



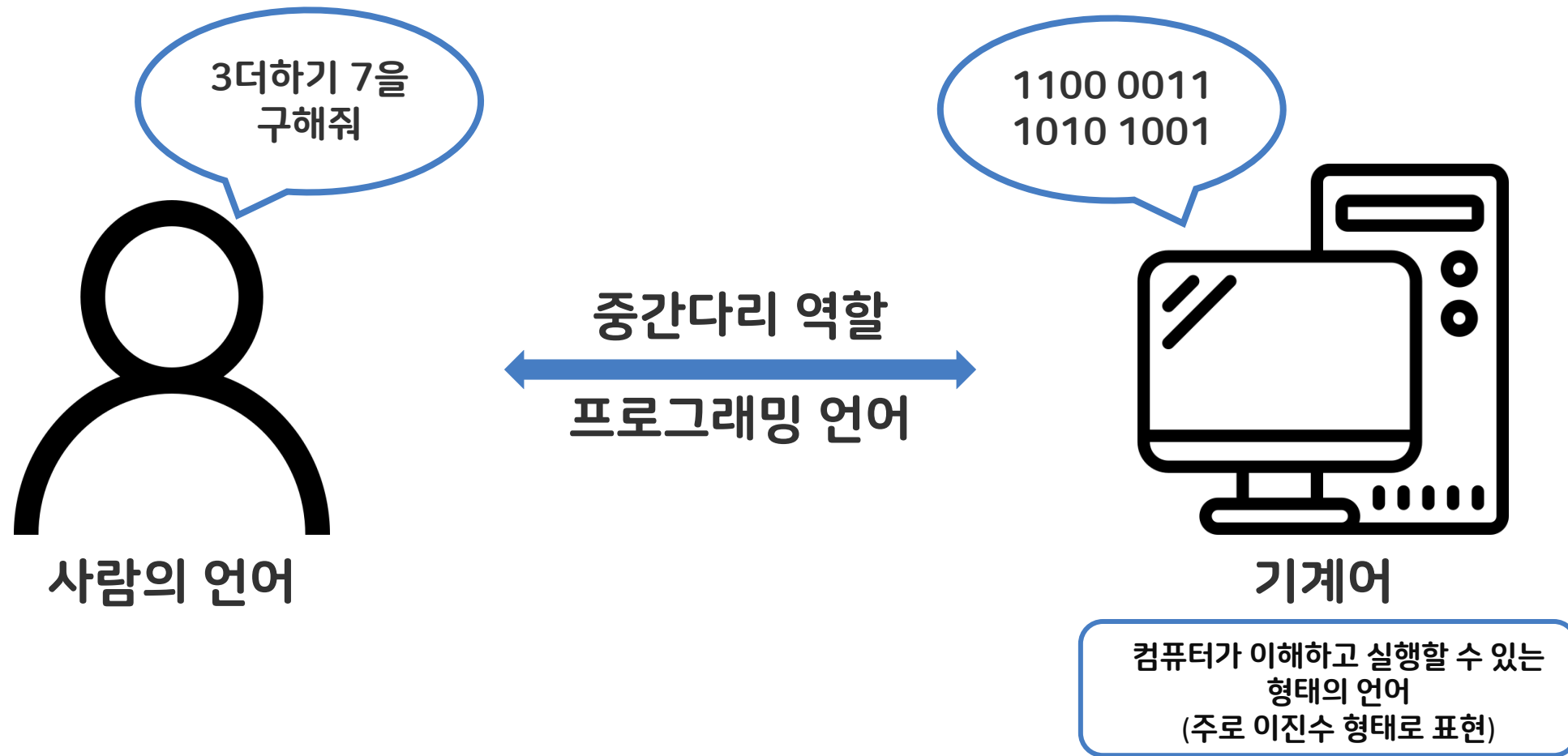
스마트인재개발원
Smart Human Resources Development

신재영



학습목표

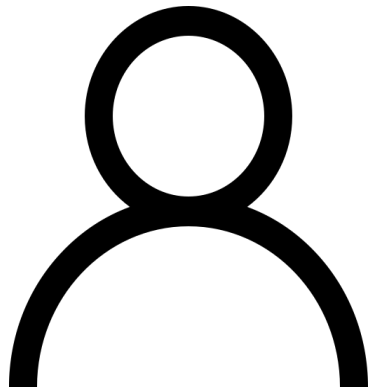
- Python의 개념, 특징을 이해한다.
- Python 개발환경구축을 할 수 있다.
- 변수 및 자료형에 대해 알 수 있다.
- 문자열 자료형을 이해하고 활용 할 수 있다.



