





- 1. Python 특징
- 2. Python 설치
- 3. Python IDE (Visual Studio Code)
- 4. Python 코딩시 주의할 점
- 5. 변수
- 6. 숫자형 데이터
- 7. 문자형 데이터
- 8. 부울 데이터
- 9. 데이터의 연산
- 10. 기본 입출력 함수
- 11. 제어문 (반복문)
- 12. 제어문 (조건문)

# 파이썬이란..?

파이썬은 인터프리터 언어로, 범용 프로그래밍 언어입니다.

파이썬은 간결하고 가독성이 좋은 문법으로, 데이터분석, 인공지능등 다양한 분야에서 사용되고 있습니다.

- 1990년 암스테르담의 귀도 반 로섬이 개발한 언어
- 자신이 좋아하는 코미디쇼에서 이름을 따왔음
- 파이썬은 인터프리터 언어
- 구글의 소프트웨어의 많은 부분이 파이썬으로 작성
- 인스타그램,드롭박스 등도 파이썬으로 제작
- 타 언어에 비하여 배우기가 쉬움



TIOBE INDEX

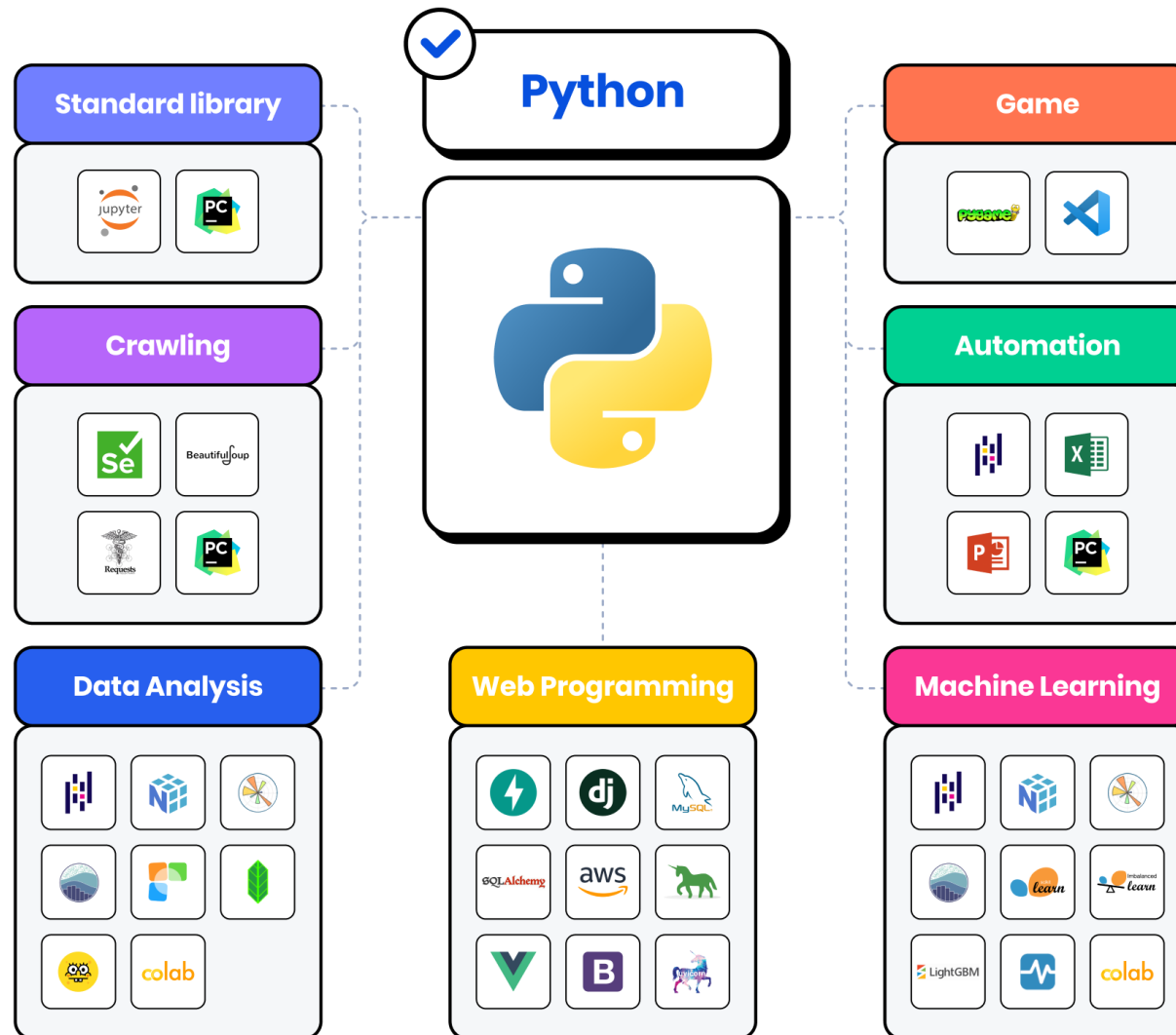


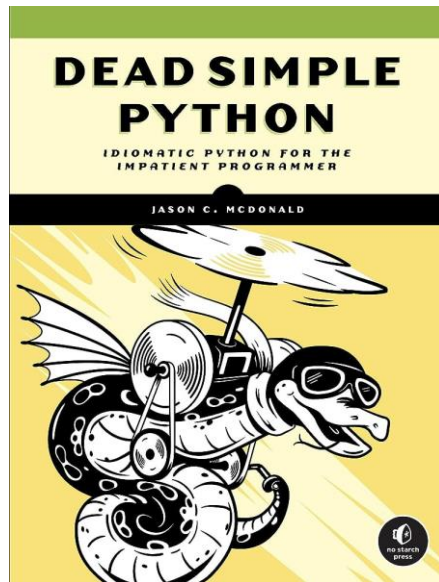
stackoverflow 2023 설문 결과

# 파이썬의 특징

- 문법이 쉽고 간결함
- 무료
- 빠른 개발속도
- 다양한 패키지제공 (Numpy, Pandas, Flask, ...)

**“Life is too short, You need python”**





```
print("Hello, Python!")
```

```
#include <stdio.h>
```

```
int main()
{
    printf("Hello, C!\n");

    return 0;
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Hello_World
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello C#!");
        }
    }
}

public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, Java!");
    }
}
```

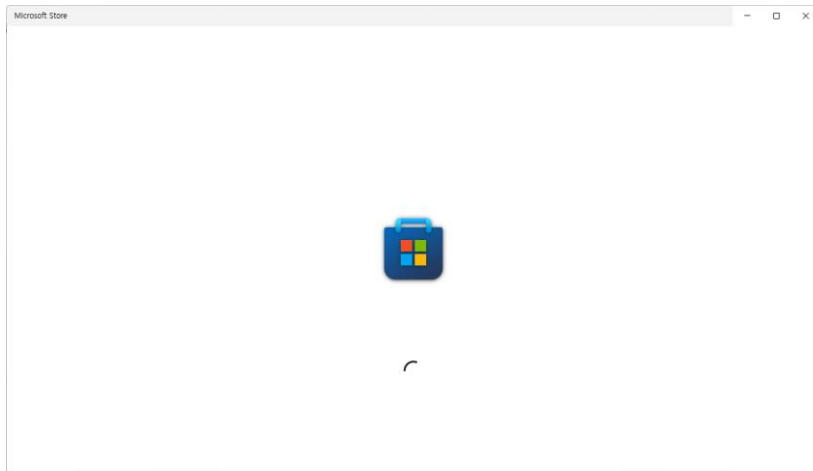
# 파이썬 3 설치

Python 실행

```
python
```

Python 종료

```
exit() 또는 quit()
```



'python'은(는) 내부 또는 외부 명령, 실행할 수 있는 프로그램, 또는 배치 파일이 아닙니다.

