



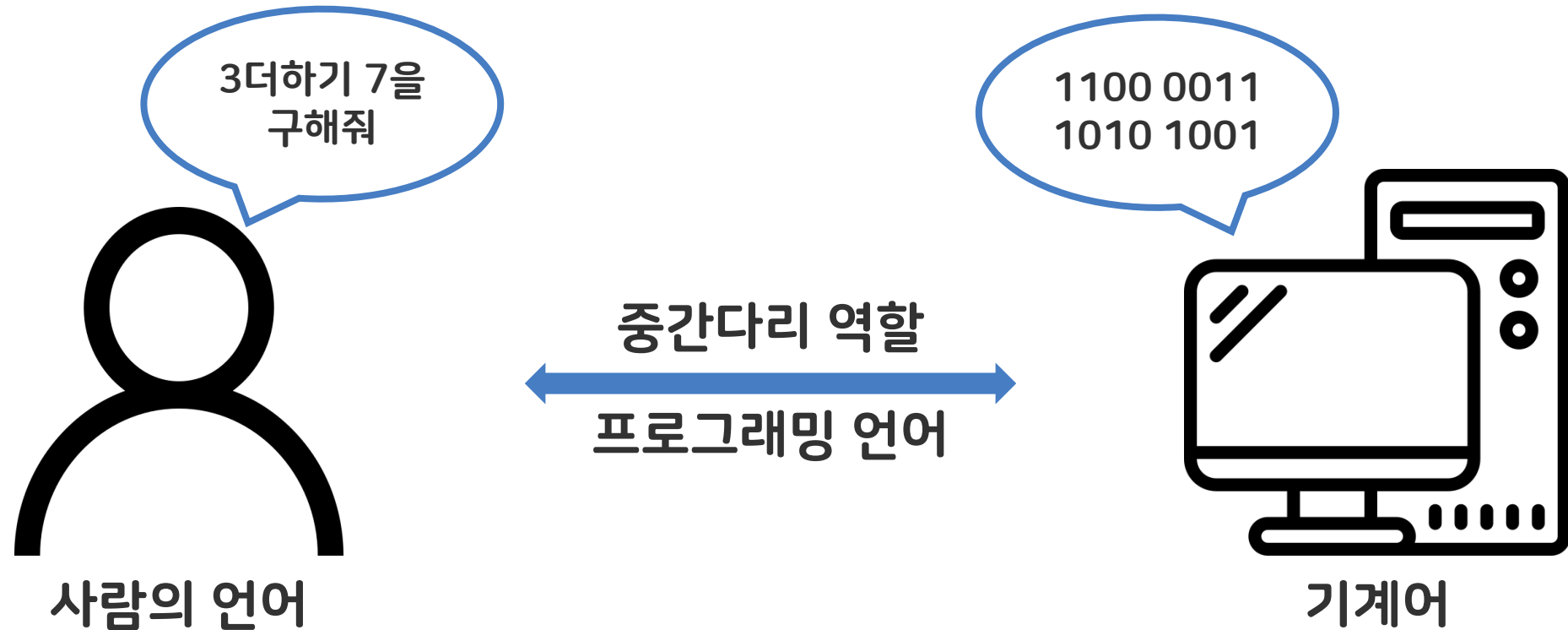
스마트인재개발원
Smart Human Resources Development

김 은 영 강사



학습목표

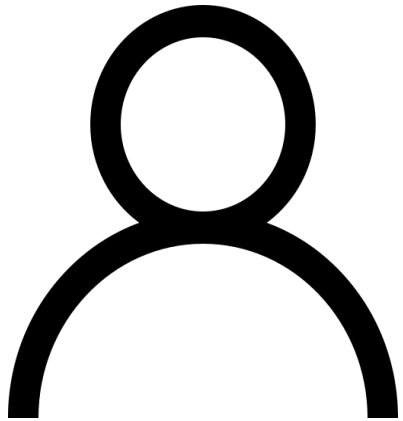
- Python의 개념, 특징을 이해한다.
- Python 개발환경구축을 할 수 있다.
- 변수 및 자료형에 대해 알 수 있다.
- 문자열 자료형을 이해하고 활용 할 수 있다.



컴퓨터를 이용하여 특정 문제를 해결하기 위한
프로그램을 작성하기 위해 사용되는 언어

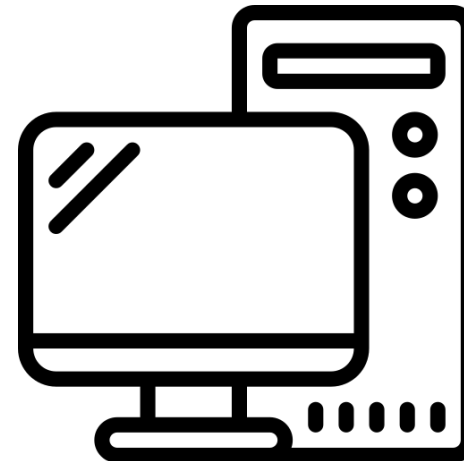
고급 언어











C, Java, Python 등



저급 언어

어셈블리어



Nov 2022	Nov 2021	Change	Programming Language		Ratings	Change
1	1			Python	17.18%	+5.41%
2	2			C	15.08%	+4.35%
3	3			Java	11.98%	+1.26%
4	4			C++	10.75%	+2.46%
5	5			C#	4.25%	-1.81%
6	6			Visual Basic	4.11%	-1.61%
7	7			JavaScript	2.74%	+0.08%
8	8			Assembly language	2.18%	-0.34%
9	9			SQL	1.82%	-0.30%
10	10			PHP	1.69%	-0.12%

