

Professional Coding Specialist

# COS Pro 파이썬 1 급

---

---

## 1 강. 문법 정리 1

---

1. 파이썬 시작
  2. 파이썬 변수와 조건문
-

## 과정 소개

COS Pro(Coding Specialist Professional) 시험은 요구사항을 분석하여 프로그램을 설계, 구현하는 능력과 주어진 프로그램을 디버깅하여 수정하는 능력을 평가하는 자격증 시험이며, COS Pro 1 급 시험은 높은 수준의 프로그래밍 능력을 증명할 수 있다. 이번 시간에는 COS Pro 1 급 파이썬 시험을 대비하기 위한 모의고사를 풀어보기 전에 파이썬의 기본 문법을 정리하는 첫 번째 시간으로 파이썬에 대한 간략한 소개와 파이썬 변수를 사용하여 데이터를 다루는 기본적인 방법, 파이썬 조건문의 사용법에 대해 정리해 보고자 한다.

## 학습 목차

1. 파이썬 시작
2. 파이썬 변수와 조건문

## 학습 목표

1. 파이썬의 특징을 이해하고 파이썬 프로그래밍을 하기 위해 사용할 수 있는 다양한 ide 에 대해 알아보고, 학습자에게 맞는 ide 를 선택하여 사용할 수 있다.
2. 파이썬 변수에 저장하는 자료형을 알아보고, 자료형 간의 형 변환 방법과 자료형 별로 출력 서식을 지정하는 방법을 익혀 프로그래밍에서 활용할 수 있다.
3. 파이썬에서 제공하는 연산자와 문자열 슬라이싱의 특징을 이해하고 사용할 수 있다.
4. 다른 프로그래밍 언어와 다른 파이썬 조건문의 특징을 알아보고 프로그래밍에서 활용할 수 있다.

# 1

## 파이썬 시작

### 1. 파이썬 소개

#### 1) 파이썬의 특징

- 1990 년 네덜란드의 귀도 반 로섬에 의해 개발된 프로그래밍 언어.
- 객체 지향 언어로서 인터프리터 방식.
- 문법이 간결하고 쉬워서 초보자도 사용하기 쉬운 프로그래밍 언어.
- 웹 개발, 빅데이터 분석, 머신러닝, IoT 등에 활용되고 있음.
- 오픈 소스형태로 무료로 배포, 사용되고 있음.
- docs.python.org 에서 라이브러리나 학습자료를 참고할 수 있음.

#### 2) 추가 라이브러리 설치

- 필요한 라이브러리를 PYPI 에서 다운받아 설치할 수 있음.
- 윈도우 명령 프롬프트창에서 명령어를 입력하여 설치.
  - 사용법 : pip install 라이브러리명
- 라이브러리를 설치한 후 프로그램 코드 안에서 import 하여 설치한 모듈을 사용할 수 있음.

### 2. 파이썬 에디터 소개

#### 1) 파이썬 IDLE : 파이썬 기본 에디터

##### ① 설치

- <https://www.python.org> 에서 다운로드 후 설치.



## [COS Pro 1 급 파이썬] 1. 문법 정리 1

- 설치 화면에서 'Add Python 3.9 to PATH' 를 반드시 체크.



( ※ 파이썬을 설치할 때 컴퓨터 명과 경로 명에 한글을 쓰지 않는 것을 권장 )

### ② 기본 사용법

- [윈도우 시작 메뉴] - [ IDLE (Python IDLE 3.9) ] 실행.

새 파이썬 파일 생성	: [File] - [New File]
파일 저장	: [File] - [Save] : .py 파일로 저장.
옵션 지정	: [Option] - [Configure IDLE]
실행	: [Run] - [Run Module]
한 번에 여러 줄의 코드에 대한 들여쓰기 조정.	: 들여쓰기를 조정할 부분을 선택한 후 [Format] - [Indent Region] / [Dedent Region]을 이용해서 조정.

