

파이썬을 활용한 인공지능 체험하기

1주: 인공지능 개요 , 개발환경 소개, 연산자



한 양 대 학 교 ERICA

- 수업소개
- 인공지능 개요
- 파이썬 및 개발환경
- 파이썬 입출력

차시	학습내용	강의일자	수업방법	학생 준비물
1 ~ 2	1. 인공지능의 개요와 개발 환경 소개 2. 파이썬 변수와 기본자료형, 표준 입·출력 3. 입출력	2022-05-14	이론 및 실습	필기도구 및 노트
3 ~ 5	1. 파이썬 조건문 2. 파이썬 반복문(while) 3. 파이썬 반복문(for): range함수이용	2022-05-21	이론 및 실습	필기도구 및 노트
6 ~ 8	1. 파이썬 리스트와 튜플 2. 인공지능 소개 및 체험	2022-05-28	이론 및 토론, 실습	필기도구 및 노트

1.인공지능개요

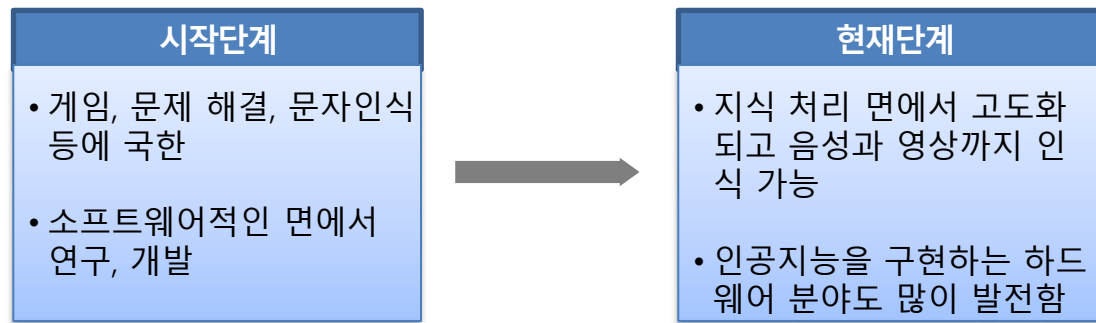
인공지능 소개

□ 인공지능이란 무엇인가?

- 인간이 행동하는 지능적인 작업을 자동으로 하기위한 연구 분야
 - 사람의 생각과 관련된 활동
 - 의사 결정, 문제 해결, 학습 등의 활동을 자동화하는 것
- 인간처럼 사고하고 인간처럼 행동하는 기계장치를 연구하는 분야
 - 지능이 요구되는 일을 할 수 있는 기계를 만드는 예술
- 인간의 지능을 기계로 만드는 연구이며, 구체적인 방법 중 하나가 머신러닝
 - 머신러닝을 구현하는 알고리즘(인공신경망, 딥러닝)
 - 특정 문제를 해결하는 지능적 행동
 - 대부분의 인공지능 접근 방향

□ 인공지능의 발전

- 1950년대 중반 시작하여, 인간처럼 생각, 판단, 학습하는 면에서 많이 발전해옴



- 인공지능이 적용된 지능적인 로봇
 - 악기를 연주하는 단계, 작곡하는 로봇출현
- 감성 인식 로봇의 개발
 - 페퍼: 인간과 대화를 나누고 감정까지 느낌
- 자율주행차 개발
 - 스스로 교통상황을 인식하고 제어하여 목적지까지 주행

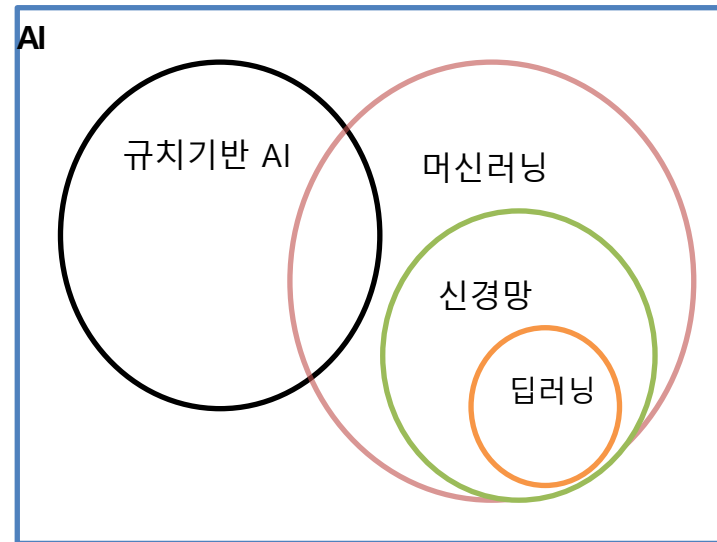
□ 인공지능 장단점

장점	단점
인간의 역할을 대신하여 편리함을 제공	인간의 일자리를 잃게 함
로봇이 24시간 일함으로써 생산성 향상	인공지능에 의지하는 습관에 종속될 가능성
빠르고 올바른 판단에 도움을 줌	인공지능에 의해 감시 받거나 지배 받을 가능성
인간의 실수와 위험을 줄임	연구와 개발이 어려움
개인 비서 역할 담당 가능	인간관계가 줄어들 수 있음