컴퓨터를 이용하여 특정 문제를 해결하기 위한
프로그램을 작성하기 위해 사용되는 언어

고급 언어

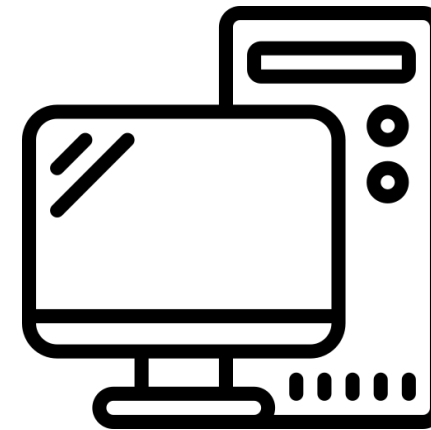
사람의 언어와 유사



C, Java, Python 등

저급 언어

기계어와 유사



어셈블리어





















TIOBE 2025 3월

<popularity>

프로그래밍 언어의
인기를 나타내는 지표

<기준>

- 해당 프로그래밍 언어를 이용
하는 엔지니어의 수
- 해당 프로그래밍 언어 관련
강좌 및 홈페이지의 수
- Google, Bing, Wikipedia
등에서 프로그래밍 언어가
검색된 횟수

Mar 2025	Mar 2024	Change	Programming Language		Ratings	Change
1	1			Python	23.85%	+8.22%
2	3	▲		C++	11.08%	+0.37%
3	4	▲		Java	10.36%	+1.41%
4	2	▼		C	9.53%	-1.64%
5	5			C#	4.87%	-2.67%
6	6			JavaScript	3.46%	+0.08%
7	8	▲		Go	2.78%	+1.22%
8	7	▼		SQL	2.57%	+0.65%
9	10	▲		Visual Basic	2.52%	+1.09%
10	15	▲		Delphi/Object Pascal	2.15%	+0.94%
11	14	▲		Fortran	1.70%	+0.48%
12	9	▼		Scratch	1.66%	+0.21%
13	12	▼		PHP	1.48%	+0.16%
14	17	▲		Rust	1.23%	+0.20%
15	13	▼		MATLAB	0.98%	-0.26%
16	21	▲		R	0.94%	+0.13%
17	11	▼		Assembly language	0.87%	-0.52%
18	24	▲		Ada	0.85%	+0.10%
19	19			Kotlin	0.85%	-0.11%
20	20			COBOL	0.84%	+0.01%

귀도 반 로섬(Guido Van Rossum)이 개발한 인터프리터 언어



Foreword for "Programming Python" (1st ed.)

Foreword for "Programming Python" (1st ed.)

This is the foreword I wrote for Mark Lutz' book "Programming Python" (1st ed.), published by O'Reilly.
See also my [foreword to the 2nd edition](#).

1989년, 크리스마스로 인해
회사가 1주일 쉬었다.

As Python's creator, I'd like to say a few words about its origins, adding a bit of personal philosophy.

취미를 찾다가 개발하기 시작

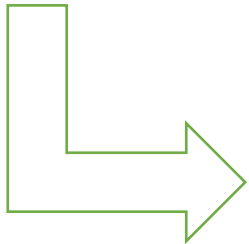
Over six years ago, in December 1989, I was looking for a "hobby" programming project that would keep me occupied during the week around Christmas. My office (a government-run research lab in Amsterdam) would be closed, but I had a home computer, and not much else on my hands. I decided to write an interpreter for the new scripting language I had been thinking about lately: a descendant of ABC that would appeal to Unix/C hackers. I chose Python as a working title for the project, being in a slightly irreverent mood (and a big fan of Monty Python's Flying Circus).

