



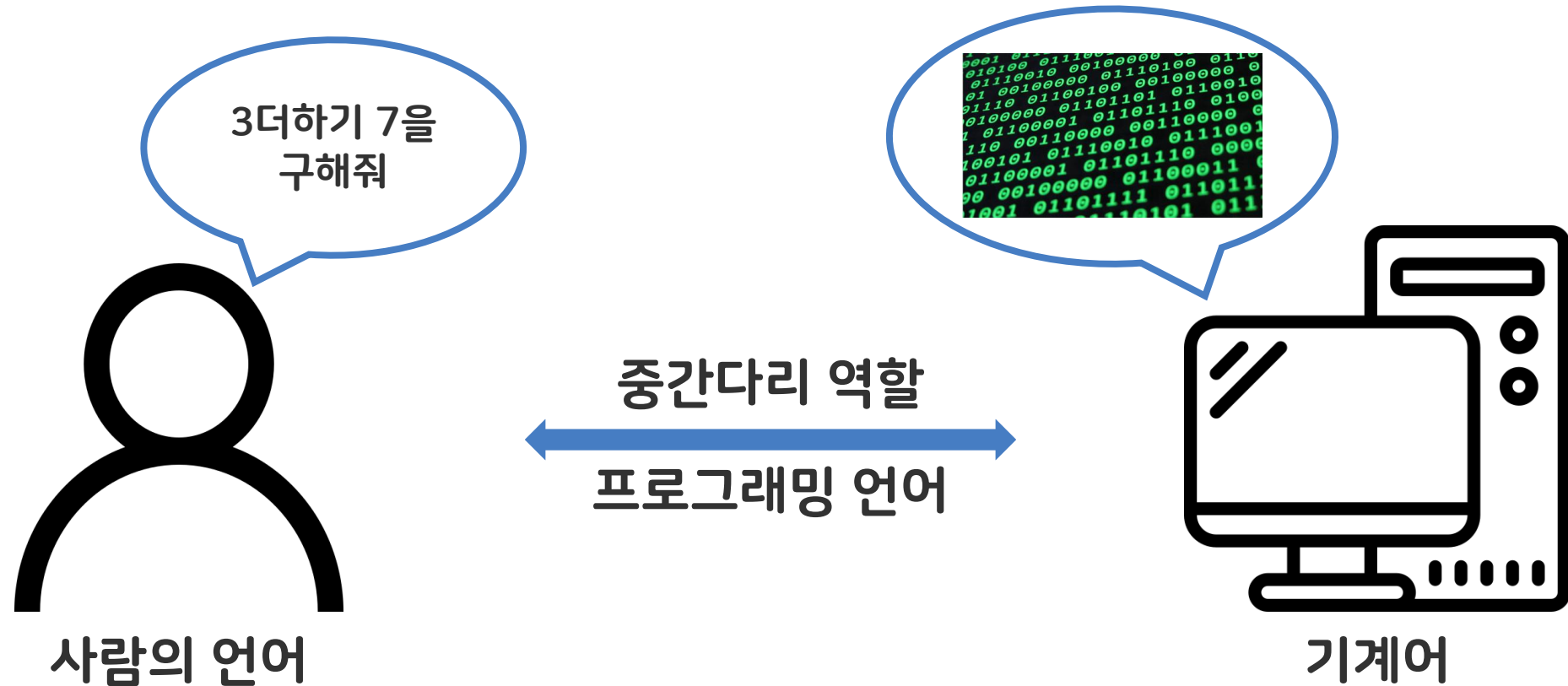
스마트인재개발원
Smart Human Resources Development

최성우 연구원



학습목표

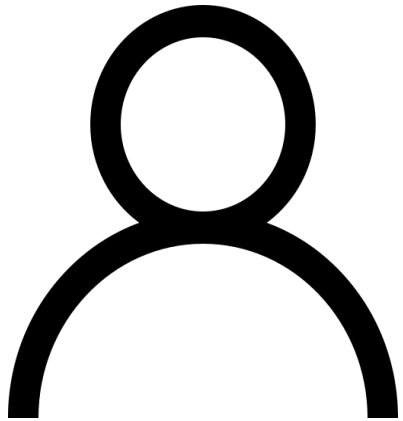
- Python의 개념을 알 수 있다.
- Python의 특징을 이해한다.
- Python 개발환경구축을 할 수 있다.
- 변수 및 자료형에 대해 알 수 있다.
- 문자열 자료형을 이해하고 활용 할 수 있다.