인공지능의 흐름

□ 규칙기반과 신경망 기반 인공지능

- 1950년대 중후반 출범, 비슷한 시기에 시작하여 치열하게 경쟁, 라이벌 관계
- 초기에는 퍼셉트론 신경망 외에는 규칙기반 인공지능 연구가 더 활성화
- 1980년대 신경망 기반이 잠시 우세, 최근 신경망 기반의 딥러닝이 인기
- 이 두 가지는 접근 방법이 전혀 다르므로, 적절한 결합으로 상승효과
- 현재 둘 다 공존, 다만 딥러닝과 알파고로 신경망이 대세로 보이는 효과



인공지능 구성요소

□ 인공지능시스템 구성

- 모델 형태
 - 신경망, 머신러닝, 다층 퍼셉트론, 전문가 시스템 등
- 하드웨어와 소프트웨어
 - GPU, SPARK와 같은 병렬처리 장치와 텐서플로 등
- 학습과 실행에 사용되는 프로그래밍 언어
 - Python, C, Lisp, Prolog 등
- 응용 분야 : 음성인식, 영상인식, 챗봇, 자연어 처리, 지식 처리 등














□ 파이썬 개발

- 1991년 귀도 반 로섬(Guido van Rossum)이 개발
 - 네델란드 암스테르담 대학 컴퓨터 공학 석사
 - 드롭박스 수석 엔지니어

□ 파이썬이란?

- 가장 배우기 쉬운 프로그래밍 언어
- 오픈 소스(Open Source) 프로그래밍 언어
- 가장 널리 쓰임
- 모든 주요 운영체제와 컴퓨터에서 사용 가능

□ IEEE Spectrum 조사 2019년 최고의 프로그래밍 언어

Rank	Language	Type	Score
1	Python	  	100.0
2	Java	  	96.3
3	C	  	94.4
4	C++	  	87.5
5	R		81.5
6	JavaScript		79.4
7	C#	   	74.5
8	Matlab		70.6
9	Swift	 	69.1
10	Go	 	68.0

□ 가장 널리 사용되고 있기 때문

- 생산성을 염두 한 언어 선택

□ 사용 되고 있는 분야

- 웹, 데스크탑
- 인공지능(머신러닝, 딥러닝)
- 임베디드

□ 파이썬 설치

- 다운로드 사이트

- <https://www.python.org/downloads/windows/>



더블클릭

Python Releases for Windows

- [Latest Python 3 Release - Python 3.8.1](#)
- [Latest Python 2 Release - Python 2.7.17](#)

Stable Releases

- [Python 3.8.1 - Dec. 18, 2019](#)

Note that Python 3.8.1 cannot be used on Windows XP or earlier.

- Download [Windows help file](#)
- Download [Windows x86-64 embeddable zip file](#)
- Download [Windows x86-64 executable installer](#)
- Download [Windows x86-64 web-based installer](#)
- Download [Windows x86 embeddable zip file](#)
- Download [Windows x86 executable installer](#)
- Download [Windows x86 web-based installer](#)

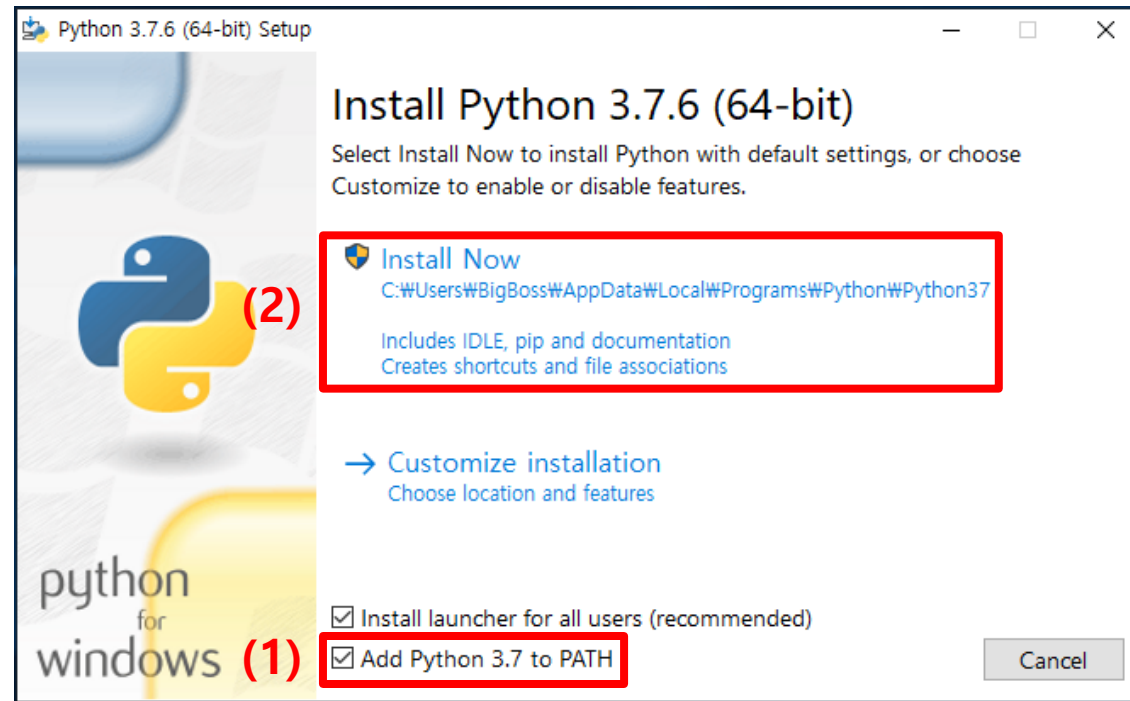
- [Python 3.7.6 - Dec. 18, 2019](#)

