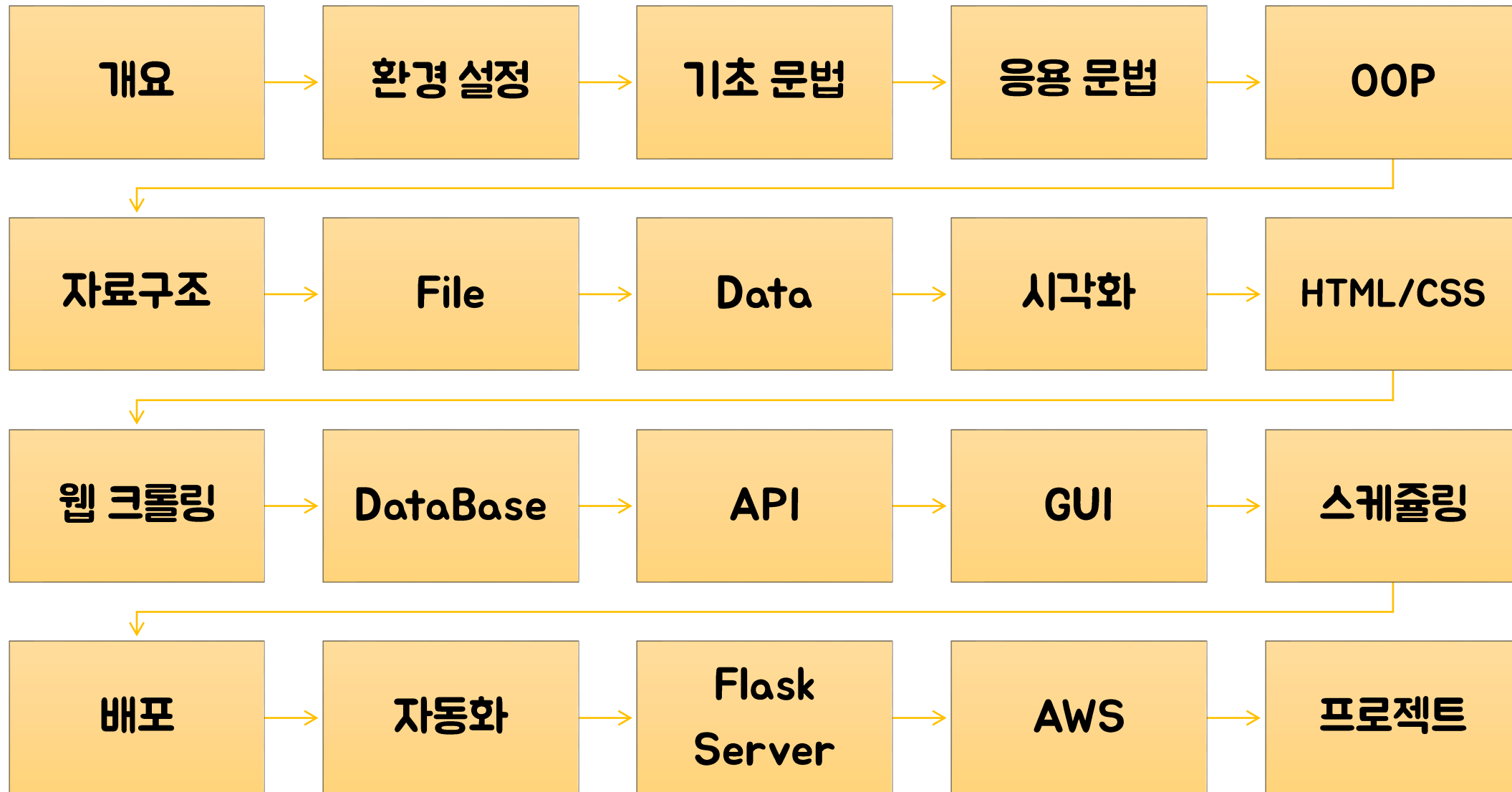




대한상공회의소  
서울기술교육센터

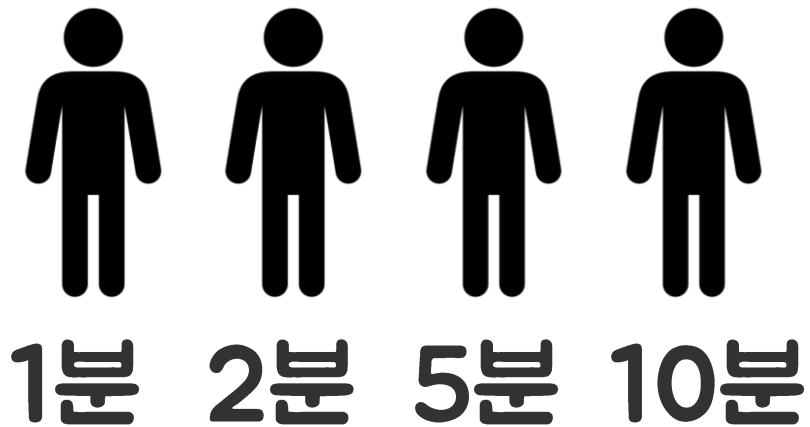
나 예 호 교수

- 수업 목표 : Python을 활용한 자동화 Script 프로젝트
- 세부 목표 : 프로그래밍 언어 기본기 다지기 + 논리적, 컴퓨팅 사고 훈련
- 수업 구성 : Python 기초 -> OOP -> 파일 처리 -> 자동화 -> 프로젝트
- 수업 방식 : 이론 수업, 코드 실습, 프로젝트



- 순서와 차례, 규칙을 바탕으로 문제를 해결하는 것