컴파일 언어 (Compiler)

빠른데 복잡함



```
public class Main {  
    public static void main(String[] args){  
        System.out.println("Hello");  
        System.out.println("World");  
    }  
}
```



컴파일러

```
11010011  
10101011  
11000011  
10101001
```

인터프리터 언어 (Interpreter)

쉬운데 느림



```
print('Hello')
```

인터프리터

Hello

```
print('World')
```

인터프리터

World

한줄씩 실행 !

장점

- 인간다운 언어

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

- 문법 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

- 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

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

단점

- 컴파일 언어에 비해 느림

Google

웹 프로그래밍
(Web programming)



데이터 분석
및
수치연산

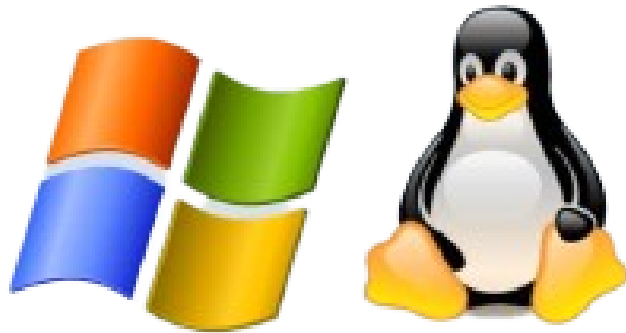


GUI 프로그래밍
(Graphical
User Interface)

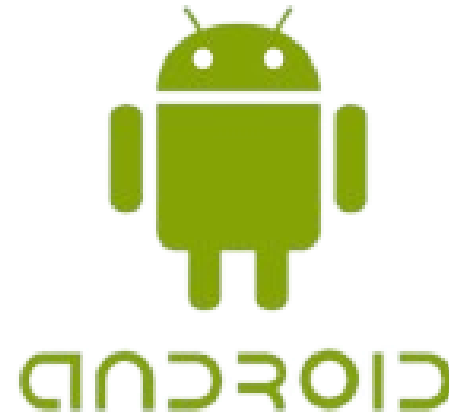


AI (Artificial Intelligence)

- Python으로 활용하기 힘든 것



운영체제



모바일





Python
패키지
(3.x)



에디터




Python을 활용한
데이터 분석, 어플리케이션 개발에
도움을 주는 플랫폼

google에서 anaconda검색

anaconda

이미지 동영상 쇼핑 뉴스 란 가상 환경 지도 Nicki Minaj Snake

검색결과 약 169,000,000개 (0.32초)

 Anaconda
<https://www.anaconda.com>

Anaconda: Unleash AI Innovation and Value


Streamline Every Phase of Your Workflow. Manage your projects from start to finish with **Anaconda's** data science tools. From ready-to-code environments to easy ...

Free Download




Pricing - Anaconda Navigator - Partners - Open Source - ...

Download 클릭!