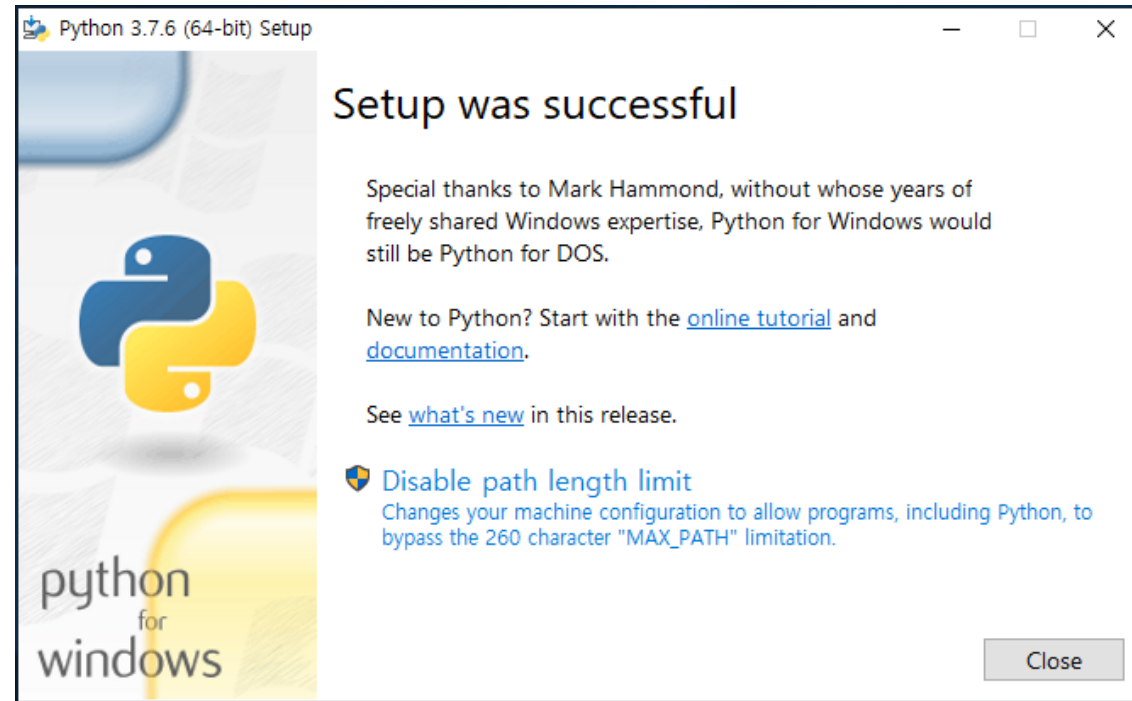
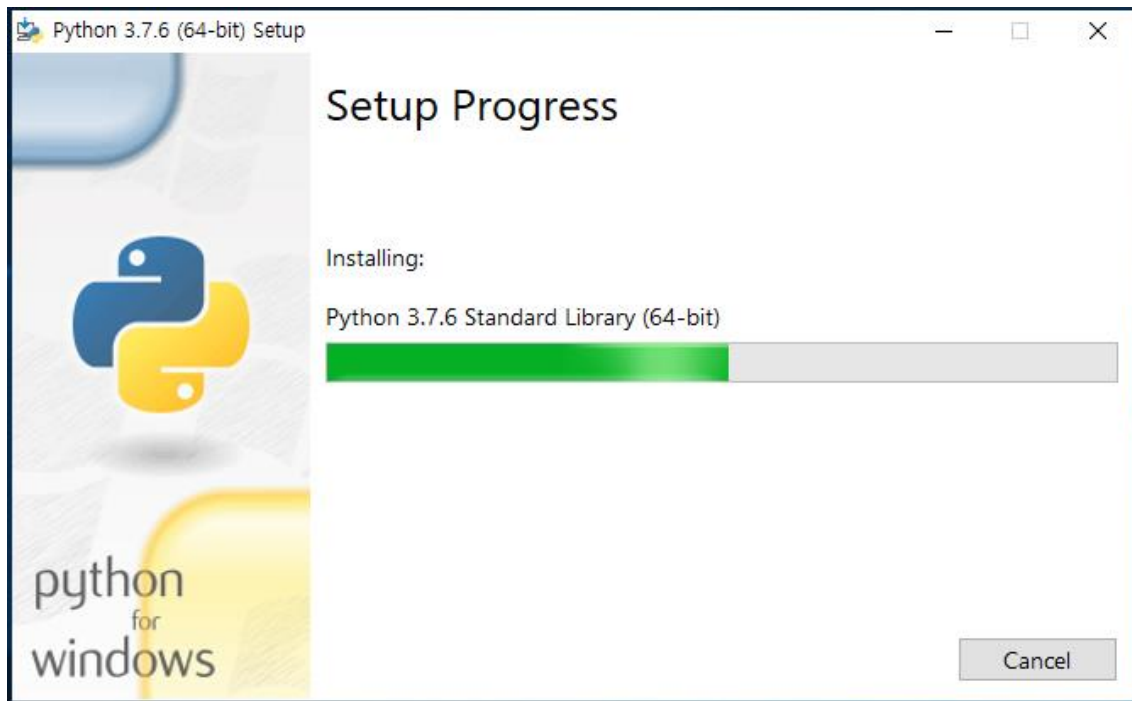
Note that Python 3.7.6 cannot be used on Windows XP or earlier.