- 컴퓨터를 이용하여 특정 문제를 해결하기 위한 프로그램을 작성하기 위해 사용되는 언어

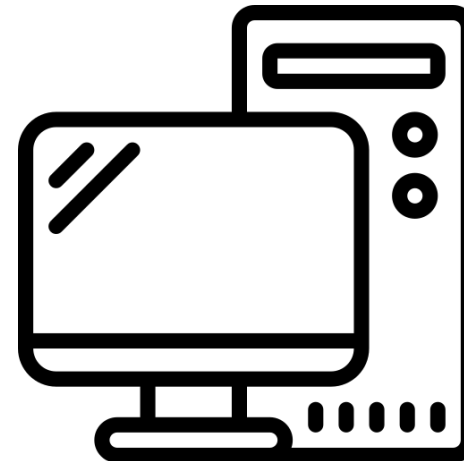
고급 언어

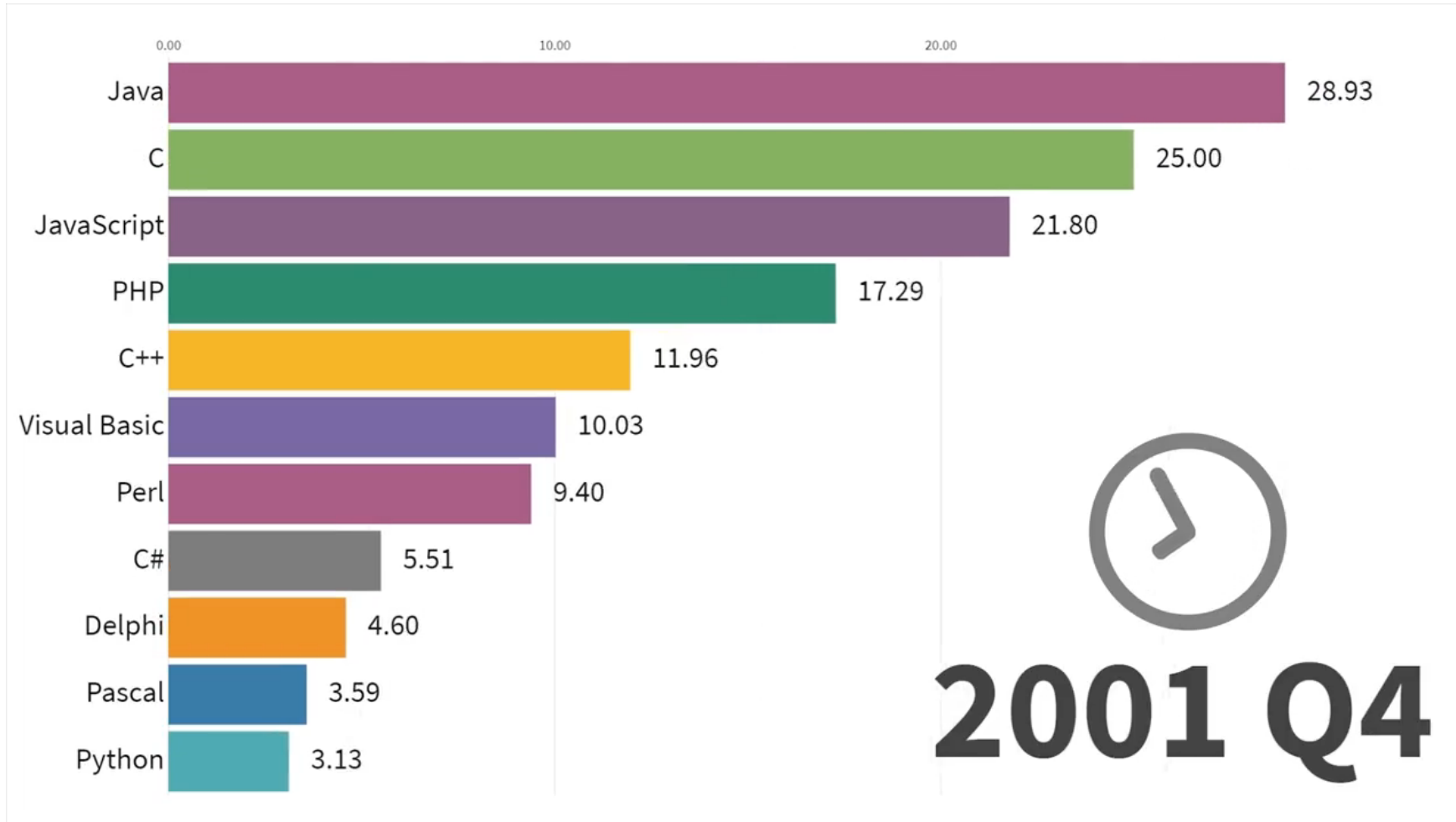
C, Java, Python 등



저급 언어

어셈블리어







- Python은 1990년 네덜란드의 **귀도 반 로섬** (Guido Van Rossum)이 개발한 **인터프리터 언어**이다.
- **인터프리터 언어**란 프로그래밍 언어의 소스 코드를 한 줄씩 실행 가능한 언어를 뜻한다.
- Java는 컴파일 언어로 한 번에 전체 코드를 실행해야 한다.
- 문법이 쉬워 프로그래밍 언어 **입문용**으로 많이 사용된다.