다리 건너기  
최대 2명 가능  
느린 쪽 시간 소요



손전등 1개  
손전등 필수



## - 기계와 의사소통을 할 수 있게 해주는 언어

1000 1011 0100 0101 1111 1000

1000 0011 1100 0100 0000 1100

0000 0011 0100 0101 1111 1100

mov eax, DWORD PTR [ebp-8]

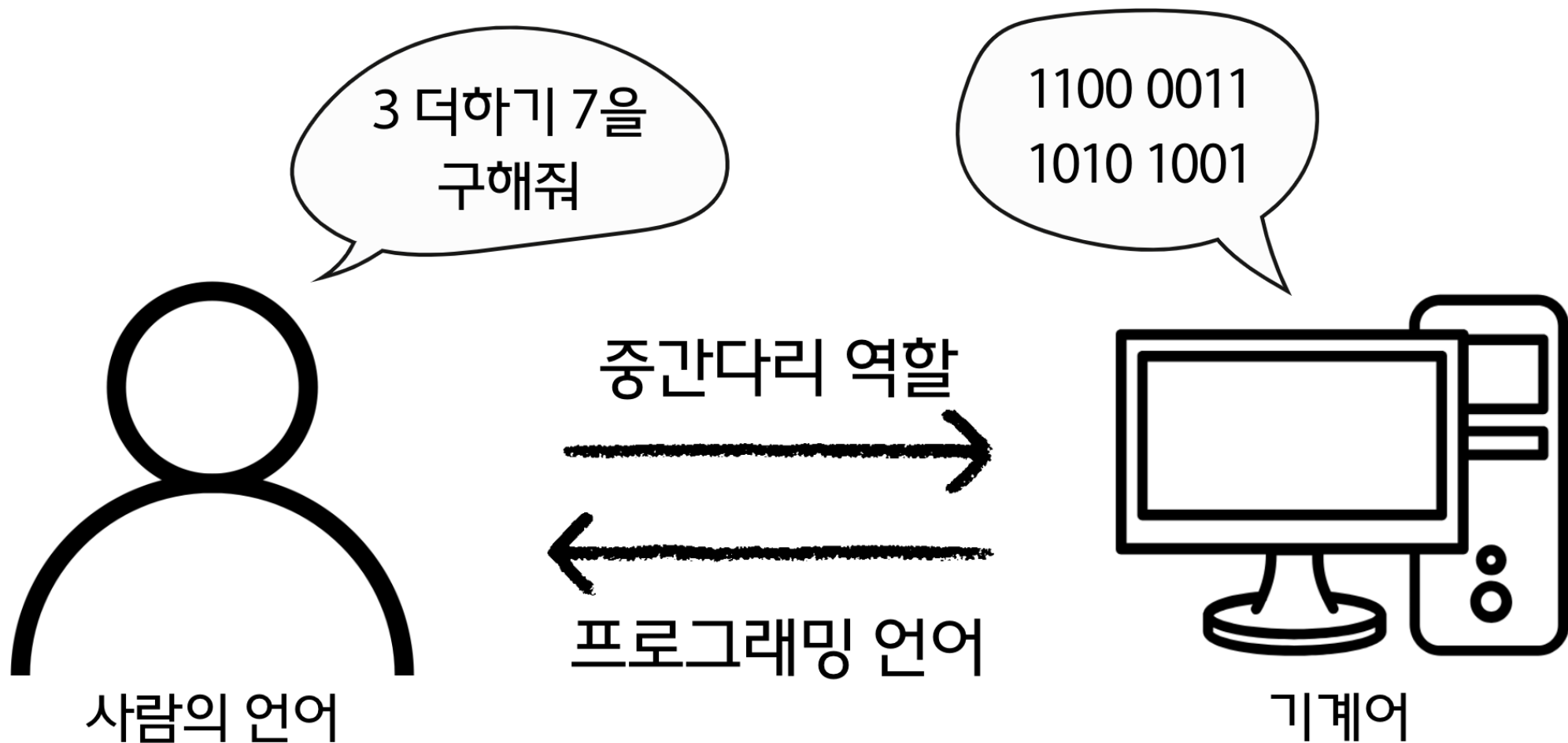
add esp, 12

add eax, DWORD PTR [ebp-4]