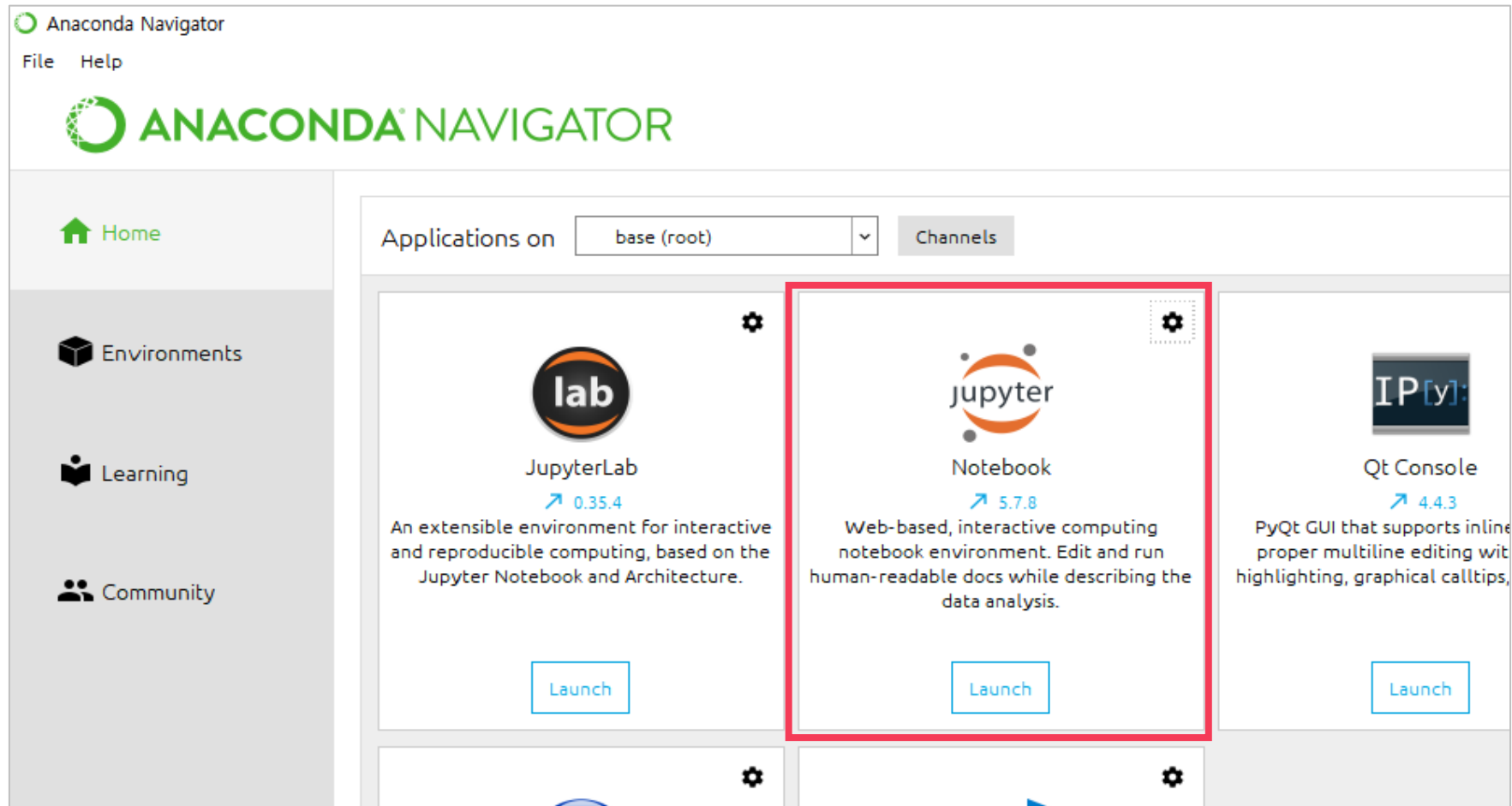
- ✓ Free distribution install
- ✓ Thousands of the most func
- ✓ Manage packages and envin
- ✓ Deploy across hardware and