python Python이란?

```
1 package Java_2주차;
2
3 public class day11_2 {
4
5     public static void main(String[] args) {
6
7         int array[][] = new int[7][7];
8         int num = 1;
9
10        for (int i = 0; i <= 3; i++) {
11            for (int j = 3-i; j <= i+3; j++) {
12                array[i][j] = num++;
13            }
14        }
15        for (int i = 4; i < array.length; i++) {
16            for (int j = i-3; j <= 9-i; j++) {
17                array[i][j] = num++;
18            }
19        }
20
21        for (int i = 0; i < array.length; i++) {
22            for (int j = 0; j < array.length; j++) {
23                System.out.print(array[i][j] + "\t");
24            }
25            System.out.println();
26        }
27    }
28 }
```

Markers Properties Servers Data Source Explorer Snippets Console

<terminated> day11_2 [Java Application] C:\Program Files\Java\jdk1.8.0_251\bin\javaw.exe (2020. 7. 17 오전 12:13:47 - 오전 1

0	0	0	1	0	0	0
0	0	2	3	4	0	0
0	5	6	7	8	9	0
10	11	12	13	14	15	16
0	17	18	19	20	21	0
0	0	22	23	24	0	0
0	0	0	25	0	0	0

1 num = 1

1 num

1

1 temp = "안녕"

1 temp

'안녕'

1 temp = temp + "여러분"

1 temp

'안녕여러분'

Google



Instagram



Dropbox



- 초보자가 배우기 쉽다.
- 다양한 라이브러리 및 툴과 연결이 가능하다.
- 머신러닝(ML), 딥러닝(DL), 인공지능(AI) 및 데이터 과학 라이브러리와 프레임워크를 활용하는데 최적화 되어 있다.
- Python의 점유율과 필요로 하는 기업의 수요가 증가하고 있다.

- 인간다운 언어이다.

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

만약 4가 1,2,3,4중에 있으면 "4가 있습니다." 를 출력한다.

- 문법이 쉬워 배우기 쉽고 코드가 간결하다.

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

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

- 시스템 소프트웨어 제작
- GUI 프로그래밍
- C/C++언어와의 결합
- 웹 프로그래밍
- 데이터베이스 프로그래밍
- 데이터 분석, 사물인터넷

- 시스템과 밀접한 프로그래밍 영역(ex 운영체제)
- 모바일(App)프로그래밍



Python
패키지
(3.x)