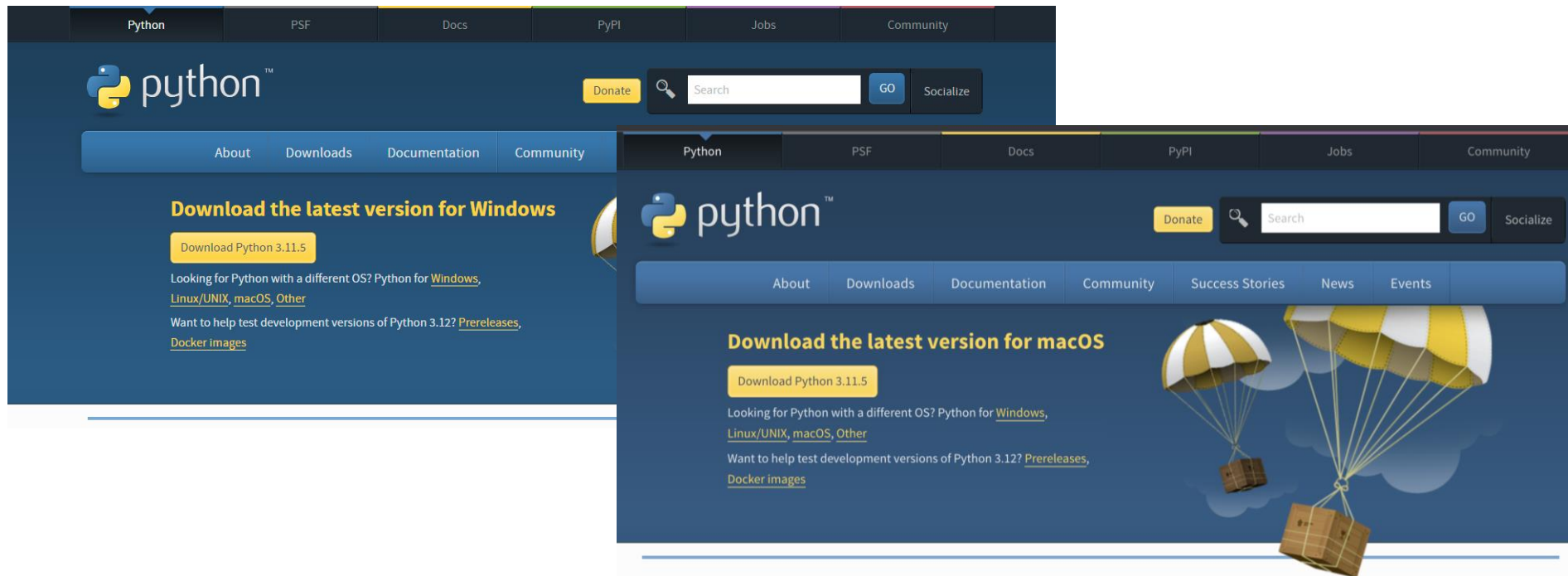
'python' 용어가 cmdlet, 함수, 스크립트 파일 또는 실행할 수 있는 프로그램 이름으로 인식되지 않습니다. 이름이 정확한지 확인하고 경로가 포함된 경우 경로가 올바른지 검증한 다음 다시 시도하십시오.

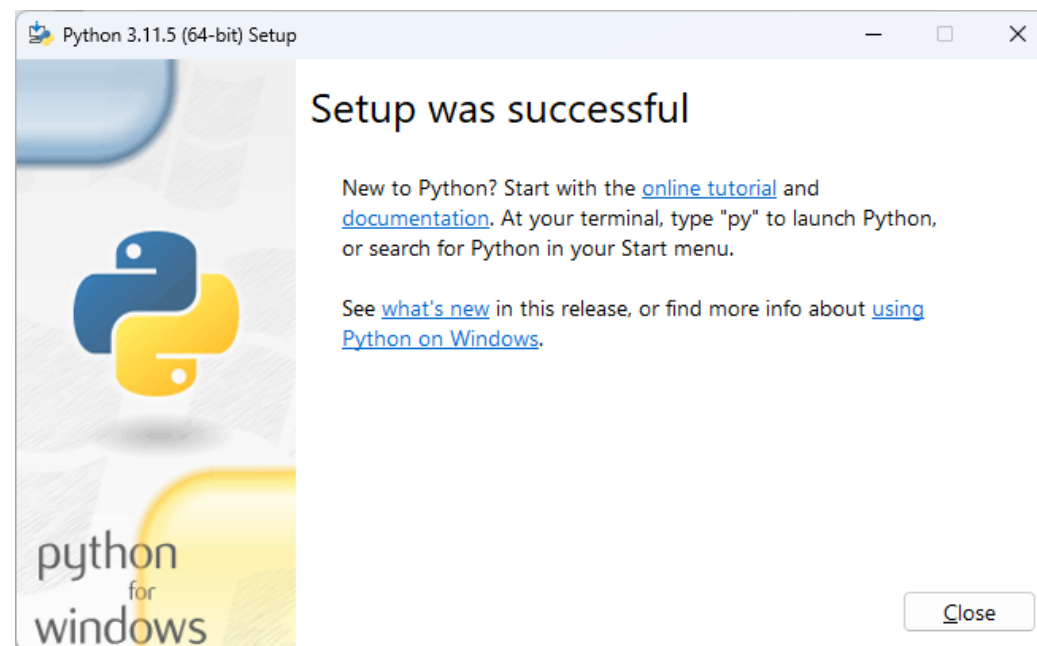
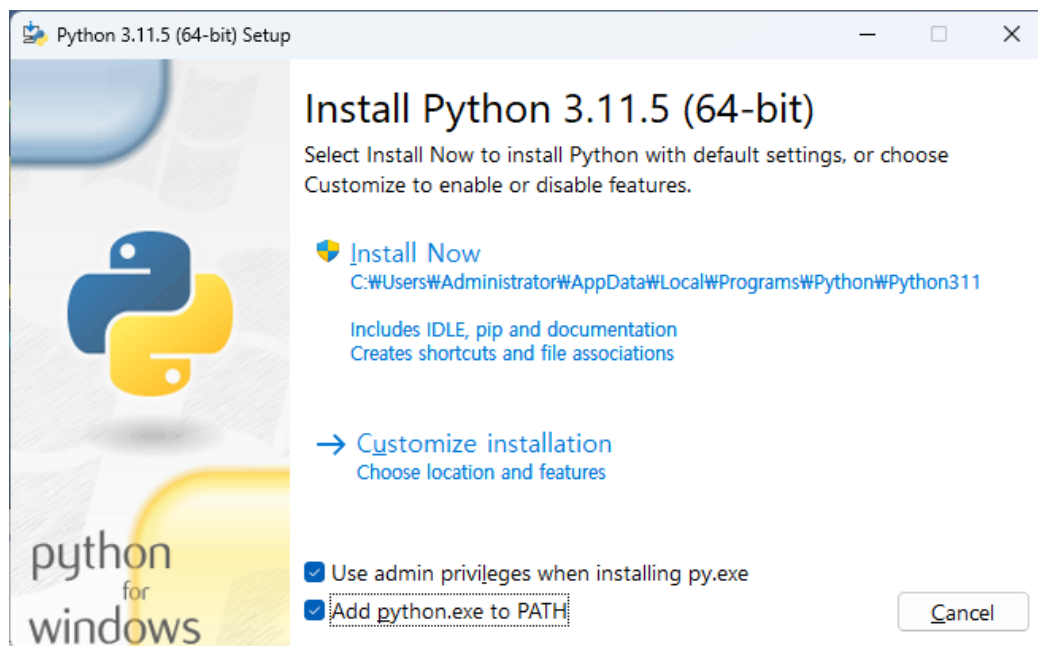
```
zsh : python: command not found
```