Get Additional Installers

 |  | 

anaconda 실행 후 Jupyter notebook Launch 클릭



Command Mode



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

Edit Mode



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

실행 단축키

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

주석

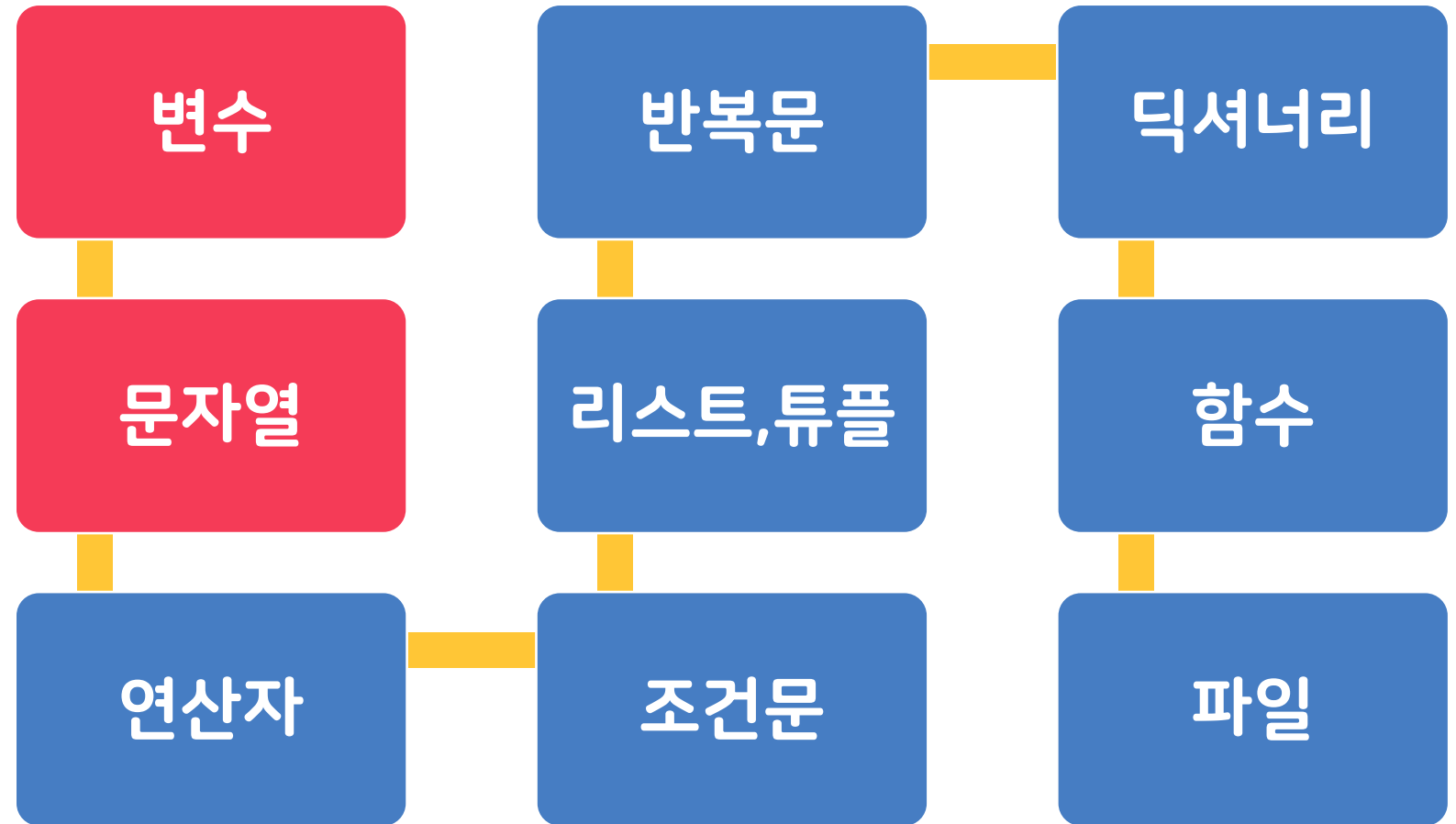
- 프로그래밍에 있어 내용을 메모하는 목적으로 사용
- 소스코드를 더 쉽게 이해할 수 있게 만드는 것이 주 목적
- 컴파일러와 인터프리터에 의해 일반적으로 무시되어 프로그램에 영향 X
- 파이썬은 “#”으로 주석

```
1+3 #1 더하기 3 실행
```

```
4
```

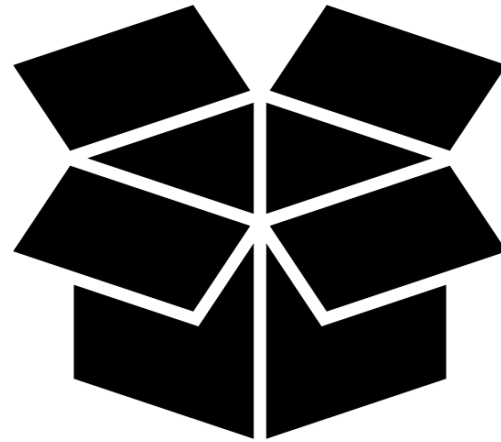



수업 진행방향



변수 (variable)