[TIOBE Index for November 2022]



- 1990년 네덜란드 프로그래머 **guido 반 로섬**
(Guido Van Rossum)
- 어원 : "Monty **Python**'s Flying Circus"
코미디 프로그램
- **인터프리터** 방식의 언어

장점

- 인간다운 언어

```
if 4 in [1, 2, 3, 4]:  
    print('4가 있습니다.')
```

- 문법이 쉽고 코드가 간결

```
for(int i=0 ; i<10 ; i++) {  
    for(int j=0 ; j<=i ; j++) {  
        System.out.print("*");  
    }  
    System.out.println();  
}
```


Java

```
for i in range(10):  
    print('*' * (i+1))
```

Python

장점

- 높은 확장성 및 이식성
- 거대한 생태계

 파이썬 사용...	
웹 개발 :	Django , Pyramid , Bottle , Tornado , Flask , web2py
GUI 개발 :	tkInter , PyGObject , PyQt , PySide , Kivy , wxPython
과학 및 숫자 :	SciPy , Pandas , IPython
소프트웨어 개발 :	Buildbot , Trac , Roundup
시스템 관리 :	Ansible , Salt , OpenStack , xonsh

단점

- 컴파일 언어에 비해 느림

Google



NETFLIX





Python
패키지
(3.x)