Machine Language  
기계어

Assembly Language  
어셈블리어

## - 기계와 의사소통을 할 수 있게 해주는 언어

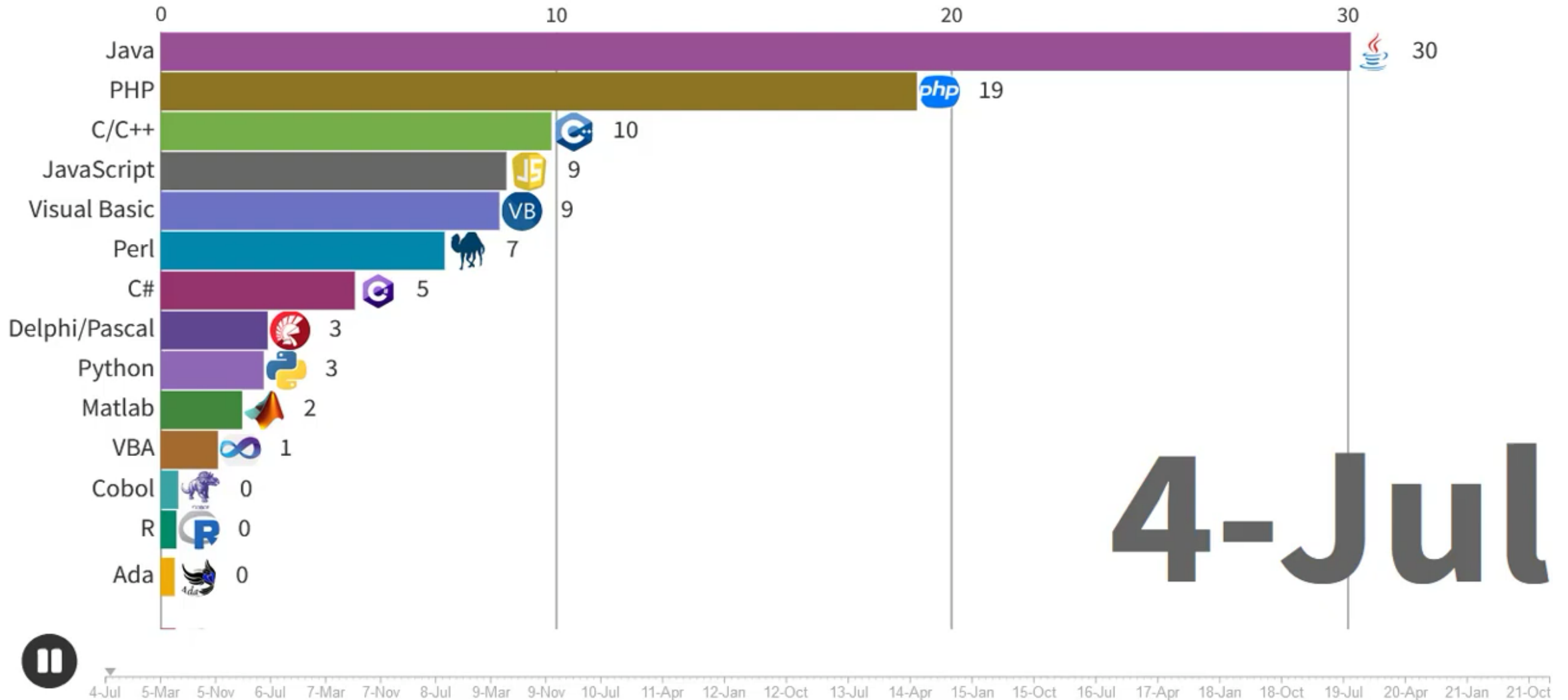


- 기계와 의사소통을 할 수 있게 해주는 언어

$result = a + b$





















[About us](#) ▾
 [Knowledge](#)
[News](#)
[Coding Standards](#)
[TIOBE Index](#)
[Contact](#)

[Products](#) ▾
 [Quality Models](#) ▾
 [Markets](#) ▾
 [Schedule a demo](#)

Jan 2025	Jan 2024	Change	Programming Language		Ratings	Change
1	1			Python	23.28%	+9.32%
2	3	⬆		C++	10.29%	+0.33%
3	4	⬆		Java	10.15%	+2.28%
4	2	⬇		C	8.86%	-2.59%
5	5			C#	4.45%	-2.71%
6	6			JavaScript	4.20%	+1.43%
7	11	⬆		Go	2.61%	+1.24%
8	9	⬆		SQL	2.41%	+0.95%
9	8	⬇		Visual Basic	2.37%	+0.77%
10	12	⬆		Fortran	2.04%	+0.94%

amazon

NETFLIX

 Spotify®



Instagram



- Life is short, you need Python.