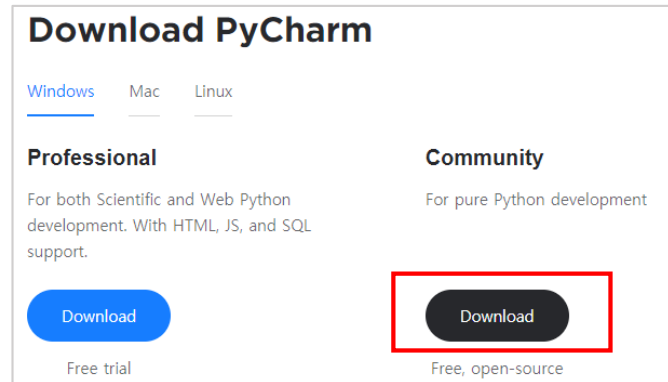
- 추가 라이브러리 설치하기

윈도우 검색창에 cmd 를 입력해서 명령 프롬프트를 찾아 실행.	
명령 프롬프트 창에서 pip install 라이브러리명을 입력 후 엔터. (예 : numpy 라이브러리 설치)	

2) 파이참 : JetBrains 가 개발한 파이썬 IDE. 편리한 디버깅 기능과 자동완성 기능을 제공.

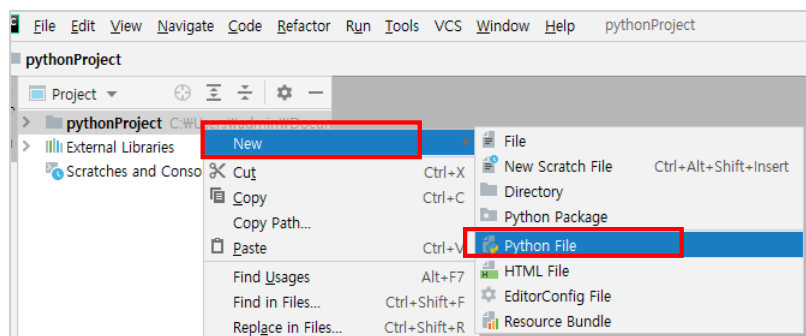
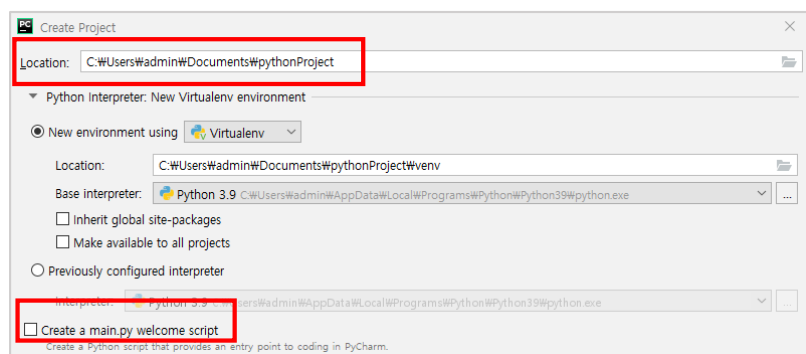
### ① 설치

- ♦ <https://www.jetbrains.com/ko-kr/pycharm> 의 다운로드 페이지에서 community 버전으로 설치.