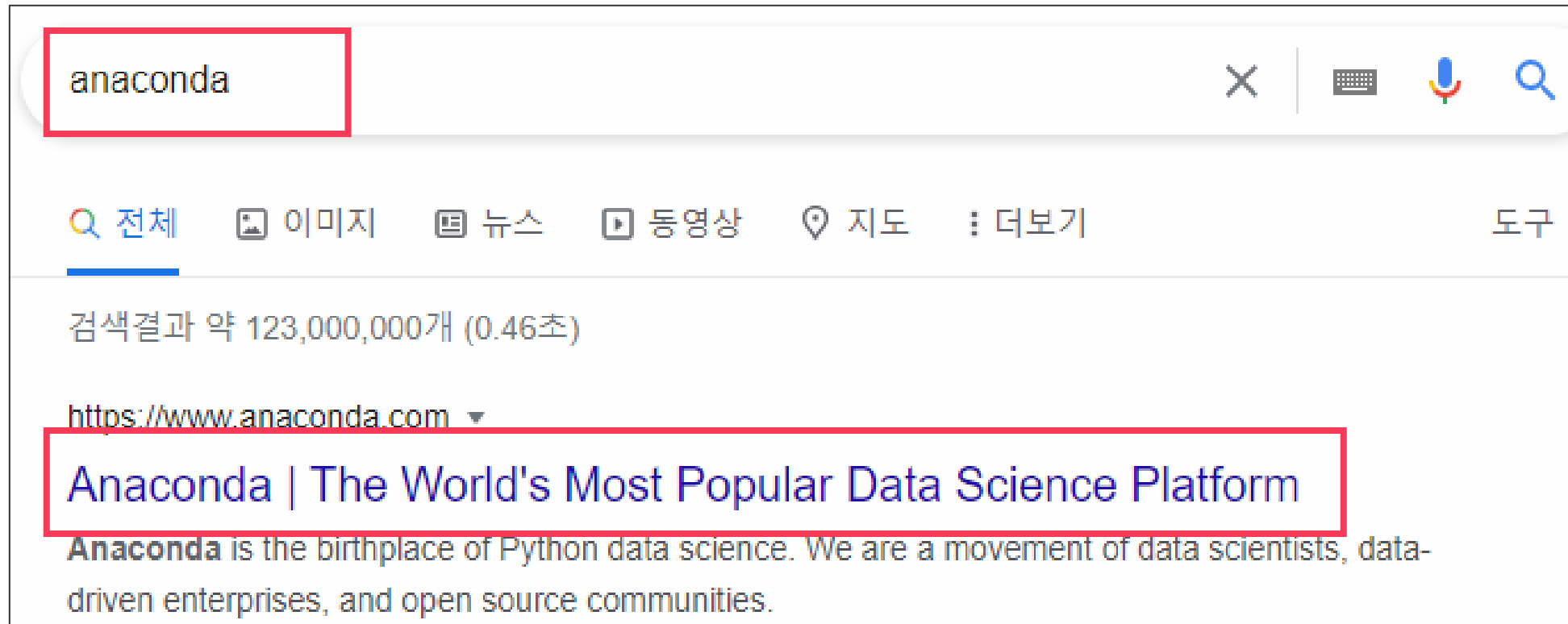


에디터

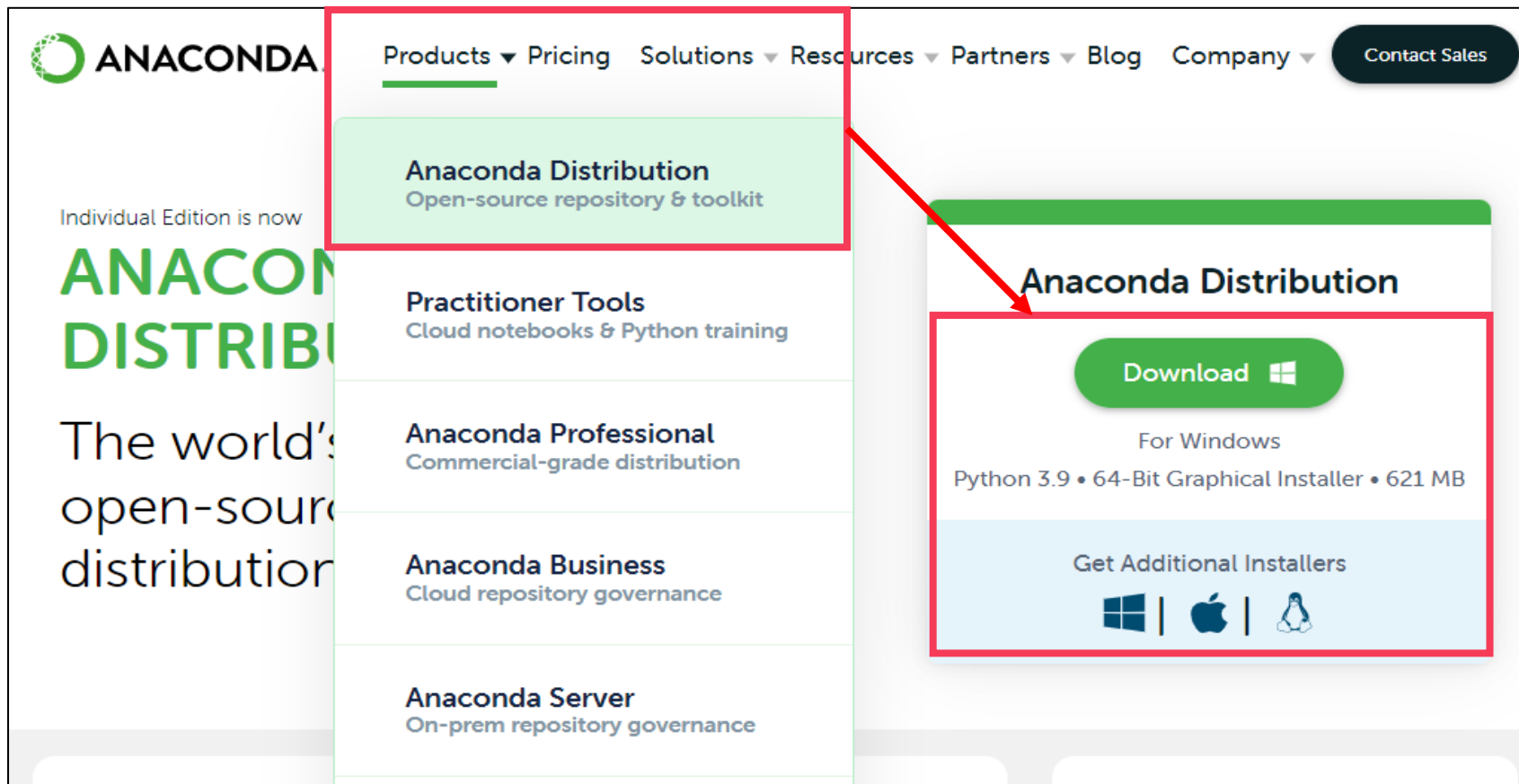


Python을 활용한
데이터 분석, 어플리케이션
개발에 도움을 주는 플랫폼

google에서 **anaconda**검색



좌측 상단 Products > Anaconda Distribution 클릭



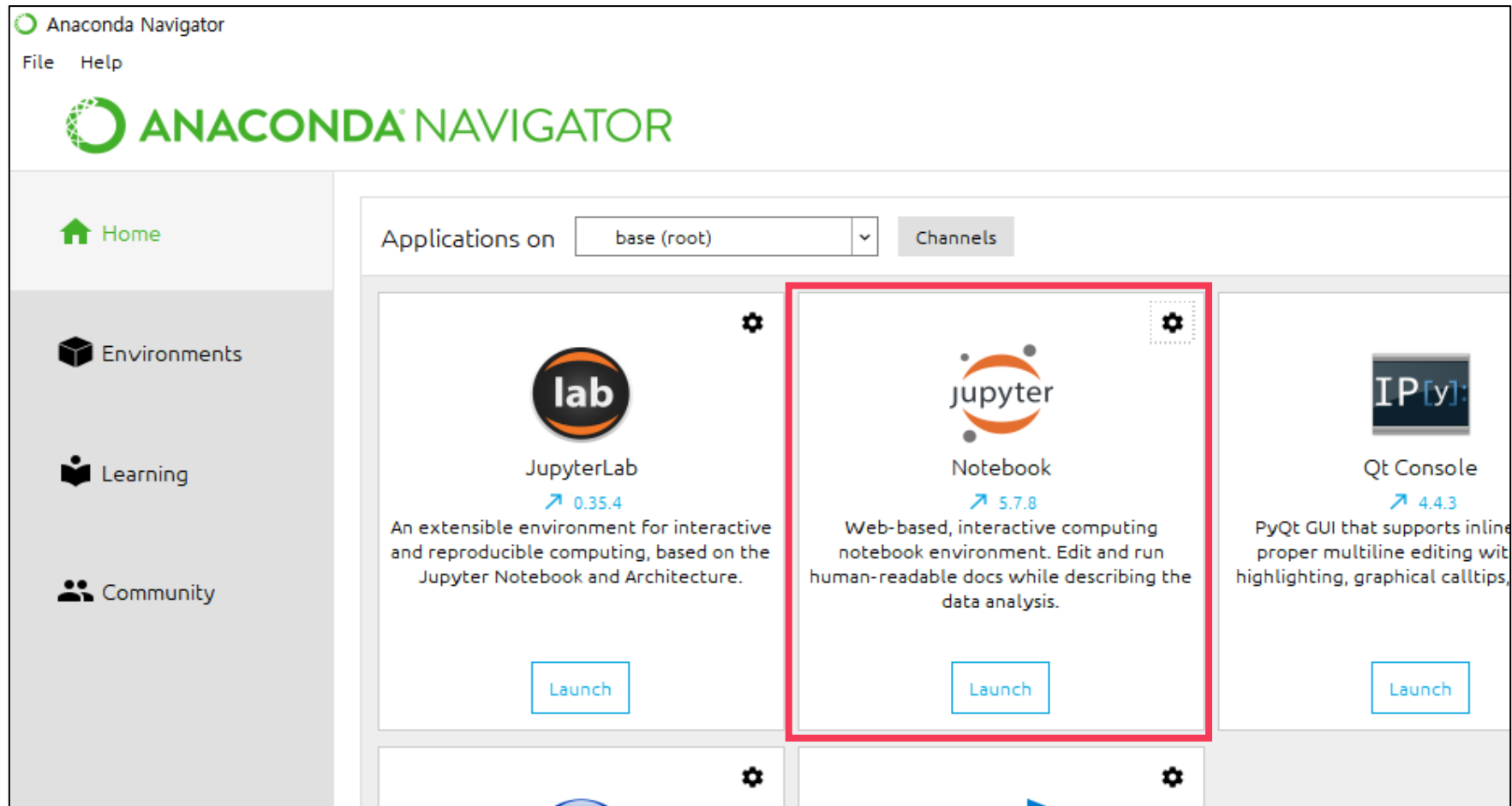
Windows -> 3.9 version 64Bit 클릭

Windows 

Python 3.9

64-Bit Graphical Installer (621 MB)

anaconda 실행 후 Jupyter notebook Launch 클릭



Jupyter notebook 확장기능(Nbextensions package)

- 실행시간, 목차, 변수 탐색기, 자동 코드정리 등의 기능 제공

1. 아나콘다 프롬프트에서 아래 코드 입력

```
pip install jupyter_nbextensions_configurator jupyter_contrib_nbextensions
```

or

```
(pip install jupyter_contrib_nbextensions && jupyter contrib nbextension install)
```

