니다.

이것은 취소선입니다.

---

## 2. 과제

---

아래의 제시문을 마크다운 문법을 활용하여 작성해주세요.

글을 어떻게 구조화 시켜서 작성할 지 곰곰이 생각해보고 작성해주세요.

### 2.1 제시문

Python 기초

#### 1. 저장

##### 1.1 저장이란?

dust = 60

dust는 60이다(X)

dust에 60을 저장(할당)한다(O)

dust == 60

dust에 저장된 값은 60과 같다

##### 1.2 무엇을 저장할 수 있을까?

###### 1.2.1 숫자

현실세계에 존재하는 모든 숫자

기본적인 연산 가능

O : 12, -365, 3.141592

X : 23ab, ba123

###### 1.2.2 글자

현실세계에 존재하는 모든 글자

따옴표로 둘러싼 글자 혹은 숫자

'미세먼지'

'58도 글자입니다'

'글자는 반드시 따옴표를 붙여야 합니다!'

### 1.2.3 참/거짓

True 혹은 False

주로 조건 및 반복에 사용됨

300 > 20 => True

150 == 161 => False

## 1.3 어떻게 저장할 수 있을까?

### 1.3.1 변수(Variable)

저장된 값을 변경할 수 있는 박스

숫자, 글자, 참거짓을 담을 수 있다

dust = 58

dust = 60

print(dust) vs print('dust')

### 1.3.2 리스트(List)

박스의 목록

여러 개의 박스가 순서대로 붙어있다

dust = [58, 40, 70]

print(dust[1])

### 1.3.3 딕셔너리(Dictionary)

견출지를 붙인 박스들의 묶음

dust = {'영등포구': 58, '강남구': 40}

print(dust['영등포구'])

## 2. 조건

특정 조건일 때 코드를 실행할 수 있다

if 조건식1:

코드1

elif 조건식2:

코드2

else:

코드3

### 3. 반복

#### 3.1 while

while에 해당하는 조건일 동안 계속 반복한다.

조건식을 만족하면 계속 반복하고, 만족하지 않으면 반복을 끝낸다.

종료 조건이 반드시 필요하다.

```
n = 0
```

```
while n < 3:
```

```
    print('출력')
```

```
    n = n+1
```

#### 3.2 for

정해진 박스 내에서의 반복 시 사용한다.

'가지고 있는 모든 것을 꺼낸다'

종료 조건이 필요없다.

```
dust = [59, 24, 102]
```

```
for i in dust:
```

```
    print(i)
```