## ② 사용법

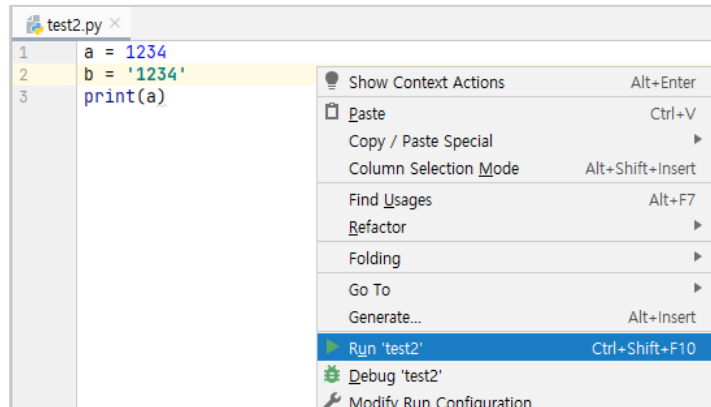
- ♦ 파일 생성
  - [File]-[New Project] 를 선택하여 새 프로젝트 생성.
  - 새 프로젝트를 저장할 폴더를 지정.
  - main.py 파일은 생성하지 않도록 Create a main.py 를 언체크.
  - 파이썬 파일을 생성하기 위해 [File] - [New] 를 선택하거나, 프로젝트 창에서 프로젝트를 선택한 후 팝업 메뉴를 띄워 [New] - [Python File] 을 선택.



- ♦ 실행

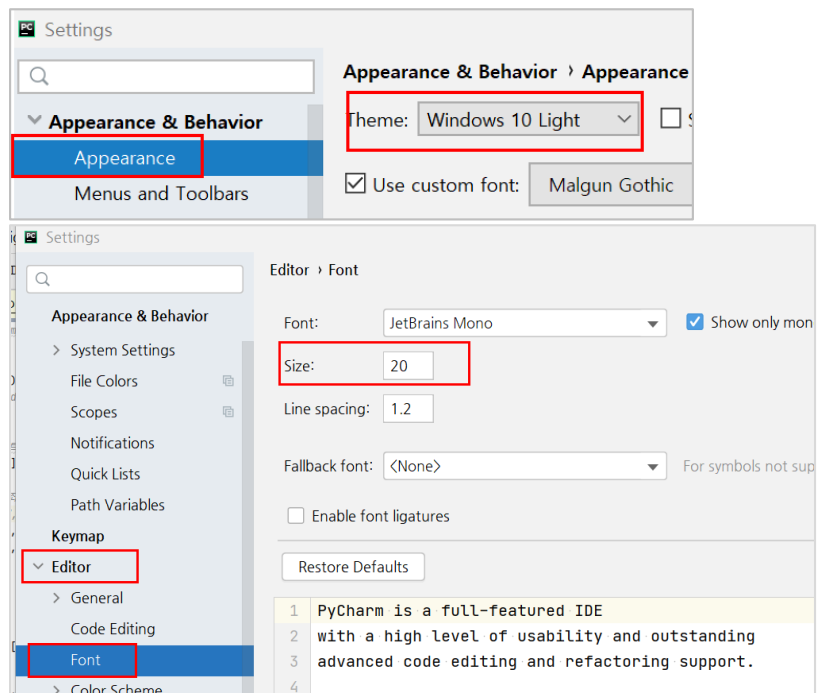
## [COS Pro 1 급 파이썬] 1. 문법 정리 1

- 코드를 모두 작성한 뒤에 마우스 오른쪽 버튼으로 팝업 메뉴를 띄워 [Run 파일 이름] 을 선택해서 실행.



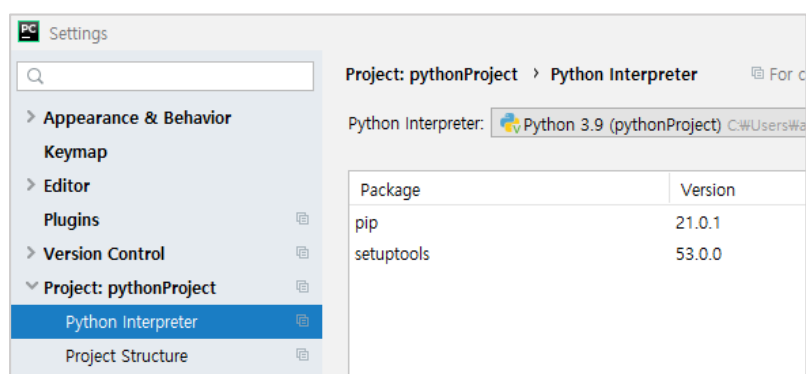
### ◆ 파이참 모양과 폰트 변경

- [File] - [Settings] 의 Appearance 항목과 Editor 항목을 이용하여 파이참 화면의 색이나 폰트를 원하는 것으로 선택 가능.