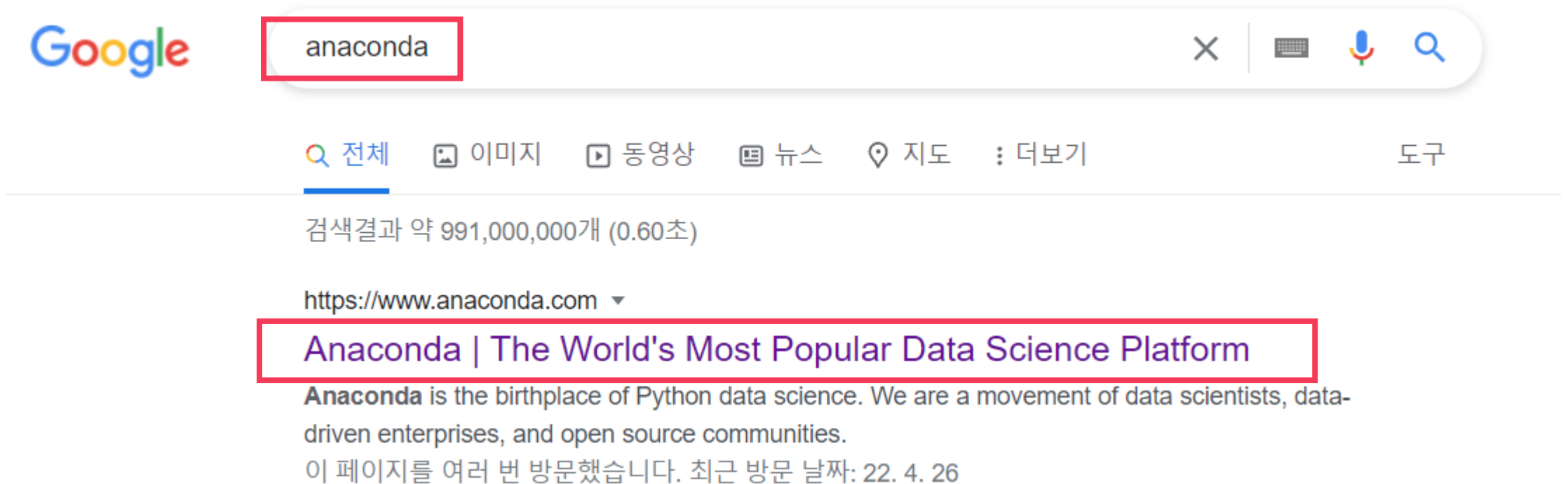


에디터



Python을 활용한
데이터 분석, 어플리케이션 개발에
도움을 주는 플랫폼

Google에서 anaconda 검색



The image shows a Google search interface. The search bar contains the word "anaconda". Below the search bar, there are tabs for "전체" (All), "이미지" (Images), "동영상" (Videos), "뉴스" (News), "지도" (Maps), and "더보기" (More). The "전체" tab is selected. Below the tabs, it says "검색결과 약 991,000,000개 (0.60초)". The first search result is from "https://www.anaconda.com" and is titled "Anaconda | The World's Most Popular Data Science Platform". The title is highlighted with a red box. Below the title, there is a description: "Anaconda is the birthplace of Python data science. We are a movement of data scientists, data-driven enterprises, and open source communities." and a note: "이 페이지를 여러 번 방문했습니다. 최근 방문 날짜: 22. 4. 26".

Google

anaconda

전체 이미지 동영상 뉴스 지도 더보기 도구

검색결과 약 991,000,000개 (0.60초)

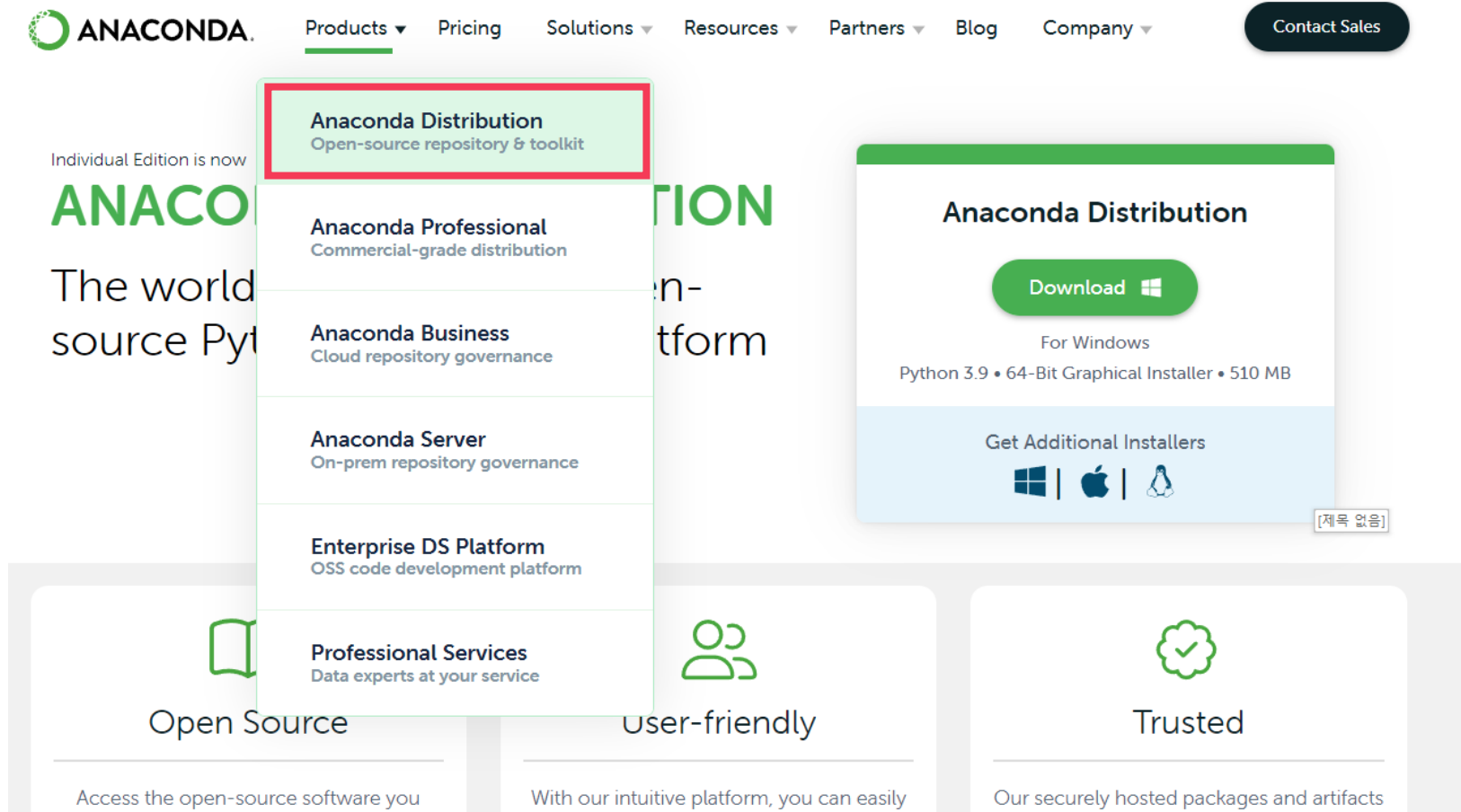
https://www.anaconda.com ▾

Anaconda | The World's Most Popular Data Science Platform

Anaconda is the birthplace of Python data science. We are a movement of data scientists, data-driven enterprises, and open source communities.