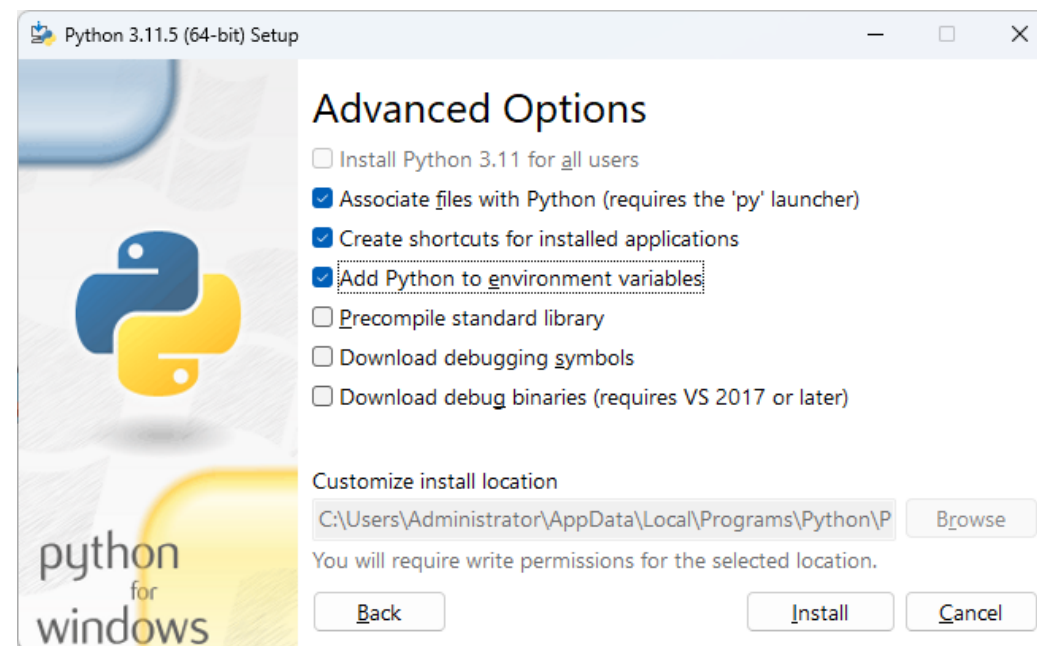
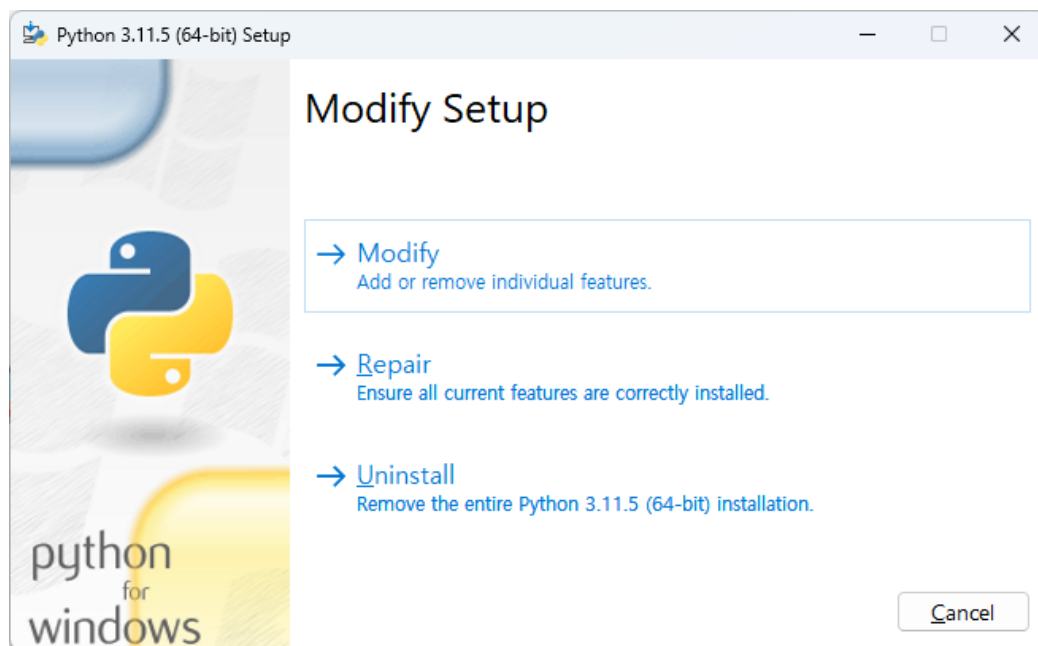