### ◆ 파이참에서 사용하는 파이썬 버전 선택

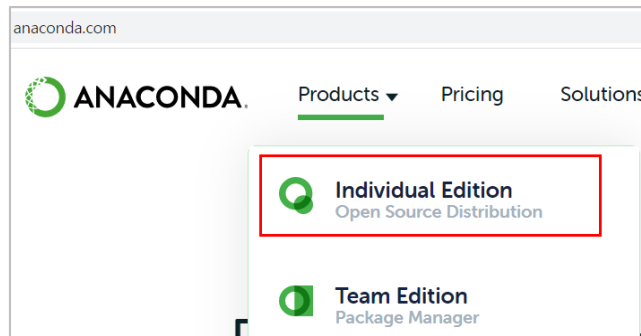
- [File] - [Settings] - [Project: pythonProject] - [Python Interpreter] 항목에는 컴퓨터에 설치되어 있는 파이썬 중에서 파이참에서 사용할 파이썬을 선택할 수 있음.



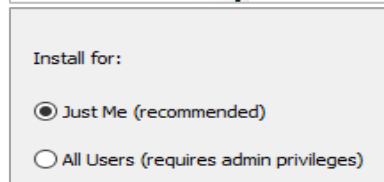
### 3) 아나콘다(Anaconda)

#### ① 설치

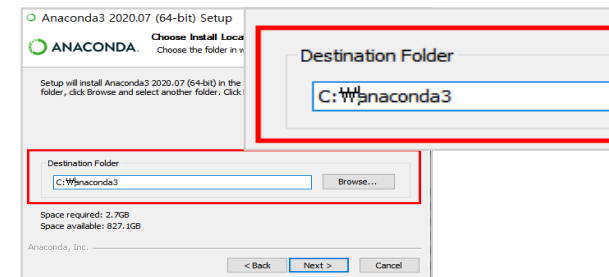
- [www.anaconda.com](http://www.anaconda.com) 에 접속한 후 [Product] - [Individual Edition] 을 선택하여 다운로드.



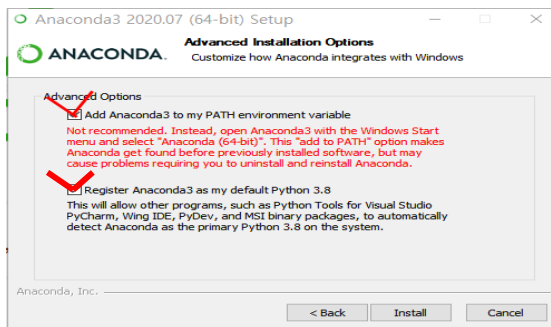
- 설치 화면에서 'Just Me' 를 선택.



- 설치 경로를 지정할 때 빈칸과 한글 명은 사용 불가.



- Path 설정할 때는 두 군데 모두 체크.



#### ② 아나콘다에서 제공하는 파이썬 에디터

주피터 노트북	<ul style="list-style-type: none"> <li>- 파이썬 프로그램의 실행 결과로 차트, 지도 등을 표현하기에 편리.</li> <li>- [윈도우 시작 메뉴] -[Anaconda] - [Jupyter Notebook]을 클릭해서 실행.</li> <li>- 주피터 노트북 실행 시 명령 창을 닫지 않도록 주의.</li> </ul>
스파이더	<ul style="list-style-type: none"> <li>- 편리한 디버깅 기능을 제공</li> </ul>