이 페이지를 여러 번 방문했습니다. 최근 방문 날짜: 22. 4. 26

좌측 상단 Products -> Anaconda Distribution 선택



The screenshot shows the Anaconda website's navigation bar with the 'Products' dropdown menu open. The 'Anaconda Distribution' option is highlighted with a red box. Below the dropdown, a detailed view of the 'Anaconda Distribution' product is shown, including a 'Download' button for Windows and information about the Python version, installer type, and size. The bottom of the page features three key benefits: Open Source, User-friendly, and Trusted.


ANACONDA Products Pricing Solutions Resources Partners Blog Company [Contact Sales](#)

Individual Edition is now **ANACONDA**

The world's most popular open-source Python platform




- Anaconda Distribution**
Open-source repository & toolkit
- Anaconda Professional**
Commercial-grade distribution
- Anaconda Business**
Cloud repository governance
- Anaconda Server**
On-prem repository governance
- Enterprise DS Platform**
OSS code development platform
- Professional Services**
Data experts at your service

Anaconda Distribution

[Download](#) 

For Windows
Python 3.9 • 64-Bit Graphical Installer • 510 MB

Get Additional Installers

[제목 없음]

Open Source
Access the open-source software you need

User-friendly
With our intuitive platform, you can easily learn, build, and share

Trusted
Our securely hosted packages and artifacts

Windows -> 3.9 version 64-Bit 클릭

Anaconda Installers

Windows 

Python 3.9

64-Bit Graphical Installer (510 MB)


32-Bit Graphical Installer (404 MB)

MacOS 

Python 3.9

64-Bit Graphical Installer (515 MB)

64-Bit Command Line Installer (508 MB)

Linux 

Python 3.9

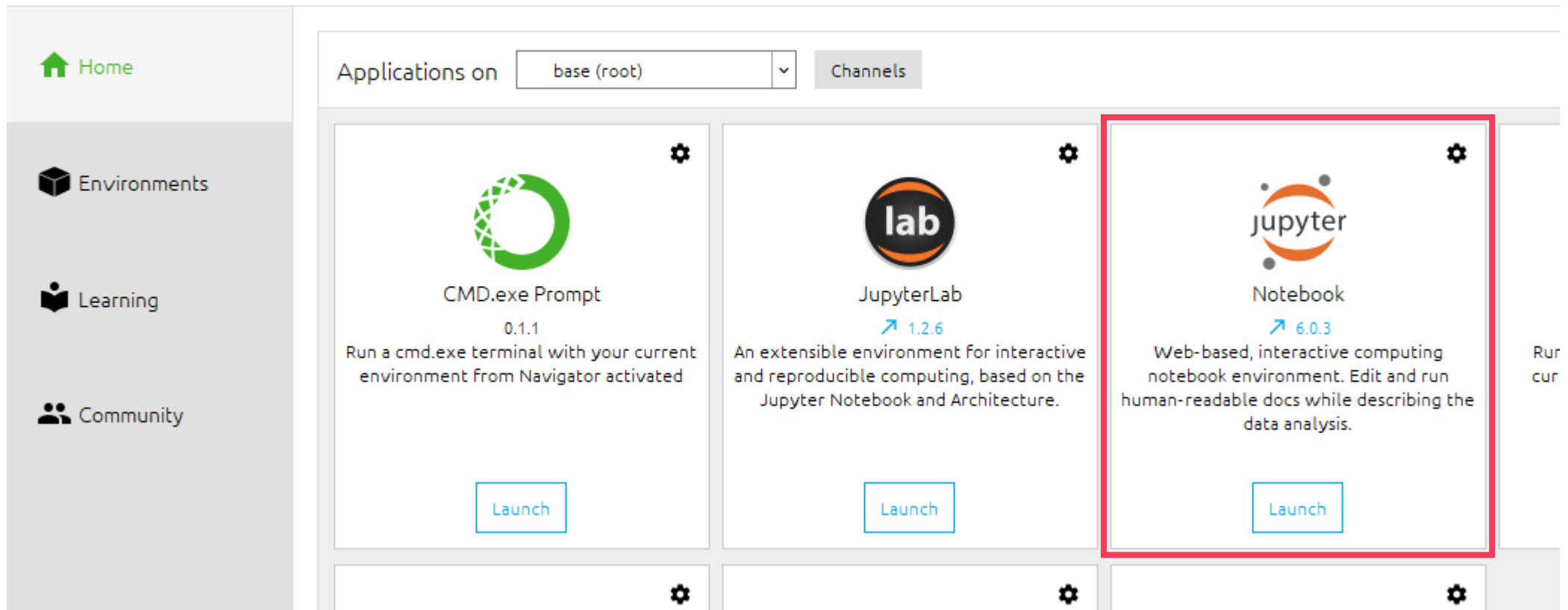
64-Bit (x86) Installer (581 MB)