2. Nbextensions를 Jupyter notebook과 연결

```
jupyter contrib nbextension install --user
```

```
jupyter nbextensions_configurator enable --user
```

3. Jupyter notebook Extension 탭 클릭 후 기능 선택

Code prettify

자동 코드정리

Execute Time

실행시간 확인

Table of Contents(2)

목차 생성(네비게이터)

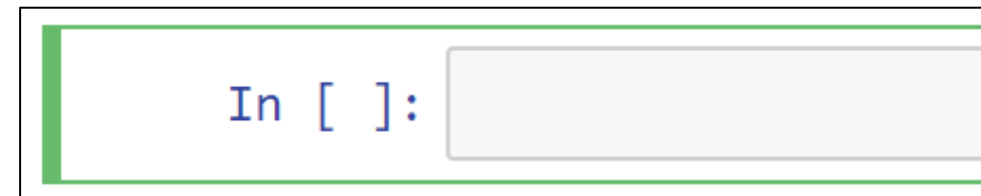
Variable Inspector

변수 탐색기

Command Mode



Edit Mode



- 프로그래밍에 있어 내용을 메모하는 목적으로 사용
- 소스 코드를 더 쉽게 이해 할 수 있도록 설명이나 참고사항을 입력
- 컴파일러와 인터프리터에 의해 일반적으로 무시되어 프로그램 실행에 영향을 주지 않음.
- 파이썬의 주석 표시 : **#**

```
1 + 3 # 1 더하기 3 실행
```

```
4
```

Command Mode



- Enter : Edit Mode로 전환
- a : 위에 셀(Cell) 추가
- b : 아래에 셀(Cell) 추가
- m : Markdown으로 전환
- dd : 셀(cell) 삭제
- y : 셀(cell) 상태를 뒤로 되돌리기