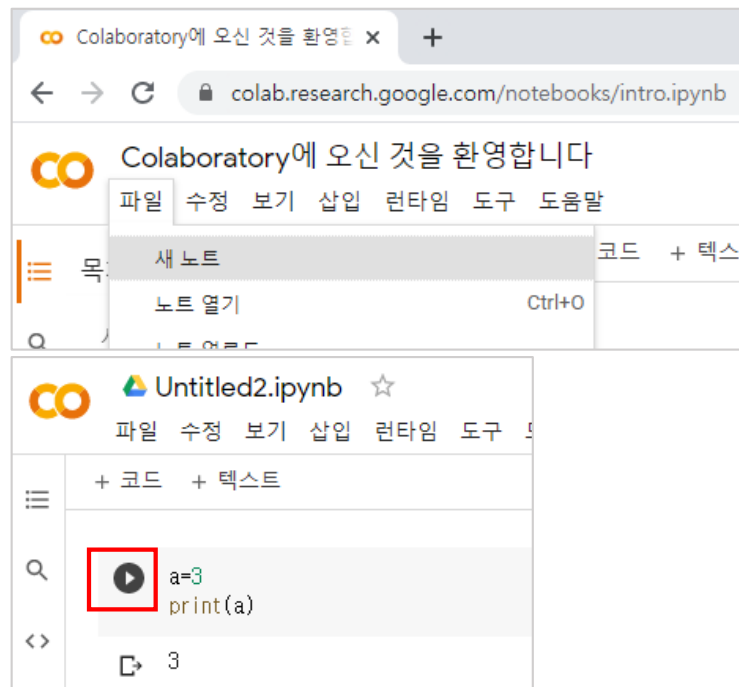
#### 4) Colab

##### ① 소개

- 구글에서 제공하는 클라우드 에디터로 Colab 을 사용하기 위해서는 구글 아이디가 필요함.
- 주피터 노트북과 비슷한 형태.
- 많이 사용되는 라이브러리가 탑재되어 있어서 라이브러리 추가 설치가 필요하지 않음.

##### ② 사용법

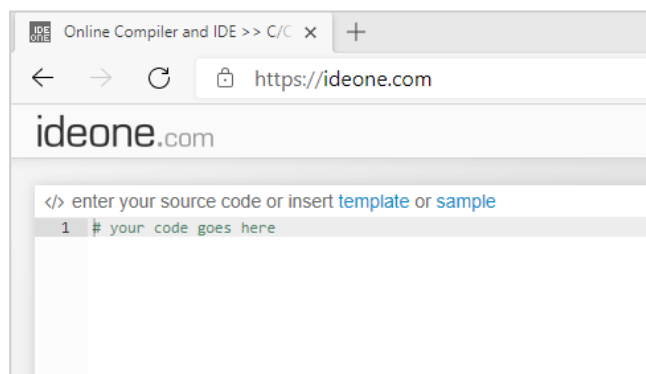
- 크롬 브라우저를 이용하여 구글 사이트에서 구글 코랩 (colab.research.google.com) 을 검색해서 접속.
- [파일] - [새 노트]를 클릭해서 노트를 생성.
- 코드를 입력한 후 셀의 오른쪽에 있는 실행 버튼을 누르거나 [Shift]+{Enter} 를 누르면 코드가 실행되어 결과가 셀의 아래쪽에 나타남.
- 파일을 저장하면 구글 드라이브에 저장됨.