64-Bit (Power8 and Power9) Installer (255 MB)

64-Bit (AWS Graviton2 / ARM64) Installer (488 MB)

64-bit (Linux on IBM Z & LinuxONE) Installer (242 MB)

ANACONDA NAVIGATOR 실행 후 Jupyter Notebook Launch(혹은 Install) 클릭



Command Mode



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

Edit Mode



- Esc : Command Mode로 전환
- Ctrl + z : 되돌리기
- Ctrl + y : 앞으로 되돌리기
- Ctrl + d : 코드 1줄 삭제

공통 단축키

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

주석

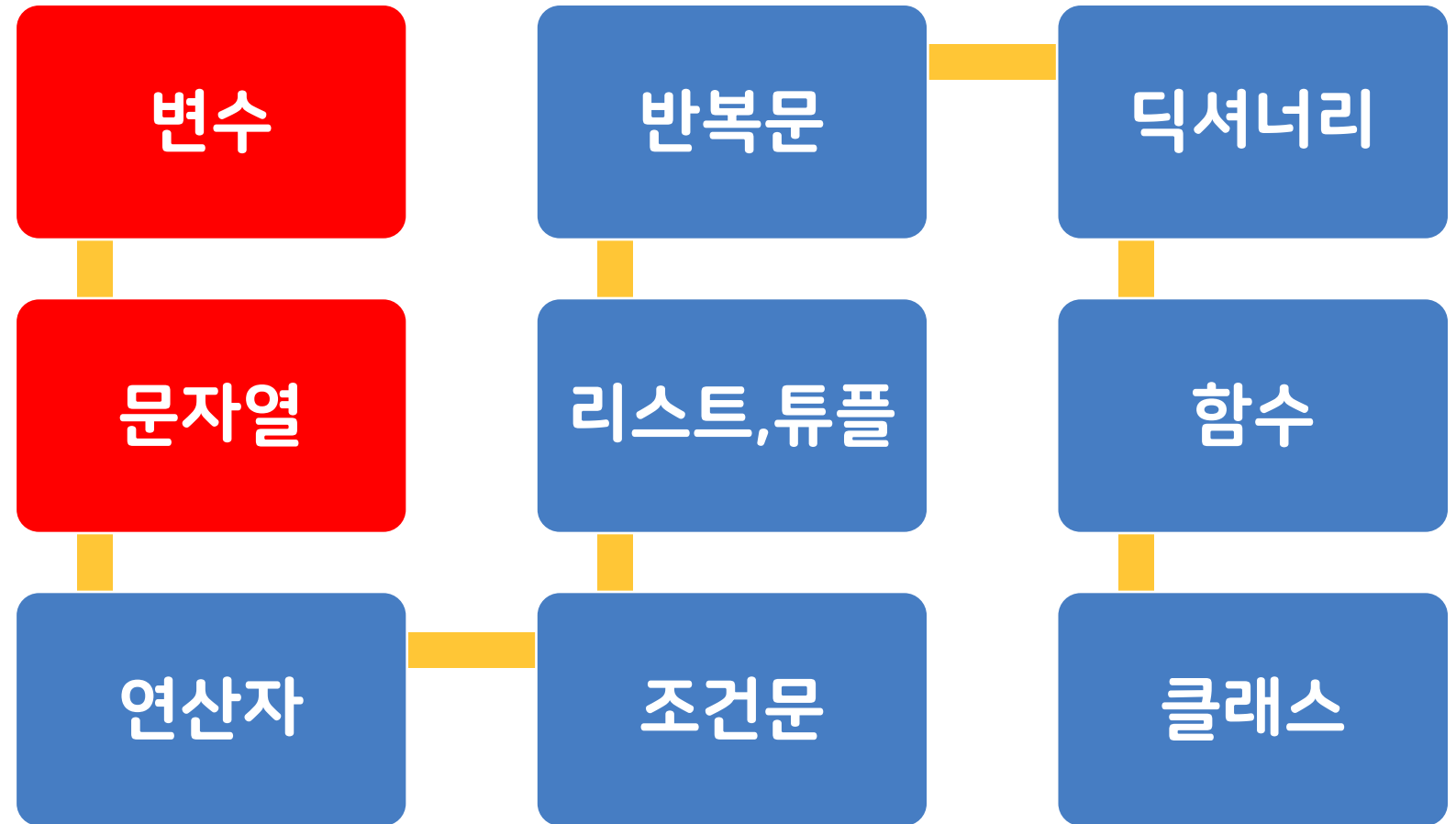
- 프로그래밍에 있어 내용을 **메모**하는 목적으로 사용
- 소스코드를 더 쉽게 이해할 수 있게 만드는 것이 주 목적
- 컴파일러와 인터프리터에 의해 일반적으로 무시되어 프로그램 실행에 영향을 주지 않음
- 파이썬은 "**#**"으로 주석을 사용

```
1+3 #1 더하기 3 실행
```

```
4
```