- Download [Windows help file](#)
- Download [Windows x86-64 embeddable zip file](#)
- Download [Windows x86-64 executable installer](#)
- Download [Windows x86-64 web-based installer](#)
- Download [Windows x86 embeddable zip file](#)
- Download [Windows x86 executable installer](#)
- Download [Windows x86 web-based installer](#)

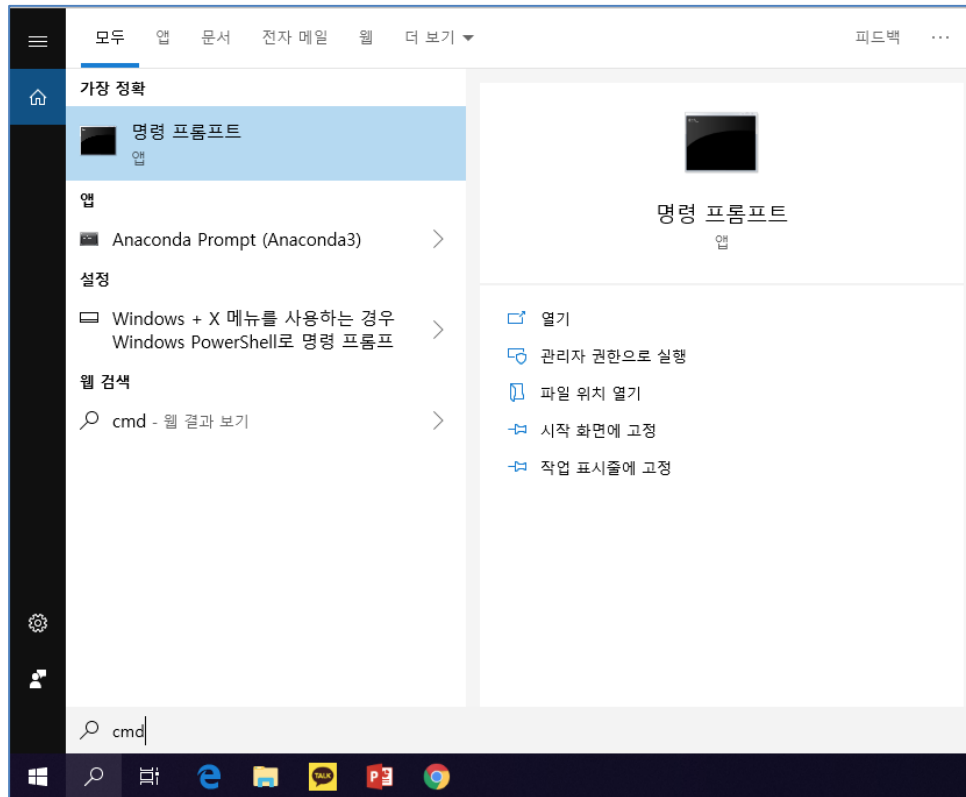
2. 개발환경

※ 반드시 (1)번 'Add Python 3.7 to PATH' 우선 체크하고, (2)번 'Install Now'를 클릭

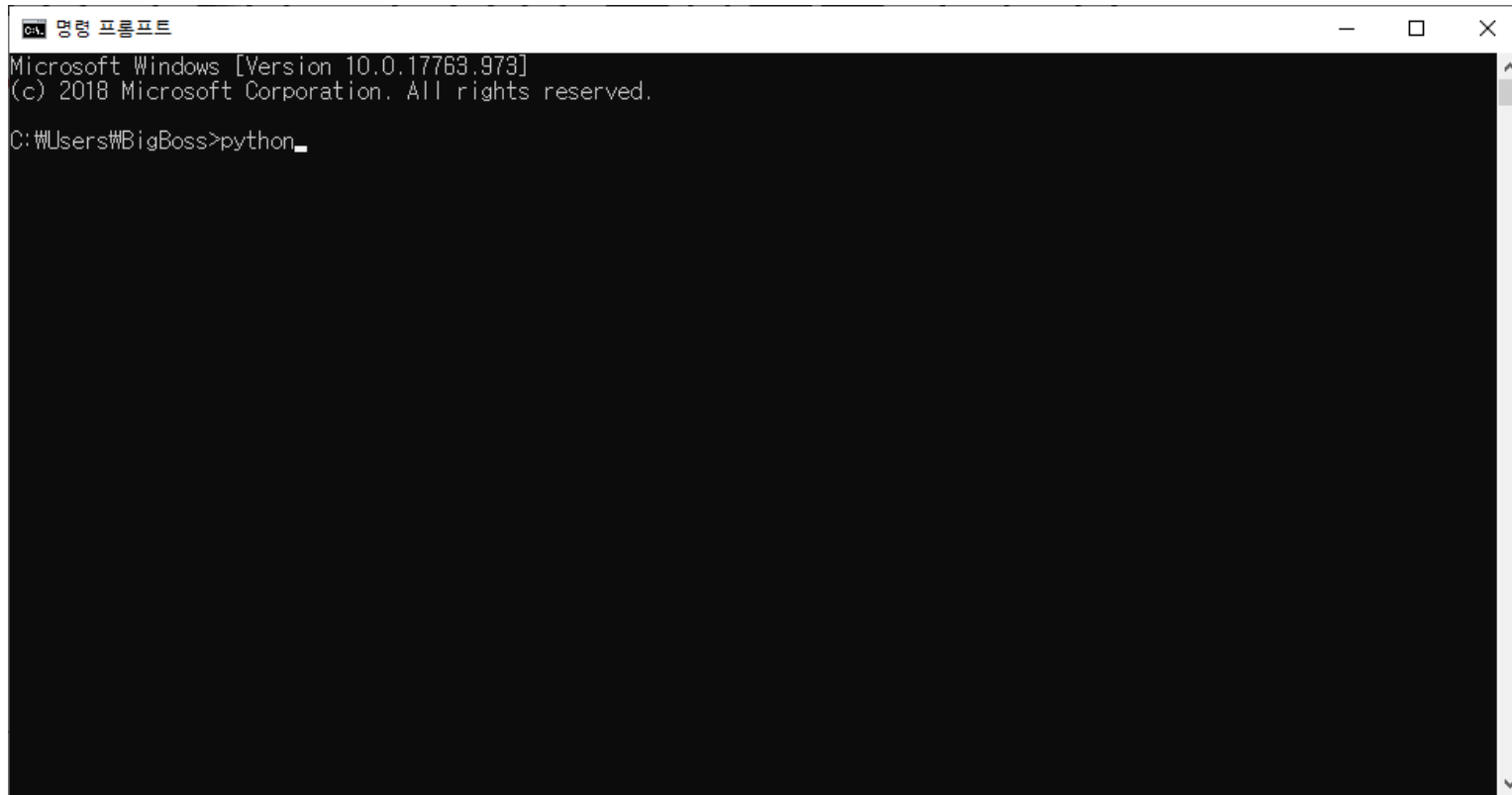




- ❑ 윈도우 검색모드에서 'cmd' 입력
 - '명령 프롬프트' 클릭하여 실행
- ❑ 탐색창에서 'idle(python 3.x xxbit)' 클릭