Edit Mode



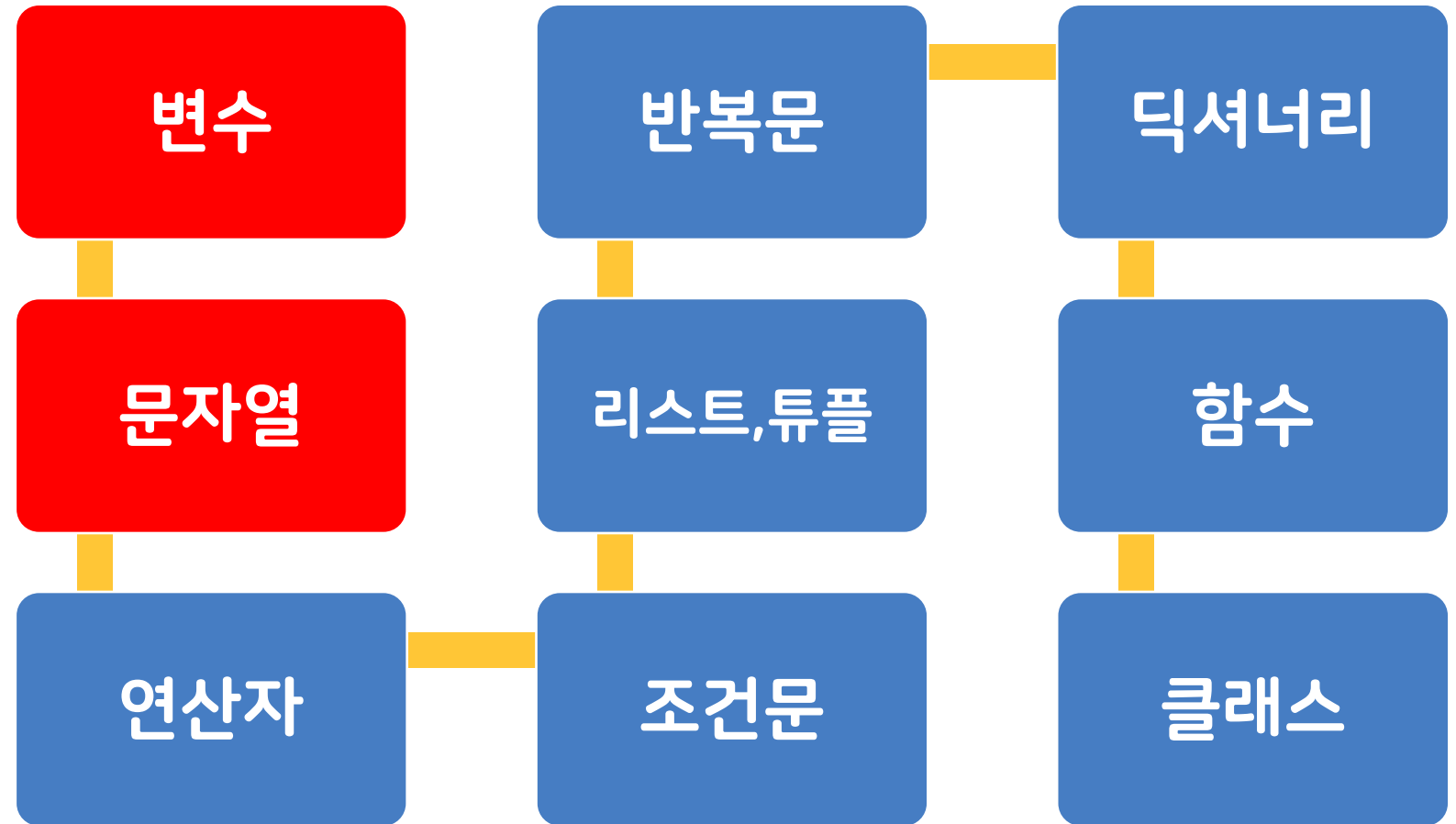
- Esc : Command Mode로 전환
- Ctrl + z : 앞으로 되돌리기
- Ctrl + y : 뒤로 되돌리기
- Ctrl + d : 커서가 위치한 줄 삭제

공통 단축키

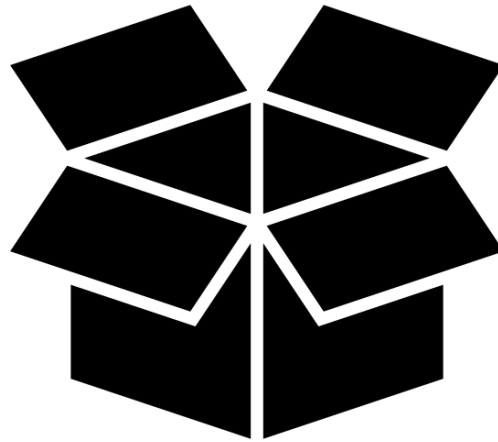
- Ctrl + Enter : 셀(cell) 실행
- Shift + Enter / Alt + Enter : 셀(cell) 실행 후 아래로 셀(cell) 추가 후
커서 이동



수업 진행방향



- 사전적 : “변화를 줄 수 있는 ” 또는 “변할 수 있는 수”
- 프로그래밍 : 데이터를 담을 수 있는 공간



num = 3

↓ ↓ ↓

변수명 대입 값



```
num = 3  
print(num)  
print(type(num))
```

3
<class 'int'>

동적 자료형

1. 영문자, 숫자, 언더바(_) 사용

단, 영문자는 대문자와 소문자를 다르게 인식

```
number = 10
Number = 20
print("number :", number)
print("Number :", Number)
```

```
number : 10
Number : 20
```

```
num1_num2 = 10
num1_num2
```

```
10
```

2. 숫자로 시작 불가능

```
1a = 10
```

```
File "<ipython-input-3-1be9f8edb7cd>", line 1
```

```
1a = 10
```

```
^
```

```
SyntaxError: invalid syntax
```


3. 키워드(예약어) 사용 불가능 (내장함수명 사용을 권장하지 않음)

```
if = 30  
if
```

```
File "<ipython-input-5-23187ed1a6a1>", line 1
```

```
    if = 30
```

```
        ^
```

```
SyntaxError: invalid syntax
```