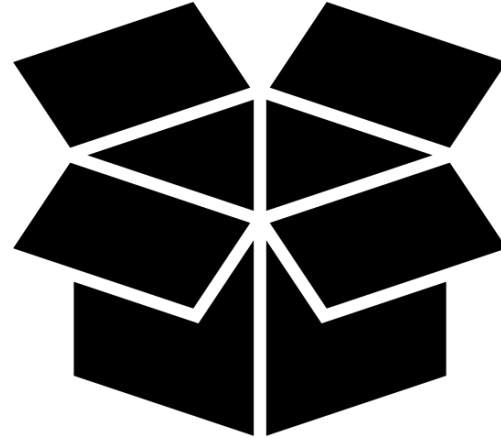


수업 진행방향



변수 (variable)

- 사전적 의미로는 “변할 수 있는 수” 라는 뜻 이지만
- 프로그래밍 언어에서는 데이터를 담을 수 있는 공간의 의미



num = 3

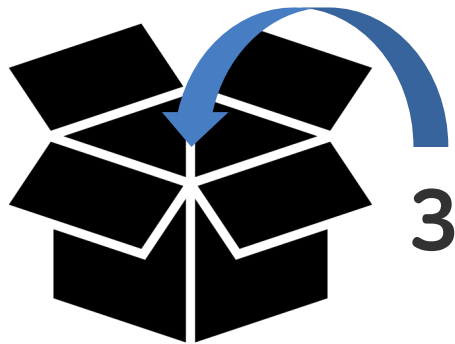
↓ ↓ ↓

변수명 대입 값



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

자료형	설명
숫자(정수)	-2, -1, 0, 1, 2
숫자(실수)	3.2, 3.14, 0.12
문자열	'Hello World!', "Hi", "123"
논리	True, False



변수에 숫자(정수) 대입

- num1 변수에 숫자 13을 대입하고 값을 확인해 보세요.
- num2 변수에 숫자 25를 대입하고 값을 확인해 보세요.
- num2 변수에 숫자 77을 대입하고 값을 확인해 보세요.
- 변수 num2의 자료형을 확인해 보세요.

1	num1 = 13
2	num1

13

변수에 숫자(실수) 대입

- num3 변수에 숫자 3.1425을 대입하고 값을 확인해 보세요.
- num4 변수에 숫자 1.25를 대입하고 값을 확인해 보세요.
- num4 변수에 숫자 3.12을 대입하고 값을 확인해 보세요.
- 변수 num4의 자료형을 확인해 보세요.

1	num3 = 3.1425
2	num3

3.1425

변수에 문자열 대입

- str1 변수에 문자열 "Funny python"을 대입하고 값을 확인해 보세요.
- str2 변수에 숫자 'Easy python' 을 대입하고 값을 확인해 보세요.
- 변수 str2의 자료형을 확인해 보세요.

```
str1 = "Funny python"  
str1
```

```
'Funny python'
```

1. 영문자, 숫자, 언더바(_)를 사용할 수 있다.
 - 단, 영문자는 대문자와 소문자를 다르게 인식한다.

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

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

```
num1_num2 = 10
num1_num2
```

```
10
```

※ print() : 변수 혹은 문자열 등의 값들을 출력(표시) 해주기 위한 명령(여러 결과를 한 번에 출력 가능)

2. 숫자로 시작 할 수 없다.

```
1a = 10
```

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

```
1a = 10
```

```
^
```

```
SyntaxError: invalid syntax
```

※ **SyntaxError** : 파이썬 언어의 문법이 틀렸을 경우 출력되는 에러

3. 예약어 사용이 불가능하다.(명령어는 가능하지만 원래 기능이 사라짐)

```
if = 30  
if
```

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

```
    if = 30
```

^

SyntaxError: invalid syntax

```
1 for  
2 if  
3 class  
4 def  
5 sum  
6 max  
7 min
```

예약어

명령어

```
1 import keyword  
2 keyword.kwlist
```

파이썬 예약어 확인

프로그래머가 **가장** 힘들어하는 일은?

권장사항

- 변수명의 첫 글자는 가급적 소문자로 만든다.
- 두 가지의 문자를 섞어서 변수명을 만들 경우 두 단어를 구분 할 수 있게 한다.

ex) numberList, NumberList, number_list 등

카멜 표기법

파스칼 표기법

스네이크 표기법

- 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

문자열 안에 작은따옴표나 큰따옴표를 포함시키고 싶을 때

she's gone

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

```
File "<ipython-input-17-c9d183e05d09>", line 1
```

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

```
SyntaxError: invalid syntax
```

문자열 안에 작은따옴표나 큰따옴표를 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 작은따옴표나 큰따옴표를 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 대입 하고 싶을 때

```
s = "여러줄로 구성된 \n문자열을 하나로 대입할 때"  
print(s)
```

여러줄로 구성된
문자열을 하나로 대입할 때

따옴표 3개를 연속으로 써서 양쪽 둘러 싸기

```
s = '''여러줄로 구성된  
문자열을 하나로  
대입할 때'''  
print(s)
```