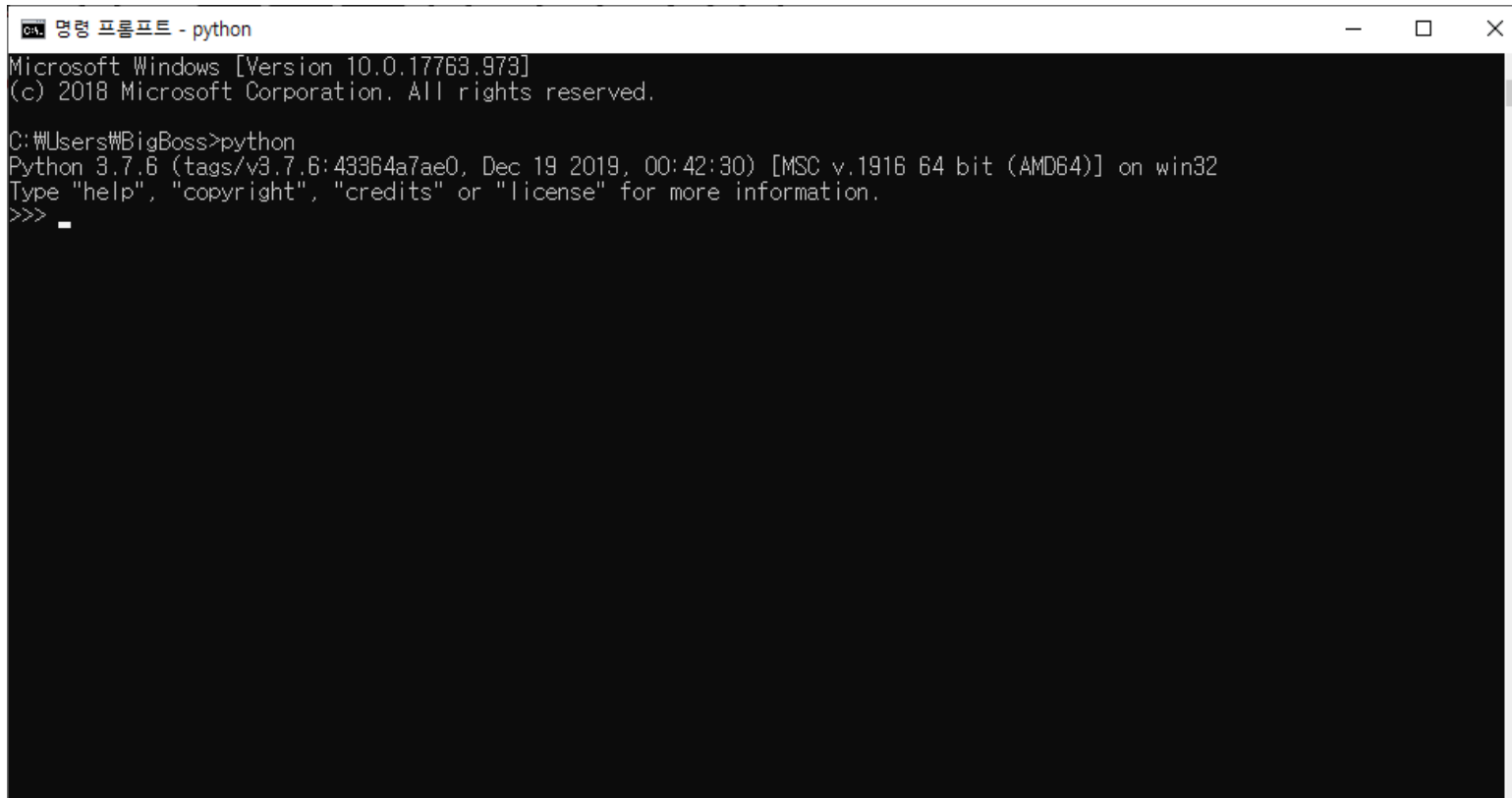
- 검정색 화면의 명령 프롬프트가 열림
 - 'python'이라고 입력 후 'Enter' 키 누름



```
명령 프롬프트
Microsoft Windows [Version 10.0.17763.973]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\BigBoss>python_
```

□ 파이썬 프로그램이 실행됨

- 코드 입력을 기다리는 프롬프트(>>>)가 보임



* 파이썬 실행 종료
Ctrl + Z

3. 입력과 출력

□ Python에서 자주 사용되는 입력 함수는 2가지 이다.

`variable_name=input()`

사용자로부터 입력을 받는다.

`variable_name=input('문자열')`

'문자열'에 해당하는 내용을 출력 후 사용자로부터 입력을 받는다.

[예]

```
>>> name = input()
Gildong
>>> name
'Gildong'
>>>
```

```
>>> name = input('What is your first name? ')
What is your first name? Gildong
>>> name
'Gildong'
>>>
```

input() 함수의 자료형 변환

□ input() 함수의 입력 결과는 '문자열'이다.

– 산술 계산을 하기 위해서는 문자열을 정수 또는 실수 자료형으로 변환하여야 가능

- int() : 문자열 → 정수
- float() : 문자열 → 실수

```
>>> a = input("입력:")
입력:50
>>> print("자료형:", type(a))
자료형: <class 'str'>
>>> print(a+100)
Traceback (most recent call last):
  File "<pyshell#5>", line 1, in
<module>
    print(a+100)
TypeError: can only concatenate str
(not "int") to str
```

```
>>> a = input("입력:")
입력:50
>>> print("자료형:", type(a))
자료형: <class 'str'>
>>> print(int(a)+100)
>>> 150
```

```
>>> a = input("입력:")
입력:50
>>> print("자료형:", type(a))
자료형: <class 'str'>
>>> print(float(a)+100)
>>> 150.0
```

□ 다음 코드의 괄호 안 결과를 예측하여 보자

```
>>> favorite_color = input('What is your favorite color?')
```

What is your favorite color? Blue

```
>>> favorite_color
```

```
(          )
```

다음 input 명령문을 활용하여 스케줄을 작성해보자.

- 1) applicant = input("Enter the applicant's name: ")
- 2) interviewer = input("Enter the interviewer's name: ")
- 3) time = input("Enter the appointment time: ")

```
>>> applicant=input("Enter the applicant's name: ")
Enter the applicant's name: Gildong
>>> interviewer=input("Enter the interviewer's name: ")
Enter the interviewer's name: Minwoo
>>> time=input("Enter the appointment time: ")
Enter the appointment time: 14:00
>>> applicant
'Gildong'
>>> interviewer
'Minwoo'
>>> time
'14:00'
>>>
```


- 입력을 위한 `input()` 함수가 있다면, 출력을 위한 `print()` 함수도 존재한다.
- Python에서 자주 사용되는 출력 함수는 3가지 이다.