#### 5) 기타 에디터 소개

##### ① ideone

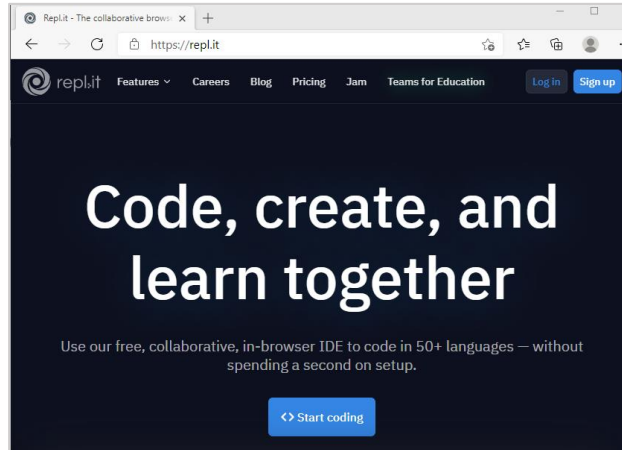
- ideone.com 에 접속하여 작업할 수 있음.
- JAVA, C, Python 등 여러 언어로 프로그래밍 할 수 있는 온라인 에디터.



##### ② 리플 :



- <https://repl.it/> 에 접속해서 사용하는 온라인 IDE.
- 회원가입 후 사용 가능.



## 2

## 파이썬 변수와 조건문

### 1. 파이썬 변수

#### 1) 특징

- 변수 사용을 위한 선언이 필요하지 않음.
- 변수 명으로는 알파벳, 밑줄, 숫자를 이용할 수 있으나 변수 명 시작은 알파벳이나 밑줄로 해야 함.
- 변수에 할당된 값에 따라 해당 변수의 자료형(type)이 결정됨.

#### 2) 자료형

자료형(type)	예시
정수(Integer)	30, 1, 2, 3, -1, -2, -3
실수(float)	3.14, 5.0, -1.0, 17.9874
문자열(string)	"안녕", "hello", "1" 문자열을 만들 때는 큰 따옴표( " ) 혹은 작은 따옴표( ' )을 문자 앞/뒤에 붙임.

#### 3) 주석(comment)


- # : 한 줄 주석을 생성할 때는 #을 앞에 붙여서 생성
- ''' 혹은 """ : 여러 줄의 주석을 생성할 때는 큰 따옴표 세 개 혹은 작은 따옴표 세 개를 주석의 앞 뒤에 붙여서 생성

#### 4) 변수 값 출력

- print() : 인수로 전달된 값을 출력하는 함수.

예) 

```
x = 3
print("파이썬", x, "버전", end="\n\n")
print("파이썬", x, "버전", sep="^^")
```

 파이썬 3 버전  
파이썬^^3^^버전

- end : print( ) 함수에 전달된 값을 출력한 후 마지막으로 출력하는 문자를 지정. 기본값은 'Wn'. 만일 print( ) 함수를 사용할 때 end="" 라고 지정하게 되면 값을 출력한 후 줄바꿈이 일어나지 않음.
- sep : print( ) 함수로 출력하는 항목들 사이에 출력하는 문자를 지정. 기본 값은 빈 칸( ' ' ).

♦ 특수 문자 표시

특수 문자	표현 내용
Wn	Enter
Wt	Tab 키
W'	'
W"	"
WW	W

5) 변수의 자료형(type) 변환

함수	변환	예시	결과
int(x)	x 를 정수로	int('10')	10
		int(5.3)	5
float(x)	x 를 실수로	float('5')	5.0
		float(5)	5.0
str(x)	x 를 문자열로	str(5)	'5'
		str(3.5)	'3.5'

6) 출력 서식 - 서식 문자열

예	기능
%s	문자열 출력.
%10s	10 자리 오른쪽 정렬. 나머지 빈 칸.
%5.3f	총 5 자리 실수. 소수 이하는 3 자리.
%10d	10 자리 정수.
%010d	10 자리 정수. 남은 자리는 0 으로 채움.
%-10s	10 자리 문자열. 왼쪽 정렬.
%-10d	10 자리 정수. 왼쪽 정렬. 나머지 빈 칸.