- 사전적 의미로는 “변화를 줄 수 있는 ” 또는 “변할 수 있는 수”
- 프로그래밍에서는 데이터를 담을 수 있는 공간



num = 3

↓ ↓ ↓

변수명 대입 값



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

동적 자료형

- 동적 자료형은 값을 넣어 줌과 동시에 자료형이 정해짐
- 변수 실행단계(Run time)에서 만들어 줌

1. 영문자, 숫자, 언더바(_)를 사용 가능

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 사용 불가능

또한 내장함수명 사용을 권장하지 않음

```
if = 30
if
```

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

```
    if = 30
```

```
        ^
```

```
SyntaxError: invalid syntax
```

키워드 확인

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

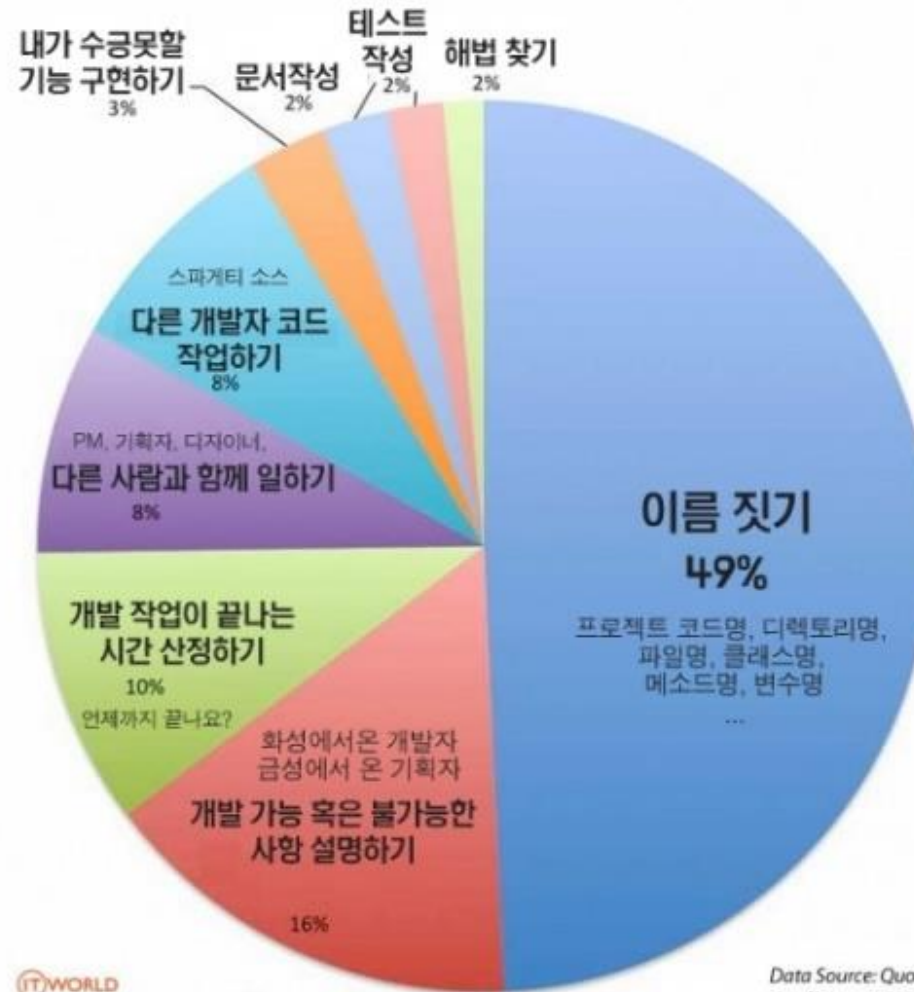
```
import keyword
print(keyword.kwlist)
```