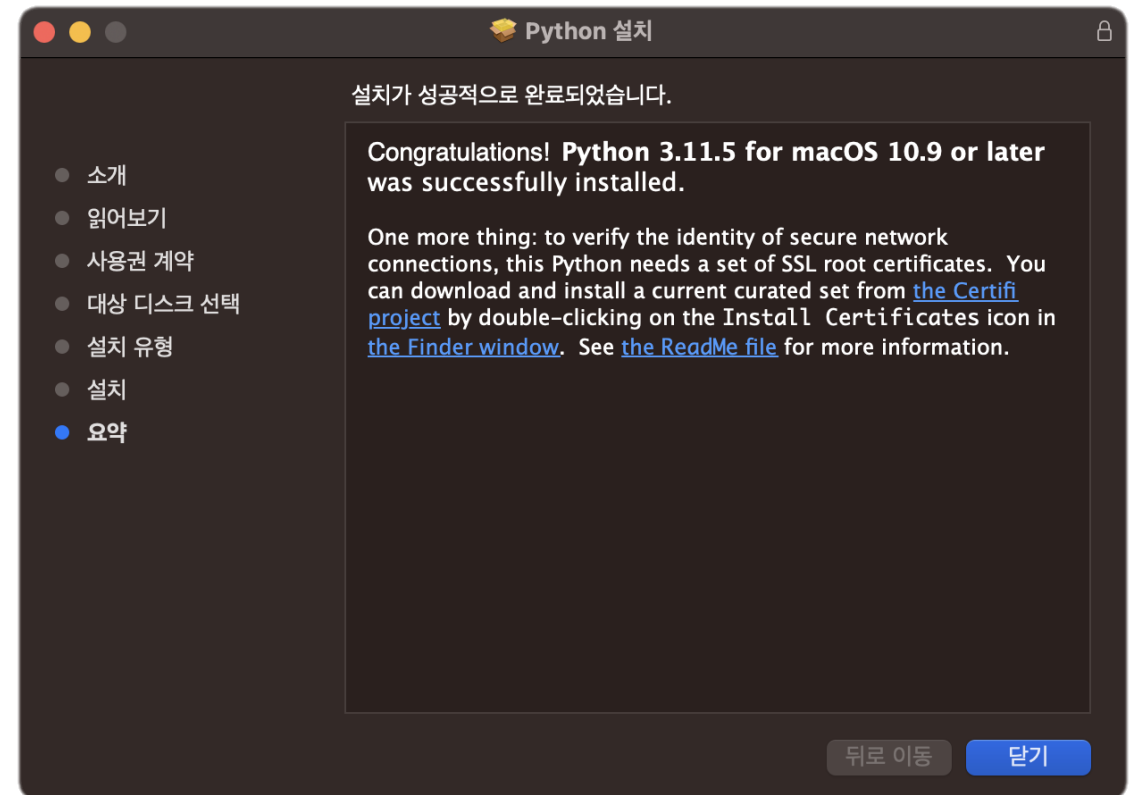
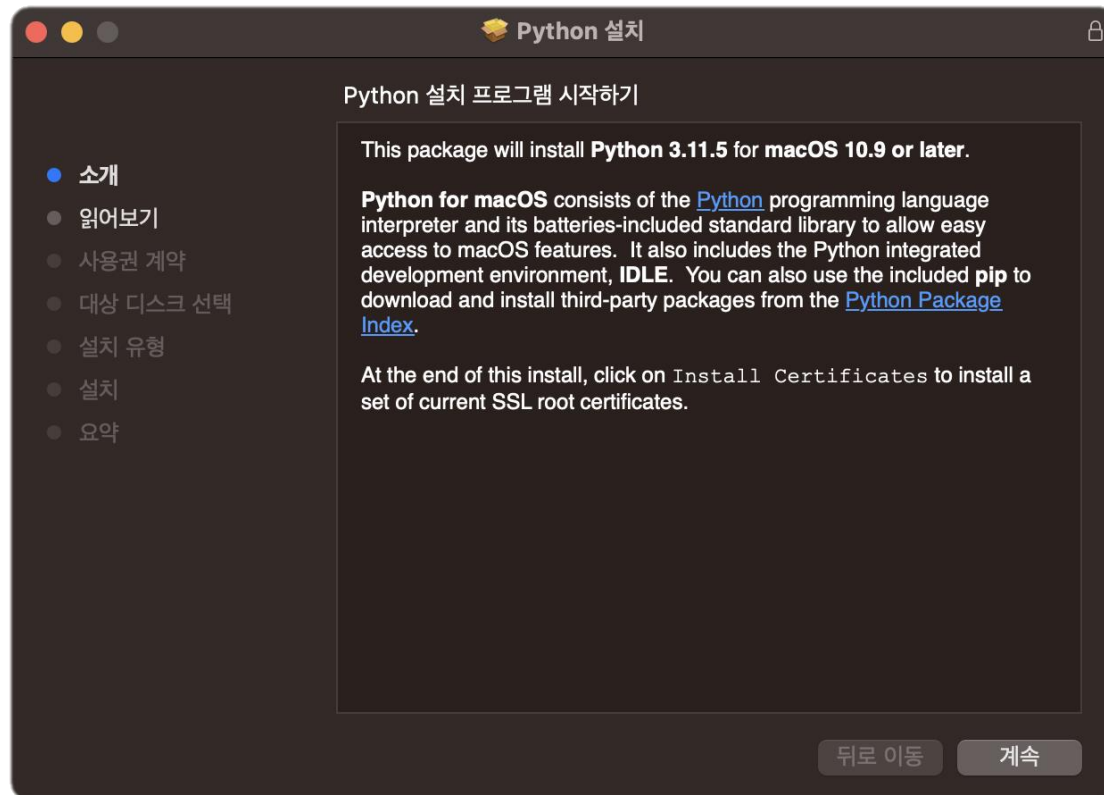
# 파이썬 3 설치

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









```
echo 'alias python="python3"' >> ~/.zshrc && source ~/.zshrc
```

```
echo 'alias pip="pip3"' >> ~/.zshrc && source ~/.zshrc
```

# 파이썬 3 설치

Python 설치 버전 확인

```
C:\Users\Administrator> python --version  
  
Python 3.11.5
```

Python 실행

```
C:\Users\Administrator> python  
  
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023,  
14:38:34) [MSC v.1936 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license"  
for more information.  
>>>
```

# 파이썬 IDE



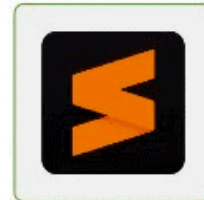
PyCharm



Visual Studio Code



Sublime Text



Vim



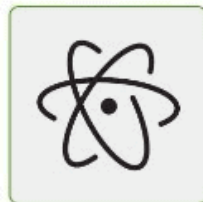
GNU Emacs



Spyder



Atom



Jupyter



Eclipse



IntelliJ IDEA



Notepad++



- 통합개발환경(integrated development environment, IDE):  
코딩, 디버그, 컴파일, 배포 등 프로그램 개발에 관련된 모든 작업을 하나의 프로그램 안에서 처리하는 환경을 제공하는 소프트웨어

```
C:\Users\Administrator> mkdir python_study
```

```
C:\Users\Administrator> cd python_study
```

```
C:\Users\Administrator\python_study> code .
```



## Python Extension Pack v1.7.0

Don Jayamanne |  6,157,476 | ★★★★★ (15)

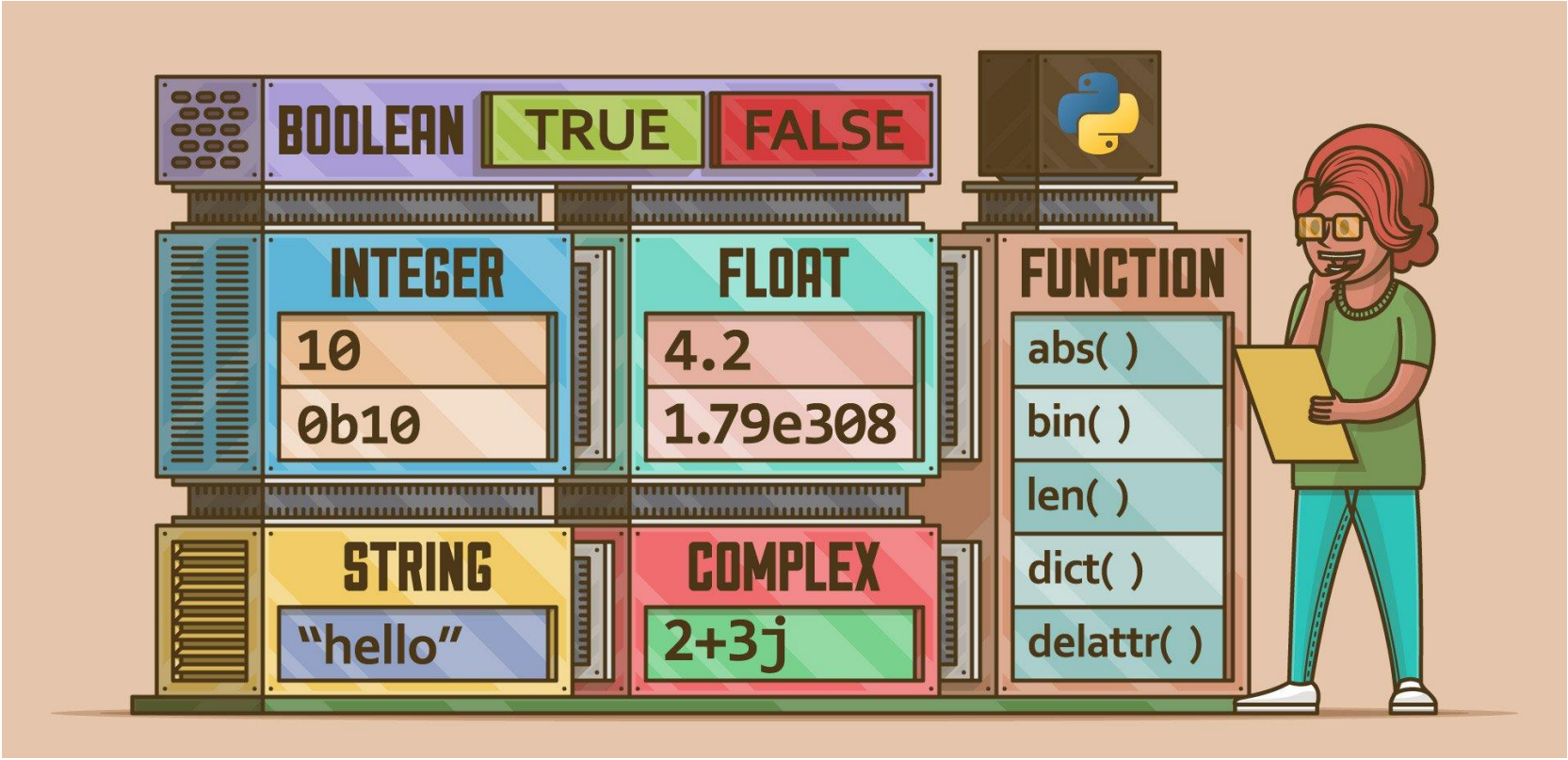
Popular Visual Studio Code extensions for Python