♦ 예 1)

```
a="파이썬"
b=20
print("%s아~ 반가워~ %d살 이구나~"%(a, b))
```



파이썬아~ 반가워~ 20살 이구나~

• 예 2)

```
a=23 ; b=32; c=(a+b)/2
print(c)
print("%d"%c)
print("%4d"%c)
print("%f"%c)
print("%3.1f"%c)
print("%6.2f"%c)
```



```
27.5
27
27
27.500000
27.5
27.50
```

- %4d : 정수 4 자리로 표현.
- %6.2f : 실수 6 자리 중 소수 이하는 2 자리. 소수점도 한 자리를 차지하므로 (3 자리.2 자리)가 표현됨.

## 7) 출력 서식 – format

'{}'.format(value) : 출력 문자열 안의 원하는 위치에 {} 로 표시하고 format( ) 의 인수로 value 를 전달.

• 예 1)

```
print("오늘은 {}월 {}일 입니다.".format(6, 26))
```



오늘은 6월 26일 입니다.

• 예 2)

```
print("오늘은 {:d}월 {:2f}일 입니다".format(6, 26))
```



오늘은 6월 26.00일 입니다

• 예 3)

```
a=['cos', 'pro', 'python']
print('{} {} {}'.format(a[0], a[1], a[2]))
print('{0} {1} {2}'.format(*a))
```



```
cos pro python
cos pro python
```

## 2. 연산자

### 1) 산술 연산자

연산자	의미
+, -, *, /	더하기, 빼기, 곱하기, 나누기
//	나눈 몫
%	나눈 나머지
**	거듭제곱

문자열+문자열	두 문자열을 결합
문자열*숫자	숫자만큼 문자열을 반복

예)

```
print("파이썬" + "신나요")  ➡ 파이썬신나요
print("파이썬"*5)           ➡ 파이썬파이썬파이썬파이썬파이썬
```

## 2) 비교 연산자

연산자	의미
==	같다
!=	같지 않다
>	크다
>=	크거나 같다
<	작다
<=	작거나 같다

## 3) 논리 연산자

연산자	의미
조건 1 and 조건 2	조건 1 과 조건 2 가 모두 참이어야 참(True) 그렇지 않으면 거짓(False)
조건 1 or 조건 2	조건 1 과 조건 2 중 하나만 참이면 참(True) 모두 거짓일 때만 거짓(False)
not x	x 가 아니다. (즉, x 가 참이면 거짓, x 가 거짓이면 참)

## 3. 문자열 자르기(Slice)

### 1) 문자열 인덱스

문자열 인덱스 번호	P	Y	T	H	O	N
앞에서 부터	0	1	2	3	4	5
뒤에서 부터	-6	-5	-4	-3	-2	-1

### 2) 문자열 자르는 형식 : 문자열 변수 이름[Start:Stop:Step]

- Start 인덱스 위치부터 시작하여 Step 만큼씩 증가하여 Stop-1 인덱스의 위치까지 자름.
- Start 를 생략하면 문자열의 처음부터 가져옴.

- Stop 을 생략하면 문자열의 끝까지 가져옴.
- Step 을 생략하면 1 만큼씩 증가
- 예)

a = "PYTHON"	
a[2]	T
a[-4]	T
a[1:3]	YT
a[0:5:2]	PTO

## 4. 조건문

### 1) 사용 형식

```

if 조건식 1:
    실행문 1
elif 조건식 2:
    실행문 2
elif 조건식 3:
    실행문 3
...
else:

```

➔ 파이썬에서 조건문, 함수 등을 사용할 때는 시작, 끝을 알려주는 괄호를 사용하지 않고 4 칸 들여쓰기를 사용해서 조건문, 함수에 속하는 명령문을 나타낸다.

### 2) 사용 예

```

p=int(input("점수 입력"))
if p>=800:
    print(p, "점은 교수~!")
elif 600 <= p < 800:
    print(p, "점은 보통~!")
else:
    print("노력하세요~")

```



점수 입력 900  
900 점은 교수~!