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', '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']
```

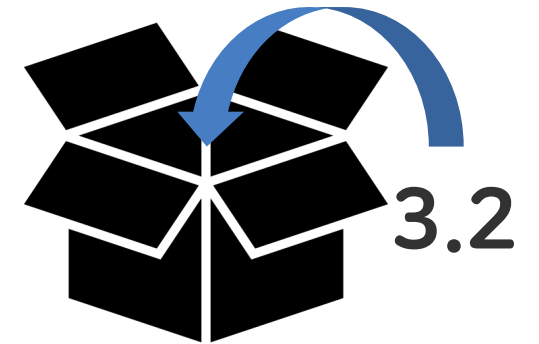
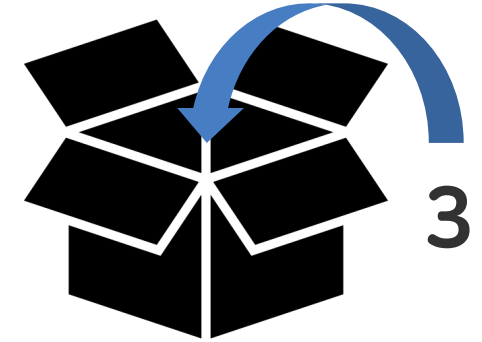
권장사항

- 변수명의 첫 글자는 항상 소문자로 지정
- 두가지의 문자를 섞어서 변수명을 만들 경우 두 단어를 구분
ex) number_list(Snake Case),
NumberList(Pascal Case),
numberList(Camel Case)

프로그래머가 가장 힘들어하는 일은?



자료형	설명
숫자형	-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':'jy', 'birth':'0223'}
집합 set	{1,2,3}
불리언 boolean	True, False



변수에 숫자(정수, 실수) 대입

- num1 변수에 숫자 13을 대입하시오.
- num2 변수에 숫자 25를 대입하시오.
- num2 변수에 숫자 25.45을 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.

👀 ✨ 여기서 발견할 수 있는 오류는?

```
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
```

이스케이프 코드

- 프로그래밍 할 때 사용할 수 있도록 미리 정의해둔 "문자 조합"

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

문자열 안에 작은따옴표나 큰따옴표를 포함시키고 싶을 때

she's gone

```
s1 = 'she\' gone'  
print(s1)
```

she' gone

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

- 이스케이프코드 활용

```
?
```

```
print(s)
```

파이썬

- 동적언어
- 직관적, 명시적, 간결함

여러 줄인 문자열을 변수에 대입 하고 싶을 때

- 따옴표 3개를 연속으로 써서 양쪽 둘러 싸기

```
'''
?
'''
print(s)
```

파이썬

- 동적언어
- 직관적, 명시적, 간결함

인덱싱(indexing)

- 무엇인가를 '가리킨다'는 의미

슬라이싱(Slicing)

- 무엇인가를 '잘라낸다'는 의미



인덱스 번호(위치) 기반 접근

```
s = "My name is JY"
```

M	y		n	a	m	e		i	s		J	Y
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

문자 인덱싱(indexing)

M	y		n	a	m	e		i	s		J	Y
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

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

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


문자열 슬라이싱(Slicing) - "My", "name", "is" 문자열 가져오기

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

```
s = "My name is JY"  
print(s[0:2])  
print(s      )  
print(s      )
```

```
My  
name  
is
```

문자열 슬라이싱(Slicing)

- "is JY" 문자열 가져오기

M	y		n	a	m	e		i	s		J	Y
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

```
s = "My name is JY"  
print(s[8:13])  
print( )  
print( )
```

```
is JY  
is JY  
is JY
```

다음과 같은 문자열에서 날짜와 날씨를 출력하시오.

```
day = "2020년 3월 3일의 날씨는 맑음입니다."
```

```
날짜: 2020년 3월 3일  
날씨 : 맑음
```

문자열 포매팅(Formatting)

- 문자열 안의 특정한 값을 바꿔야(삽입) 할 경우가 있을 때 사용

```
s = "오늘은 7월 7일 입니다."
```

```
s = "오늘은 7월 8일 입니다."
```

- %기호 포매팅
- format 함수 포매팅
- f문자열 포매팅

"%d"를 이용해서 정수 대입

```
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.'

2개 이상 값을 포매팅 할 때

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

오늘은 7월 7일 입니다.

f 문자열을 사용한 포매팅

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

오늘은 7월 7일 입니다.

lang변수에 '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.



다음시간에는?

연산자