[Install](#) 

# 파이썬 코딩 시 주의할 점

- 들여쓰기      Python에서는 들여쓰기는 선택사항이 아닌 필수!
- 네이밍 컨벤션      변수, 함수, 클래스 등의 이름을 지을 때는 네이밍 컨벤션을 따를 것! (PEP8)
- 주석      충분한 주석은 코드의 이해와 수정 시 큰 도움이 됩니다.
- 모듈과 패키지      Python에서는 코드를 모듈과 패키지로 구성할 수 있습니다.
- 예외 처리      예외 처리를 제대로 하지 않으면 프로그램이 비정상적으로 종료될 수 있습니다.





# 변수

데이터를 저장하기 위한 메모리 공간에 이름을 붙인 것

동적 타이핑 언어인 Python은 변수를 선언할 때 데이터 타입을 지정하지 않아도 됩니다.

**“Life is too short, You need python”**



```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34)
[MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more
information.
```

```
>>> variable = "hello world"
>>> type(variable)
<class 'str'>
>>> variable
hello world
>>> variable = 3
>>> type(variable)
<class 'int'>
>>> variable
3
```



## 1. 소문자와 밑줄 사용

- 변수 이름은 항상 소문자로 시작하고, 여러 단어의 경우 단어 사이에 밑줄('\_')을 사용해야 합니다.
- 예를 들어, `my_variable`이나 `user_name`과 같이 사용합니다.

## 2. 의미 있는 이름 사용

- 변수 이름은 해당 변수의 역할이나 의미를 명확하게 나타내야 합니다.
- 가독성을 높이기 위해 짧고 알기 쉬운 이름을 사용하는 것이 좋습니다.

## 3. 상수

- 상수 변수의 이름은 대문자와 밑줄로 작성하며, 가독성을 높이기 위해 모든 문자를 대문자로 작성합니다.
- 예를 들어, `PI`나 `MAX_LENGTH`와 같이 사용합니다.

## 4. 단일 문자 변수

- 보통 루프 변수나 간단한 임시 변수에만 단일 문자 변수를 사용하며, 변수의 역할을 나타내는 이름을 사용하는 것이 좋습니다.
- 예를 들어, `i`나 `temp`와 같이 사용합니다.

## 5. 파이썬 문법에 사용되는 단어는 사용 불가능

- `True`, `False`, `None`, `and`, `as`, `assert`, `break`, `class`, `continue`, `def`, `del`, `elif`, `else`, `except`, `finally`, `for`, `from`, `global`, `if`, `import`, `in`, `is`, `lambda`, `nonlocal`, `not`, `or`, `pass`, `raise`, `return`, `try`, `while`, `with`, `yield` 등...

# 숫자형

- 정수 (int)                    양의 정수, 음의 정수, 0
- 실수 (float)                소수점이 포함된 숫자
- 복소수 (complex)        허수 ( $\sqrt{-1}$ )

항목	사용 예
정수	123, -456, 0
실수	12.345, -12.345, 3.4e10
복소수	1+2j, -3j
8진수	0o34, 0o25
16진수	0x12, 0xFF

```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34)
[MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more
information.
```

```
>>> type(1)
<class 'int'>
>>> type(0)
<class 'int'>
>>> type(-1)
<class 'int'>
```

```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34)
[MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more
information.
```

```
>>> type(1.51)
<class 'float'>
>>> type(0.0)
<class 'float'>
>>> type(-5.99)
<class 'float'>
```

```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34)
[MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more
information.
```

```
>>> type(2+3j)
<class 'complex'>
>>> type(0j)
<class 'complex'>
>>> type(3-4j)
<class 'complex'>
```





# 문자열

- 문자나 단어 등으로 이루어진 문자들의 집합
- 문자열은 큰 따옴표(" ") 또는 작은 따옴표(' ')로 생성

문자열 내부에 작은 따옴표를 사용하고 싶다면? I'm Iron man

문자열 내부에 큰 따옴표를 사용하고 싶다면? He said "I am Iron man"

문자열 내부에 큰 따옴표와 작은 따옴표 모두를 사용하고 싶다면? He said "I'm Iron man"

```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34)
[MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more
information.
```

```
>>> type("Hello, world!")
<class 'str'>
>>> type('Hello, world!')
<class 'str'>
>>> type('''문자열을 따옴표 세 개로 감싸면,
...     여러 줄로 입력할 수 있습니다.
...     이렇게요!''')
<class 'str'>
```



```
>>> str1 = "Hello, "  
>>> str2 = "Python!"  
>>> str3 = str1 * 2 + str2  
>>> len(str3)  
  
>>> str3[4]  
  
>>> str3[0:5]  
  
>>> str3[:5]  
  
>>> str3[14:]  
  
>>> str3[4:8]
```



```
>>> 'Height: %d' % 180 # 숫자 데이터 대입
>>> 'Name: %s' % '홍길동' # 문자 데이터 대입
>>> 'Name: %s, Age: %d' % ('홍길동', 30) # 여러 데이터 대입
```

```
>>> 'Height: {0}'.format(180) # 숫자 데이터 대입
>>> 'Name: {0}'.format('홍길동') # 문자 데이터 대입
>>> 'Name: {0}, Age: {1}'.format('홍길동', 30) # 여러 데이터 대입
```

```
>>> 'Height: {height}, Name: {name}'.format(height=180, name='홍길동')
```

```
>>> # 0번째 인자를 소수점 둘째자리까지
>>> '{0:0.2f}'.format(1.2345)
>>> # 0번째 인자를 소수점 셋째자리까지, 문자열의 길이 최소 10
>>> '{0:10.3f}'.format(12.2345)

>>> 'Name: {0}, Height: {1:0.0f}'.format('홍길동', 180.123)
```

```
>>> name = '홍길동'
>>> '저는 {}입니다.'.format(name)
>>> f'저는 {name}입니다.'
```

```
>>> '    left_space'.lstrip()
>>> 'right_space    '.rstrip()
>>> '    space    '.strip()
```

```
>>> 'string_message'.replace('i', 'o')
>>> "can't    Strip    Space".replace(" ", "")
>>> '    space    '.strip()
```

```
>>> '덩기덕쿵더러러러쿵기덕쿵더러러러'.count('쿵')
```



```
>>> '-'.join('자축인묘진사오미신유술해')
```

```
>>> '자축인묘진사오미신유술해'.find('술')  
>>> '자축인묘진사오미신유술해'.index('술')
```