<code>print('문자열')</code>	'문자열'을 화면에 출력해준다.
<code>print(variable_name)</code>	변수 <code>variable_name</code> 에 해당하는 값을 화면에 출력해준다.
<code>print('문자열', variable_name)</code>	'문자열'과 변수 <code>variable_name</code> 에 해당하는 값을 연속해서 화면에 출력해준다.

```
> > > print('True False')
True False
> > >
```

```
> number1 = 36
> > > number2 = 42
> > > sum = number1 + number2
> > > print(sum)
78
```

```
> > >
> > > math = 74
> > > english = 87
> > > science = 80
> > > print('수학, 영어, 과학 점수 : ',
math, english, science)
수학, 영어, 과학 점수 : 74 87 80
```

- 컴퓨터를 사용하면 우리가 복잡한 계산을 직접 해야 할 필요는 없음.
- 우리는 정확한 계산을 하도록 알맞은 수식을 올바르게 작성하여 컴퓨터에 지시해야 함

$$\begin{array}{c} 35 \text{ (식의 값)} \\ \hline (12 \quad + \quad 23) \leftarrow \text{(수식)} \\ \begin{array}{ccc} \uparrow & \uparrow & \uparrow \\ \text{피연산자} & \text{연산자} & \text{피연산자} \end{array} \end{array}$$

올바른 수식 작성을 위하여

- 연산자 우선순위
- 수학 시간에 배우지 않았던 새로운 연산자
- 연산의 방향

연산자를 잘 사용하면 프로그램 코드의 길이를 줄일 수 있습니다.

산술은 수에 대한 계산

연산자	기호	사용 예	결괏값
더하기	+	$7 + 4$	11
빼기	-	$7 - 4$	3
곱하기	*	$7 * 4$	28
지수(제곱)	**	$7 ** 4$	2401
나누기	/	$7 / 4$	1.75
나누기(몫)	//	$7 // 4$	1
나머지	%	$7 \% 4$	3
대입(할당)	=	$x = 10$	



더하기(+), 빼기(-), 곱하기(*)

코드 및 실행 결과 1

```
>>> 2+17  
19  
>>> 2-17  
-15
```

코드 및 실행 결과 2

```
>>> 2*17  
34
```



지수 연산자 (**)

코드 및 실행 결과 1

```
>>> 17 ** 2  
289  
>>> 10*2**7  
1280
```

코드 및 실행 결과 2

```
>>> 2**2**3  
256
```



나누기 연산자: '/'와 '//'

코드 및 실행 결과 1

```
>>> 17 / 5 # 실수 나눗셈  
3.4
```

코드 및 실행 결과 2

```
>>> 17 // 5 # 정수 몫  
3
```




나머지 연산자 '%'

코드

```
p = int(input("나누어지는 수를 입력하시오: "))  
q = int(input("나누는 수를 입력하시오: "))  
print("나눗셈의 몫=", p // q)  
print("나눗셈의 나머지=", p % q)
```

실행 결과

나누어지는 수를 입력하시오: 17
나누는 수를 입력하시오: 5
나눗셈의 몫= 3
나눗셈의 나머지= 2

$$\begin{array}{r} 3 \leftarrow \text{몫} \\ 5 \overline{) 17} \\ \underline{15} \\ 2 \leftarrow \text{나머지} \end{array}$$



그렇다면 이 나머지 연산자를 어디에 이용하면 좋을까요?

코드

```
sec = 1000  
min = sec // 60  
remainder = sec % 60  
print(min, '분', remainder, '초')
```