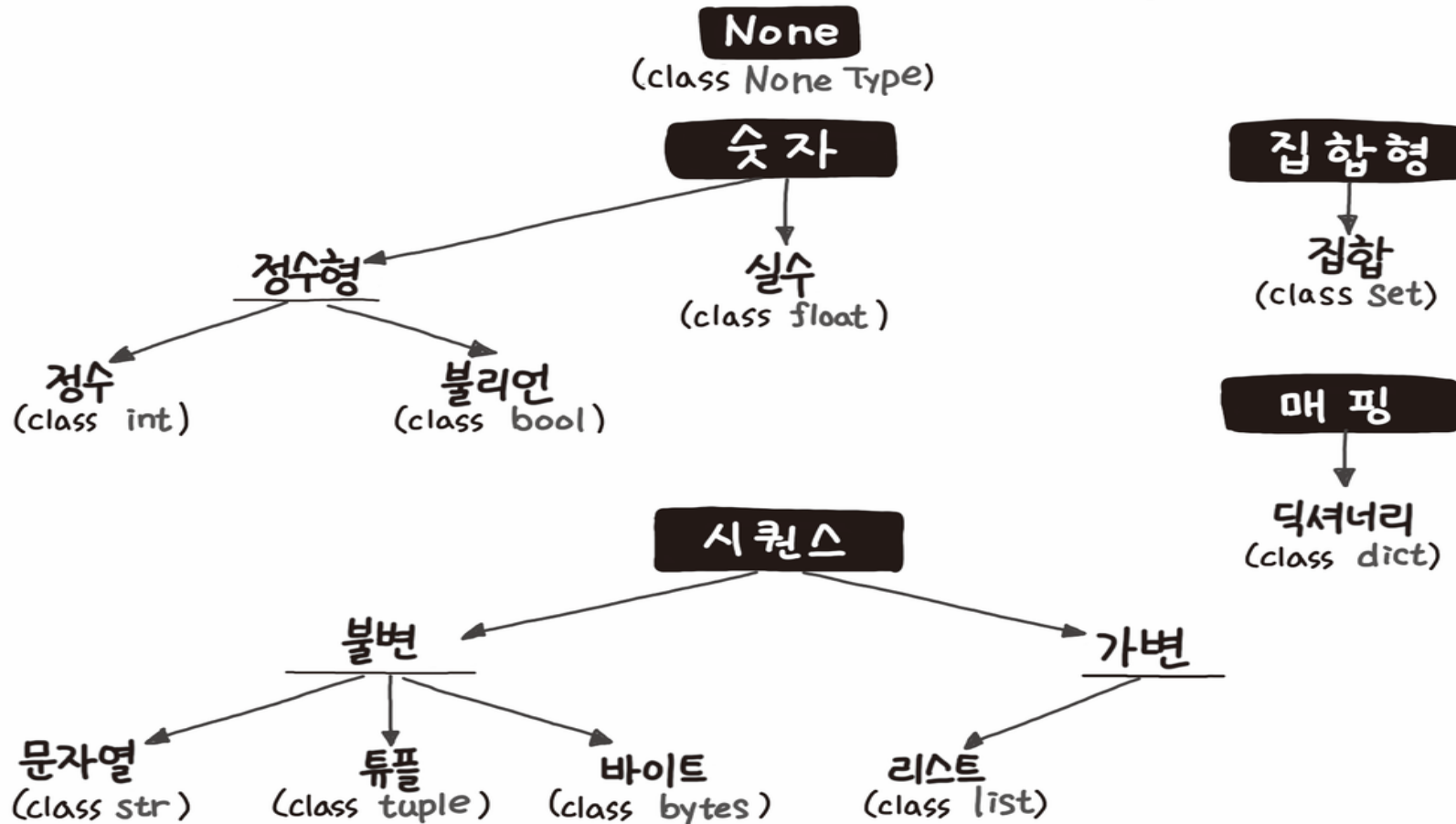
키워드(예약어) 확인

```
import keyword  
print(keyword.kwlist)
```

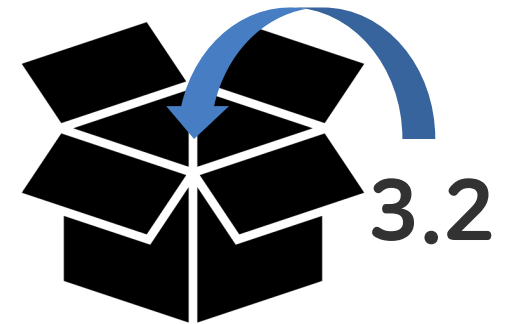
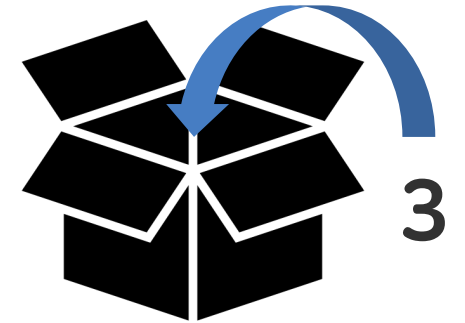
권장사항