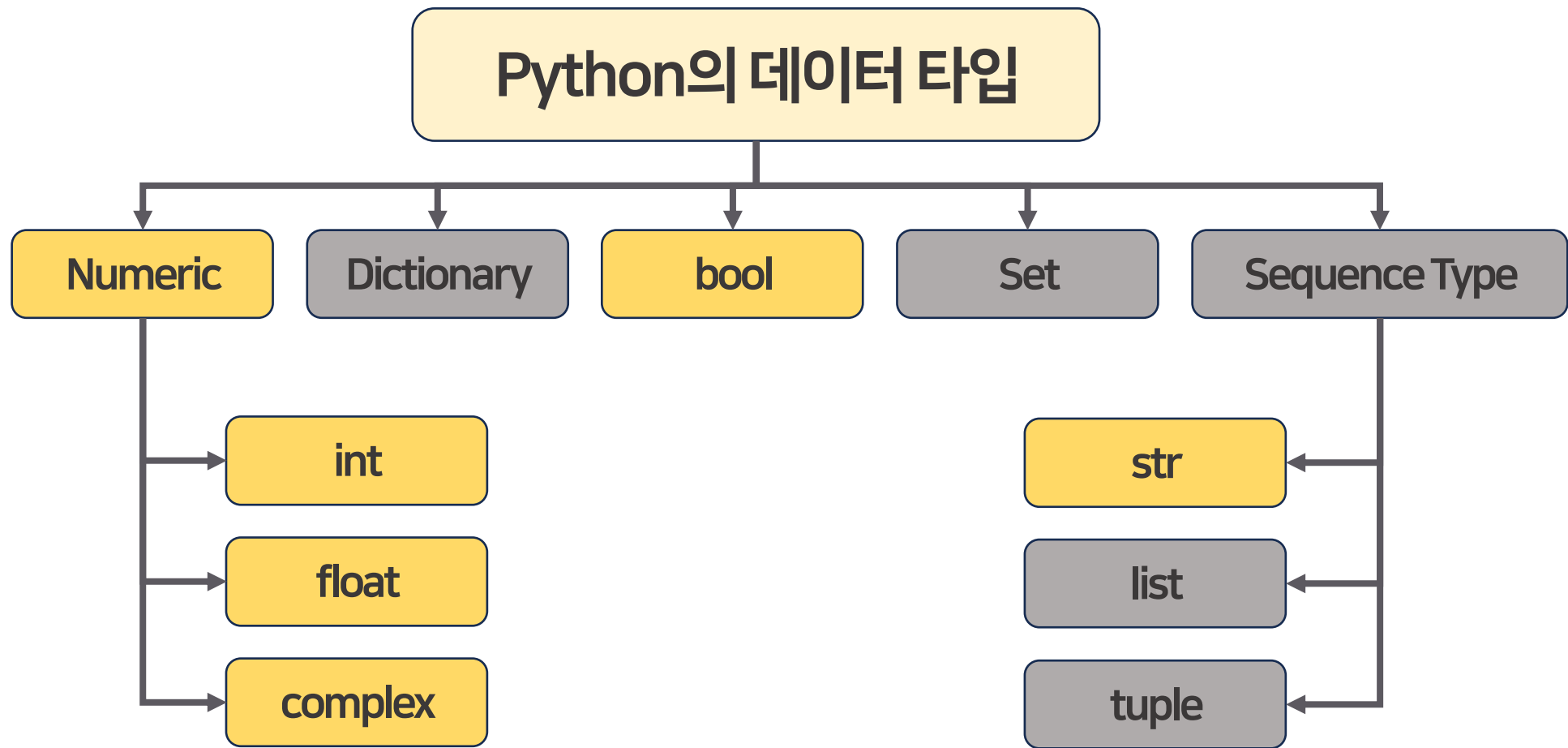
```
>>> 'korea'.upper()  
>>> 'KOREA'.lower()  
>>> 'korea'.capitalize()
```

# 불(Bool) 자료형

- True(참)와 False(거짓)의 두 가지 값을 가지고 있는 자료형

```
>>> type(True)
<class 'bool'>
>>> type(False)
<class 'bool'>
```





기호	연산자	예시	결과값
+	더하기	11 + 3.0	14.0
-	빼기	10 - 11	-1
*	곱하기	2.5 * 4.0	12.0
/	나누기	9 / 2	4.5
//	몫 반환	9 // 2	4
%	나머지	5.5 % 2	1.5
**	제곱	2 ** 5	32

```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34)
[MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more
information.
```

```
>>> data_a = 2 / 3
>>> data_b = (5 / 3) - 1
>>> data_a

>>> data_b
```

기호	예시	결과
+=	a = 7 a += 2	a는 9를 참조합니다.
-=	b = 7 b -= 2	b는 5를 참조합니다.
*=	c = 7 c *= 2	c는 14를 참조합니다.
/=	d = 7 d /= 2	d는 3.5를 참조합니다.
//=	e = 7 e //= 2	e는 3을 참조합니다.
%=	f = 7 f %= 2	f는 1을 참조합니다.
**=	g = 7 g **= 2	g는 49를 참조합니다.





```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34)
[MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more
information.
```

```
>>> user_input = input()
안녕하세요
>>> user_input
'안녕하세요'
```



```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34)
[MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more
information.
```

```
>>> user_name = input("이름을 입력하세요\n")
이름을 입력하세요
손흥민
>>> user_name
'손흥민'
```

```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34)
[MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more
information.
```

```
>>> user_age = input("나이를 입력하세요\n")
나이를 입력하세요
20
>>> user_age
'20'
>>> int(user_age)
20
```



```
>>> help(input)
```

```
Help on built-in function input in module builtins:
```

```
input(prompt='', /)
```

```
    Read a string from standard input.  The trailing newline is stripped.
```

```
    The prompt string, if given, is printed to standard output without a  
    trailing newline before reading input.
```

```
    If the user hits EOF (*nix: Ctrl-D, Windows: Ctrl-Z+Return), raise EOFError.  
    On *nix systems, readline is used if available.
```



```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34)
[MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more
information.
```

```
>>> my_number = 24
>>> my_str = "python"
>>> print(my_number)
24
>>> print(my_number * 2)
48
>>> print(my_str, "is", "interesting")
'python is interesting'
```



```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34)
[MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more
information.
```

```
>>> print("이름 :\t손흥민\n직업 :\t축구선수")
```

```
이름 :   손흥민
직업 :   축구선수
```

```
>>> print_msg = '''이름 :\t이강인
... 직업 :\t축구선수'''
```

```
>>> print_msg
'이름 :\t이강인\n직업 :\t축구선수'
```

```
>>> print(print_msg)
```

```
이름 :   이강인
직업 :   축구선수
```

```
>>> help(print)
```

```
Help on built-in function print in module builtins:
```

```
print(*args, sep=' ', end='\n', file=None, flush=False)
```

```
    Prints the values to a stream, or to sys.stdout by default.
```

```
    sep
```

```
        string inserted between values, default a space.
```

```
    end
```

```
        string appended after the last value, default a newline.
```

```
    file
```

```
        a file-like object (stream); defaults to the current sys.stdout.
```

```
    flush
```

```
        whether to forcibly flush the stream.
```

```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34)
[MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more
information.
```

```
>>> print("a", "b", "c", "d")
>>> print("a", "b", "c", "d", sep=", ")
>>> print("a", "b", "c", "d", sep=", ", end="")
```