실행 결과

16 분 40 초

오른쪽 : 수식 또는 값

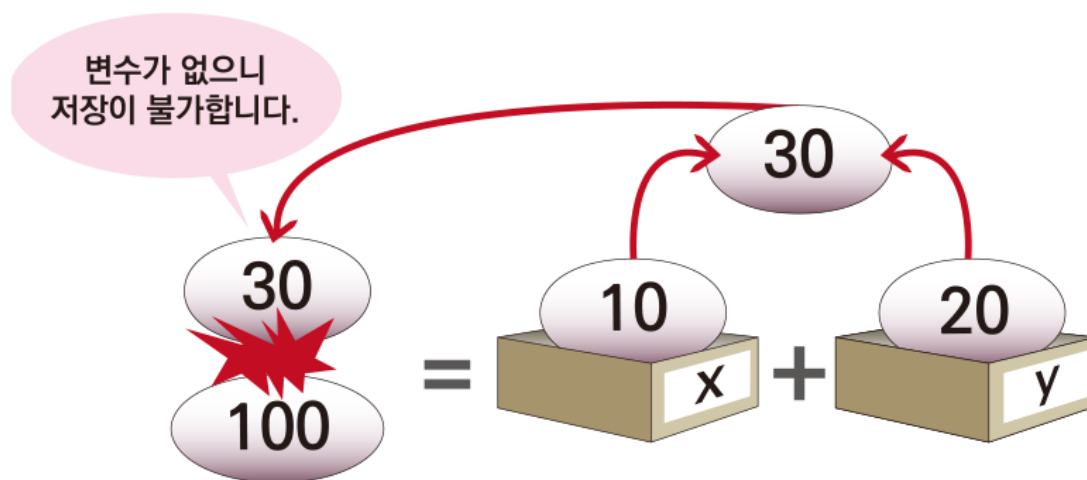
$$X = 100 + 200$$

↑
변수

↑
수식 또는 값

잘못된 예

$$100 = x + y$$





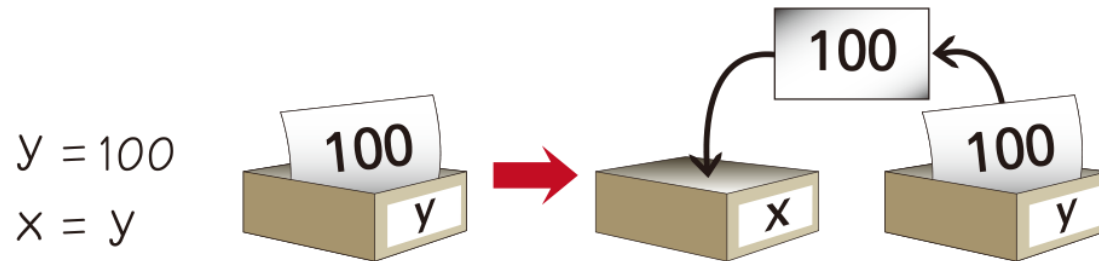
여러 개의 변수에 동일한 값을 저장하는 프로그램을 작성해 보겠습니다.

코드

```
x = y = 100  
print(x)  
print(y)
```

실행 결과

```
100  
100
```



복합 대입 연산자

복합 대입 연산자(compound assignment operator): 대입 연산자와 다른 연산자를 합쳐 놓은 것

서로 의미가 같습니다.

$$x += 2 \longleftrightarrow x = x + 2$$



다음 예제를 통해 복합 대입 연산자 '+=', '-='를 이해해 보겠습니다.

코드

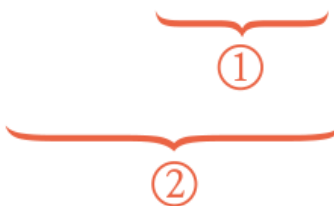
```
x = 1000  
print("초깃값 x=", x)  
x += 2;  
print("x += 2 후의 x=", x)  
x -= 2;  
print("x -= 2 후의 x=", x)
```

실행 결과

```
초깃값 x= 1000  
x += 2 후의 x= 1002  
x -= 2 후의 x= 1000
```

우선순위(precedence) : 하나의 수식에 있는 여러 연산 중에서 어떤 연산을 먼저 수행할지를

결정하는 규칙

$$X + y \times Z$$




사용자로부터 3개의 수를 입력받아서 평균을 출력하는 프로그램을 작성해 보겠습니다.

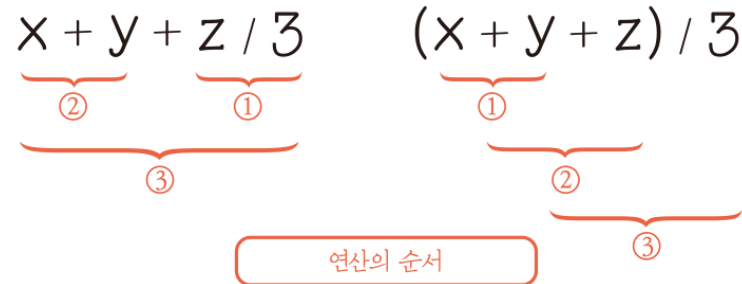
코드

```
x = int(input("첫 번째 수: "))
y = int(input("두 번째 수: "))
z = int(input("세 번째 수: "))

avg = (x + y + z) / 3
print("평균 =", avg)
```

실행 결과

첫 번째 수: 100
두 번째 수: 90
세 번째 수: 80
평균 = 90.0



실습1. 다항식의 계산

- $x=-1, y=3$ 일 때, $(-y)^3+2x^2y$ 의 값을 계산하여 출력하는 프로그램

실행 결과

다항식의 계산 결과 : -21



생각 1 : 프로그램의 순서를 생각해봅니다.

변수 x, y 에 값 저장



다항식을 파이썬의
연산자를 이용하여 표현



계산결과 출력하기

화씨온도를 섭씨온도로 변환하기

- 화씨온도를 입력 받아서 섭씨온도로 바꾸는 프로그램

$$C = (F - 32) \times \frac{5}{9}$$

실행 결과

화씨온도: 100

섭씨온도: 37.77777777777778

- 인공지능 소개
- 개발환경 소개
- 파이썬 입출력