- 변수명의 첫 글자는 항상 소문자로 지정
- 두 가지의 문자를 섞어서 변수명을 만들 경우 두 단어를 구분
ex) 카멜(Camel) 표기법 : `numberList`
스테이크(Snake) 표기법 : `number_list`
파스칼(Pascal) 표기법 : `NumberList`

파이썬3 표준 타입 계층 구조



자료형	설명
숫자형 int/float	-2, -1, 0, 1, 2, 3.2, 3.14, 0.12
문자열 str	'Hello World!', "Hi", "123"
리스트 list	[], [1, 2, 3], ['a', 'b', 'c'], ['a', 1, 0.5]
튜플 tuple	(), (1, 2, 3), ('a', 'b', 'c'), ('a', 1, 0.5)
딕셔너리 dictionary	{'name':'ey', 'lect':'python'}
집합 set	{1, 2, 3}
불리언 boolean	True, False



변수에 숫자(정수) 대입

- num1 변수에 숫자 13을 대입하시오.
- num2 변수에 숫자 25를 대입하시오.
- num2 변수에 숫자 77을 대입하시오.

변수에 숫자(실수) 대입

- num3 변수에 숫자 3.1425을 대입하시오.
- num4 변수에 숫자 1.25를 대입하시오.
- num5 변수에 숫자 3.12을 대입하시오.

변수에 문자열 대입

- str1 변수에 문자열 "Funny python"을 대입하시오.

```
str1 = 'Funny python'  
str1  
'Funny python'
```

- str2 변수에 숫자 'Easy python'를 대입하시오.

```
str2 = 'Easy python'  
str2  
'Easy python'
```

- 변수 a에 10을 대입, 변수 b에 15를 대입하시오.

```
a = 10  
b = 15  
print(a)  
print(b)
```

10
15

```
a, b = 10, 15  
print(a)  
print(b)
```

10
15

변수 str1과 str2에 “python” 문자열을 대입하시오.

```
str1 = 'python'  
str2 = 'python'  
print(str1)  
print(str2)
```

python
python

```
str1 = str2 = 'python'  
print(str1)  
print(str2)
```

python
python

변수 **x**에는 100을 대입, 변수 **y**에는 200을 대입 후 변수 **sum**에는 두 변수의 합을 대입하여 아래와 같이 출력하시오.

```
x = ?  
y = ?  
sum = ?  
print(sum)
```

300

👀 ✨ 여기서 발견할 수 있는 오류는?

- Python에서 자료형을 확인하는 방법

1) type(대상)

2) isinstance(대상, 자료형)

```
x = 10
print(type(x))
print(isinstance(x, int))
print(isinstance(x, float))
print(isinstance(x, str))
```

- 문자열 안에 작은 따옴표나 큰 따옴표를 포함시키고 싶을 때

She's gone

```
s = 'she's gone'  
s
```

```
Input In [19]  
s = 'she's gone'  
      ^
```

```
SyntaxError: invalid syntax
```

- 문자열 안에 작은 따옴표나 큰 따옴표를 포함시키고 싶을 때

She's gone

```
s = 'she\'s gone'  
print(s)
```

she's gone

이스케이프 코드

- 프로그래밍할 때 사용할 수 있도록 미리 정의해 둔 '문자 조합'

코드	설명
\n	개행(줄바꿈)
\t	수평 탭
\\	문자 "\"
'\''	단일 인용부호(')
'\"'	이중 인용부호(")

문자열 안에 작은 따옴표나 큰 따옴표를 포함시키고 싶을 때

She's gone

```
s1 = "she's gone"  
s2 = "she\'s gone"  
print(s1)  
print(s2)
```

```
she's gone  
she's gone
```

문자열 안에 작은 따옴표나 큰 따옴표를 포함시키고 싶을 때

he said that "She is gone"

```
s1 = 'he said that "she is gone"'
s2 = "he said that \"she is gone\""
print(s1)
print(s2)
```

```
he said that "she is gone"
he said that "she is gone"
```


여러 줄인 문자열을 변수에 대입하고 싶을 때

- 이스케이프 코드 활용

```
s = ?  
print(s)
```

파이썬

- 동적언어
- 직관적, 명시적, 간결함

여러 줄인 문자열을 변수에 대입하고 싶을 때

- 따옴표 3개를 연속으로 써서 양쪽 둘러 싸기

```
s = ?  
print(s)
```

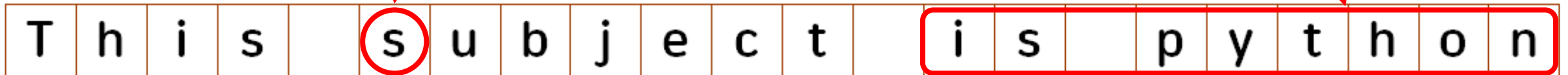
파이썬

- 동적언어
- 직관적, 명시적, 간결함

인덱싱(indexing)

- 무엇인가를 '가리킨다'는 의미

T h i s s u b j e c t i s p y t h o n



슬라이싱(slicing)

- 무엇인가를 '잘라낸다'는 의미

인덱스 번호(위치) 기반 접근

```
s = 'This subject is python'
```

T	h	i	s		s	u	b	j	e	c	t		i	s		p	y	t	h	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
-22	-21	-20	-19	-18	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

문자 인덱싱(indexing)

T	h	i	s		s	u	b	j	e	c	t		i	s		p	y	t	h	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
-22	-21	-20	-19	-18	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

```
s = 'This subject is python'  
print(s[6])  
print(s[16])
```

```
s = 'This subject is python'  
print(s[-2])  
print(s[-1])
```