기호	설명
\'	작은 따옴표 출력
\"	큰 따옴표 출력
\\	역슬래시 출력
\t	탭
\n	줄바꿈
\r	캐리지 리턴

??!

<https://www.acmicpc.net/problem/10926>

준하는 사이트에 회원가입을 하다가 joonas라는 아이디가 이미 존재하는 것을 보고 놀랐다.  
준하는 놀람을 ??!로 표현한다.  
준하가 가입하려고 하는 사이트에 이미 존재하는 아이디가 주어졌을 때,  
놀람을 표현하는 프로그램을 작성하시오.

입력	출력
joonas	joonas??!
입력	출력
baekjoon	baekjoon??!

# 1998년생인 내가 태국에서는 2541년생?!

<https://www.acmicpc.net/problem/18108>

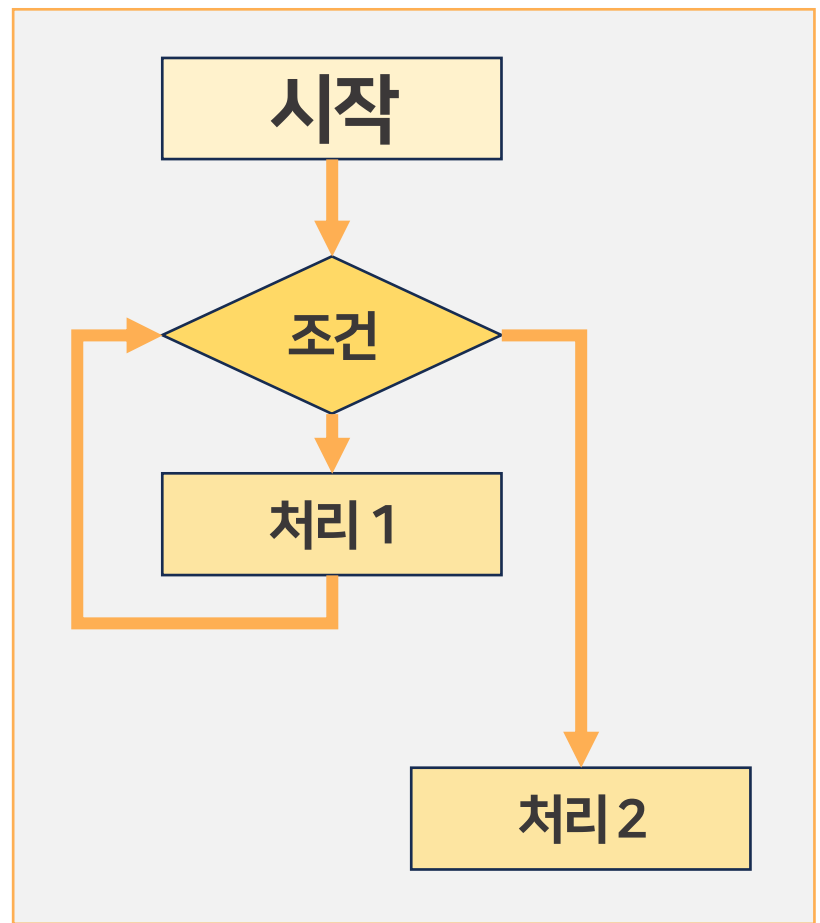
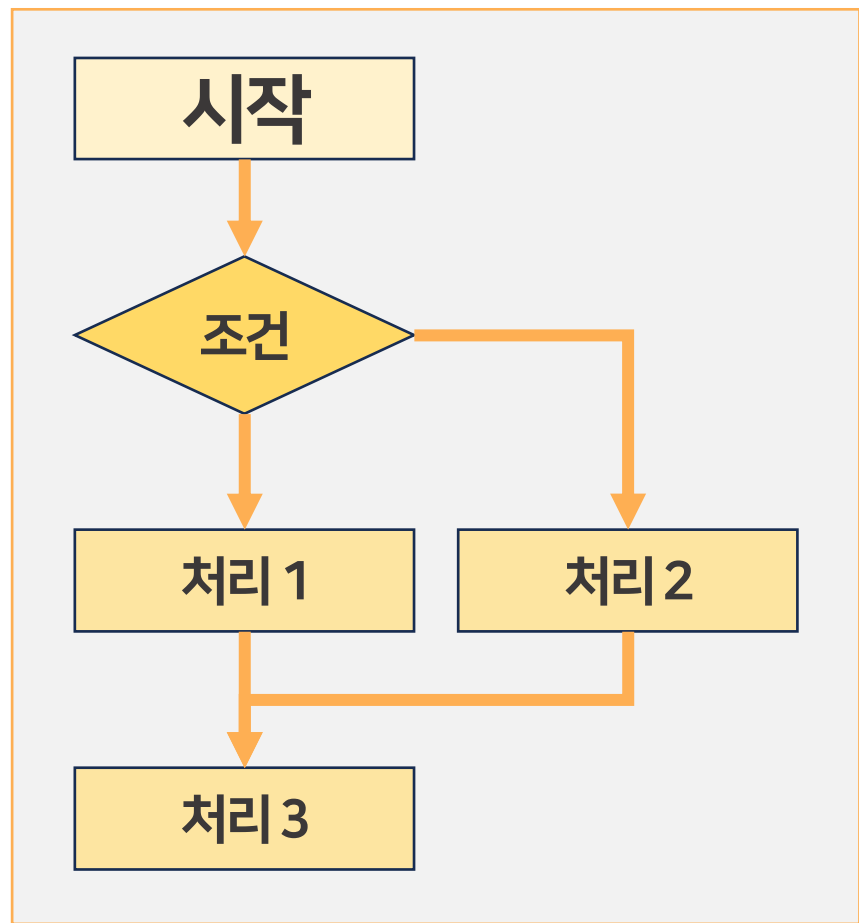
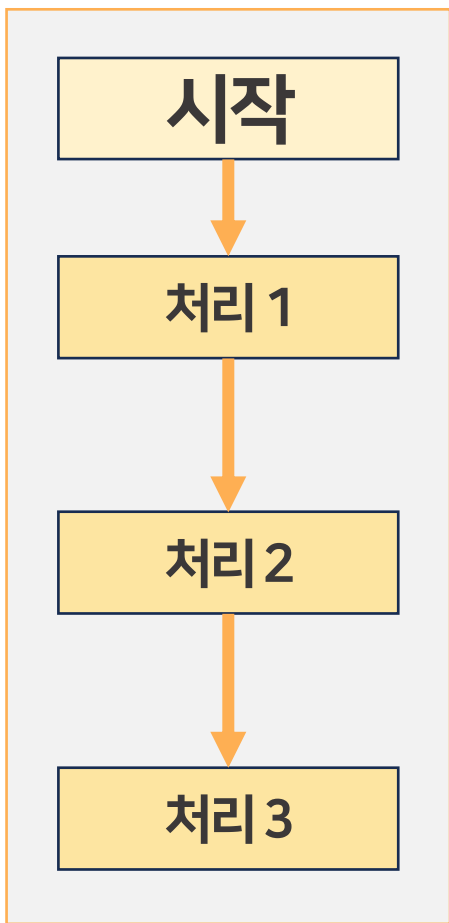
2019년에 ICPC Bangkok Regional에 참가하기 위해 수완나품 국제공항에 막 도착한 팀 레드 시프트 일행은 눈을 믿을 수 없었다. 공항의 대형 스크린에 올해가 2562년이라고 적혀 있던 것이었다.