여러줄로 구성된
문자열을 하나로
대입할 때

```
s = """여러줄로 구성된  
문자열을 하나로  
대입할 때"""  
print(s)
```

여러줄로 구성된
문자열을 하나로
대입할 때

이스케이프 코드

- Escape는 영문 그대로 일반적인 문자열 안에서 탈출한다는 뜻
- 출력 결과를 원하는 대로 변경하기 위한 용도

코드	설명
\n	개행(줄바꿈)
\t	수평 탭
\\	문자 "\"
\'	작은 따옴표(')
\"	큰 따옴표(")


```
s = "My name is SW"
```

M	y		n	a	m	e		i	s		S	W
0	1	2	3	4	5	6	7	8	9	10	11	12
-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

인덱싱(indexing)

- 무엇인가를 '가리킨다'는 의미 : 하나의 인덱스에 있는 값을 표시

슬라이싱(Slicing)

- 무엇인가를 '잘라낸다'는 의미 : 일정 범위의 인덱스에 있는 값을 표시

M	y		n	a	m	e		i	s		S	W
0	1	2	3	4	5	6	7	8	9	10	11	12

```
s = "My name is SW"  
print(s[0])  
print(s[8])  
  
---
```

M	y		n	a	m	e		i	s		S	W
-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

```
1 s = "My name is SW"
2 print(s[-2])
3 print(s[-1])
```

"My", "name", "is" 문자열 가져오기

M	y		n	a	m	e		i	s		S	W
0	1	2	3	4	5	6	7	8	9	10	11	12

```
1 s = "My name is SW"
2 print(s[0:2])
3 print(      )
4 print(      )
```

My
name
is

"My name" 문자열 가져오기

M	y		n	a	m	e		i	s		S	W
0	1	2	3	4	5	6	7	8	9	10	11	12

```
1 s = "My name is SW"
2 print(s[0:7])
3 print(    )
```

My name

My name

“is SW” 문자열 가져오기

M	y		n	a	m	e		i	s		S	W
0	1	2	3	4	5	6	7	8	9	10	11	12
-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

```
1 s = "My name is SW"
2 print(s[8:13])
3 print( )
4 print( )
```

is SW

is SW

is SW

모든 문자 가져오기

M	y		n	a	m	e		i	s		S	W
0	1	2	3	4	5	6	7	8	9	10	11	12
-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

```
1 s = "My name is SW"
2 print( )
3 print( )
```

```
My name is SW
My name is SW
```


다음과 같은 문자열에서 날짜와 날씨를 출력하시오.

```
s = "2020년 3월 3일의 날씨는 맑음입니다."
```

날짜 : 2020년 3월 3일

날씨 : 맑음

문자열 포매팅(Formatting)

- 문자열 안의 **특정한 값을 바꿔야 할 경우에 사용**

```
s = "오늘은 7월 7일 입니다."
```

```
s = "오늘은 7월 8일 입니다."
```

1. "%" 문자를 이용한 포매팅

```
day = 7  
s = "오늘은 7월 %d일 입니다." % day  
s
```

```
'오늘은 7월 7일 입니다.'
```

```
day = 8  
s = "오늘은 7월 %d일 입니다." % day  
s
```

```
'오늘은 7월 8일 입니다.'
```

문자열 포맷 코드

- 문자열 내 값 삽입

함수	설명
%s	문자열(string)
%c	문자 1개(char)
%d	0~9까지의 정수(digit)
%f	실수(float)
%%	Literal % (문자 '%' 자체)

- 2개 이상 값을 포매팅 할 때

```
month = 7
day = 7
s = "오늘은 %d월 %d일 입니다."%(month,day)
s
'오늘은 7월 7일 입니다.'
```

2. format 함수를 사용한 포매팅

```
month = 7
day = 7
s = "오늘은 {}월 {}일 입니다.".format(month, day)
s
```

'오늘은 7월 7일 입니다.'

3. f-string 포매팅

```
1 month = 7
2 day = 7
3 s = f"오늘은 {month}월 {day}일 입니다."
4 s
```

'오늘은 7월 7일 입니다.'

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

```
1 x = 100
2 y = 200
3 sum2 = x + y
4 print(?)
```

100와 200의 합은 300입니다.

함수	설명
count('문자')	문자열에 포함된 문자 개수 세기
find('문자')	문자 위치 알려주기
index('문자')	문자 위치 알려주기
'구분자'.join()	각각의 문자 사이에 '구분자' 삽입하기
upper()	소문자를 대문자로 바꾸기
lower()	대문자를 소문자로 바꾸기
lstrip()	왼쪽 공백 지우기
rstrip()	오른쪽 공백 지우기
strip()	양쪽 공백 지우기
replace('문자1', '문자2')	문자1을 문자 2로 바꾸기
split()	문자열 나누기



다음시간에는?

연산자