문자열 슬라이싱(slicing) - 'This', 'subject', 'python' 문자열 가져오기

T	h	i	s		s	u	b	j	e	c	t		i	s		p	y	t	h	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

```
s = 'This subject is python'
print(s[0:4])
print(      )
print(      )
```

```
This
subject
python
```

문자열 슬라이싱(slicing) - 'is python' 문자열 가져오기

T	h	i	s		s	u	b	j	e	c	t		i	s		p	y	t	h	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
-22	-21	-20	-19	-18	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

```
s = 'This subject is python'
print(s[13:22])
print(s      )
print(s      )
```

```
is python
is python
is python
```

다음과 같은 문자열에서 날짜와 날씨를 출력하시오.

```
day = '2022년 11월 25일의 날씨는 햇볕 쨍쨍입니다.'
```

날짜 : 2022년 11월 25일

날씨 : 햇볕 쨍쨍

함수	설명
len()	문자열의 문자 개수 반환
upper()	소문자를 대문자로 바꾸기
lower()	대문자를 소문자로 바꾸기
isupper()	문자열이 대문자인지 여부 반환
islower()	문자열이 소문자인지 여부 반환
isdigit()	문자열이 숫자인지 여부 반환
count('찾을 문자열')	'찾을 문자열'이 몇 개 있는지 개수 반환
find('찾을 문자열')	'찾을 문자열' 위치 반환
index('찾을 문자열')	'찾을 문자열' 위치 반환
strip(), lstrip(), rstrip()	양쪽 공백 제거, 왼쪽, 오른쪽 공백 제거
replace('문자1', '문자2')	문자열1을 문자열2로 바꾸기
split()	문자열 나누기

문자열 포매팅(formatting)

- 문자열 안의 특정한 값을 바꾸거나 추가할 경우가 있을 때 사용

```
s = '오늘은 11월 28일 입니다.'
```

```
s = '오늘은 11월 29일 입니다.'
```

- %기호 포매팅
- format 함수 포매팅
- f문자열 포매팅

“%d” 를 이용해서 정수 대입

```
day = 28  
s = "오늘은 11월 %d일 입니다."%day  
s
```

'오늘은 11월 28일 입니다.'

```
day = 29  
s = "오늘은 11월 %d일 입니다."%day  
s
```

'오늘은 11월 29일 입니다.'

2개 이상 값을 포매팅할 때

```
month = 11
day = 28
s = '오늘은 %d월 %d일 입니다.'%(month, day)
s

'오늘은 11월 28일 입니다.'
```

문자열 포맷 코드

- 문자열 내 값 삽입

함수	설명
%s	문자열(string)
%c	문자 1개
%d	정수(integer)
%f	실수(float-point)
%%	Literal % (문자 '%' 자체)

format 함수를 사용한 포매팅

```
month = 11  
day = 28  
s = "오늘은 {}월 {}일 입니다.".format(month, day)  
print(s)
```

오늘은 11월 28일 입니다.

f 문자열을 사용한 포매팅

```
month = 11  
day = 28  
s = f"오늘은 {month}월 {day}일 입니다."  
print(s)
```

오늘은 11월 28일 입니다.

lang 변수에 'python'을 대입하고, 아래와 같이 출력하시오.

?

```
print(s)
```

```
Life is too short, You need 'python'
```


변수 **x**에는 100을 대입, 변수 **y**에는 200을 대입 후 변수 **sum2**에는 두 변수의 합을 대입하고 포매팅을 이용하여 아래와 같이 출력하시오.

```
x = 100
y = 200
sum2 = x + y
print(
```

100와 200의 합은 300입니다.

실행시점의 현재시간(시, 분, 초)을 출력하는 출력문을 작성하시오.
단, 아래 코드를 참고하여 현재 시간을 구하시오.

```
from datetime import datetime
current_time = datetime.now()
current_time.hour
current_time.minute
current_time.second
```

```
from datetime import datetime
current_time = datetime.now()
print(current_time)
print( )
```

2022-11-17 16:25:56.742228

실행 당시 현재 시간은 16시 25분 56초 입니다.

review = "[수료생 Real 후기] 후기입력 2022.10.13. 작성자 킹수. 드디어 W'인공지능 융합서비스 개발자과정W'을 수료했습니다!! 6개월이라는 길다면 길고 짧다면 짧은 기간 동안 많은 노력을 통해 받아낸 수료인 만큼 마음이 벅차네요. 그치만 정말 재밌고 즐거운 6개월이었습니다. 많은 것을 배웠고, 또 자신감도 얻은 기간이었다고 생각합니다. 이 후기를 본 많은 분들이 저를 따라서 수료를 목표로 도전하셨으면 좋겠습니다!!! 사랑해요 여러분"

[기사출처] 스마트인재개발원

```
start =  
end =  
day =  
print(f'후기가 업로드된 일자 는 {day}입니다.')
```

후기가 업로드된 일자 는 2022.10.13입니다.

<풀이 접근 방식>

1. 문자열 함수 활용 특정 문자열 시작 인덱스 번호 확인
2. 인덱스번호 이용 특정 문자열 슬라이싱
3. 슬라이싱 된 문자열을 포매팅으로 문장 연결하기



다음시간에는?

연산자