불교 국가인 태국은 불멸기원(佛滅紀元), 즉 석가모니가 열반한 해를 기준으로 연도를 세는 불기를 사용한다. 반면, 우리나라는 서기 연도를 사용하고 있다.

불기 연도가 주어질 때 이를 서기 연도로 바꿔 주는 프로그램을 작성하시오.

입력	출력
2541	1998







# 조건문 IF

```
if 조건:  
    조건이 True일 때의 동작
```

```
if 조건1:  
    조건1이 True일 때의 동작  
else :  
    조건1이 False일 때의 동작
```

```
if 조건1:  
    조건1이 True일 때의 동작  
elif 조건2:  
    조건2가 True일 때의 동작  
else:  
    모든 조건이 False일 때의 동작
```

자료형	참	거짓
숫자	0이 아닌 모든 숫자	0
문자열	비어있지 않은 모든 문자열	""
bool	True	False

### # 숫자의 참과 거짓

```
print("1 -> ", bool(1))
print("0 -> ", bool(0))
print("-1 -> ", bool(-1))
print("0.0 -> ", bool(0.0))
print("0.1 -> ", bool(0.1))
print("-0.0 -> ", bool(-0.1))
```

### # 문자열의 참과 거짓

```
print("'abc' -> ", bool("abc"))
print("' ' -> ", bool(" "))
print("' ' -> ", bool(""))
```



기호	설명
>	왼쪽이 더 크면 True, 그렇지 않으면 False를 반환
<	왼쪽이 더 작으면 True, 그렇지 않으면 False를 반환
>=	왼쪽이 더 크거나 같으면 True, 그렇지 않으면 False를 반환
<=	왼쪽이 더 작거나 같으면 True, 그렇지 않으면 False를 반환
==	서로 같으면 True, 그렇지 않으면 False를 반환
!=	서로 다르면 True, 그렇지 않으면 False를 반환
not	False면 True, True면 False를 반환
and	둘 다 True면 True, 그렇지 않으면 False를 반환
or	둘 중에 하나라도 True면 True, 그렇지 않으면 False를 반환

# 조건문 IF

```
input_value = input("나이를 입력하세요\n")
age = int(input_value)
```

```
if age >= 20 :
    print("입장 가능한 연령입니다.")
    print("환영합니다.")
```

```
input_value = input("나이를 입력하세요\n")
age = int(input_value)
```

```
if age >= 20 :
    print("입장 가능한 연령입니다.")
    print("환영합니다.")
else :
    print("입장이 불가능합니다.")
```

```
input_value = input("나이를 입력하세요\n")
age = int(input_value)
```

```
if age >= 20 :
    print("입장 가능한 연령입니다.")
    print("환영합니다")
elif age >= 15 :
    print("제한된 입장이 가능합니다")
else :
    print("입장이 불가능합니다.")
```

# 시험성적

<https://www.acmicpc.net/problem/9498>

시험 점수를 입력받아 90 ~ 100점은 A, 80 ~ 89점은 B, 70 ~ 79점은 C, 60 ~ 69점은 D, 나머지 점수는 F를 출력하는 프로그램을 작성하시오.

입력	출력
100	A

# 윤년

<https://www.acmicpc.net/problem/2753>

연도가 주어졌을 때, 윤년이면 1, 아니면 0을 출력하는 프로그램을 작성하시오.  
윤년은 연도가 4의 배수이면서, 100의 배수가 아닐 때 또는 400의 배수일 때이다.

예를 들어, 2012년은 4의 배수이면서 100의 배수가 아니라서 윤년이다. 1900년은 100의 배수이고 400의 배수는 아니기 때문에 윤년이 아니다. 하지만, 2000년은 400의 배수이기 때문에 윤년이다.

입력	출력
2000	1
입력	출력
1999	0

[illegible]





```
x = 0
```

```
while x < 100:  
    print("홍길동")  
    x += 1
```

```
for x in range(0, 100):  
    print("홍길동")
```



```
print("range(10)의 결과 -> ", list(range(10)))  
print("range(5, 10)의 결과 -> ", list(range(5, 10)))  
print("range(1, 10, 3)의 결과 -> ", list(range(1, 10, 3)))
```

```
class range(object)  
| range(stop) -> range object  
| range(start, stop[, step]) -> range object  
|  
| Return an object that produces a sequence of integers from start (inclusive)  
| to stop (exclusive) by step. range(i, j) produces i, i+1, i+2, ..., j-1.  
| start defaults to 0, and stop is omitted! range(4) produces 0, 1, 2, 3.  
| These are exactly the valid indices for a list of 4 elements.  
| When step is given, it specifies the increment (or decrement)
```



`while` 조건:  
    조건이 참일 때의 구문  
    조건에 변화를 주는 구문

`for` 변수 `in` 반복객체:  
    반복할 구문

## 1부터 100까지의 합계를 구하는 코드

```
sum = 0
for x in range(1, 100+1):
    sum += x
print(sum)
```

```
sum = 0
count = 1
while count <= 100:
    sum += count
    count = count + 1
print(sum)
```

# while 응용

```
import random
random_number = random.randint(1, 30) # 1 ~ 30 중 무작위 하나 선택

while True:
    user_input = int(input("맞춰보세요"))

    if user_input == random_number :
        print("정답")
        break
    if user_input < random_number :
        print("up")
    if user_input > random_number :
        print("down")
```

# break, continue

break : 반복문 안에서 강제로 빠져나옵니다

continue : 해당 루프를 건너뛰고, 다음 반복을 계속해서 진행합니다.

```
for i in range(10):  
    if i == 5:  
        break  
    print(i)  
print("end")
```

```
for i in range(10):  
    if i == 5:  
        continue  
    print(i)
```



# 구구단

<https://www.acmicpc.net/problem/2739>

N을 입력받은 뒤, 구구단 N단을 출력하는 프로그램을 작성하시오.  
출력 형식에 맞춰서 출력하면 된다.

입력	출력
2	2*1=2 2*2=4 2*3=6 2*4=8 2*5=10 2*6=12 2*7=14 2*8=16 2*9=18

# 별 찍기

<https://www.acmicpc.net/problem/2444>

예제를 보고 규칙을 유추한 뒤에 별을 찍어 보세요.

입력	출력
5	<pre>  *  *** ***** ***** ***** *****   *  ***   *</pre>