Guido van Rossum (1991)

### 장점

직관적인 코드  
많은 라이브러리  
플랫폼 독립적  
오픈 소스  
자동 메모리 관리

### 단점

실행속도가 느리다  
멀티 스레딩 어려움  
모바일 개발 어려움  
메모리 사용량이 크다  
번거로운 배포





- Life is short, you need Python.

Guido van Rossum (1991)

적합

데이터 분석

머신 러닝, 딥 러닝

웹 개발(Flask)

업무 자동화

시스템 관리 스크립트

부적합

초고속 연산

모바일 앱 개발

-> Kotlin, Swift

게임 개발

-> Unity, Unreal



**PyCharm**

**Jupyter  
Notebook**

**VS Code**

- 독립적인 프로젝트를 위한 개별적 공간
- '의존성(dependencies)'과 '라이브러리(libraries)' 관리 용이
- `conda env list`
- `conda create --name myenv python=3.9`
- `conda activate myenv`
- `pip3 install jupyter notebook`

## Command Mode



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

## Edit Mode



- Esc : Command Mode로 전환
- Ctrl + z : 되돌리기
- Ctrl + y : 앞으로 되돌리기



# 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

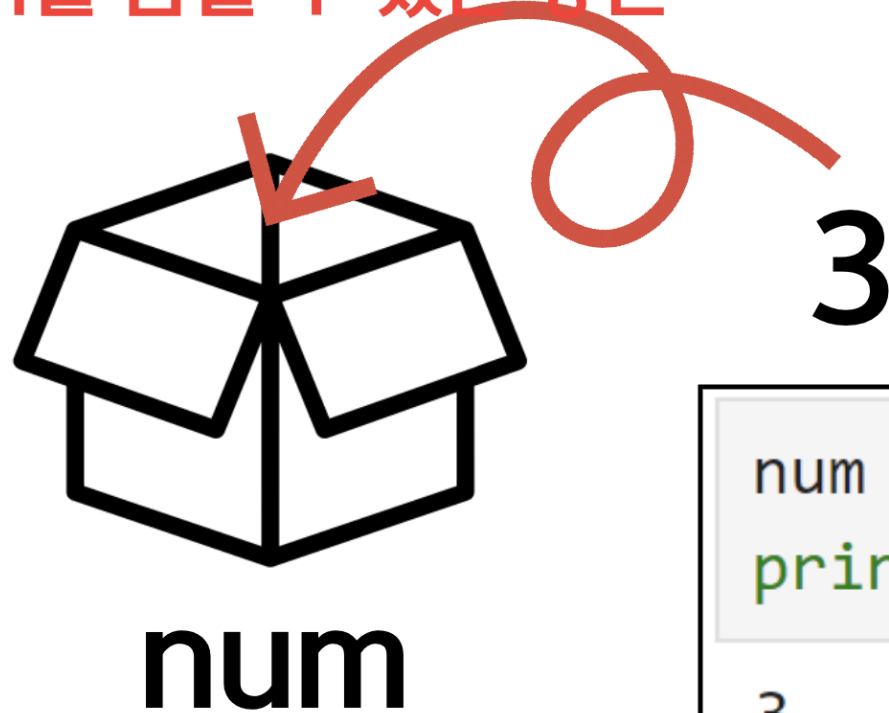
## 변수(variable)

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

num    =    3

↓        ↓        ↓

변수명    대입    값

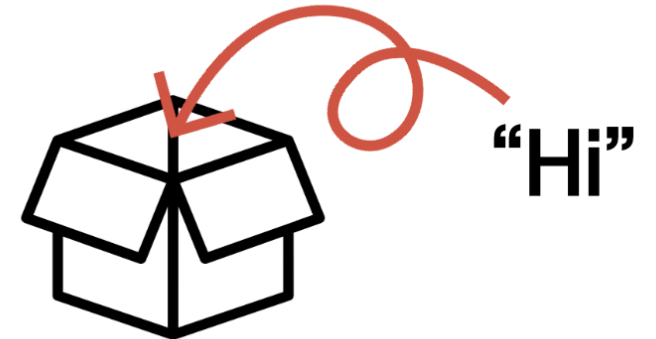
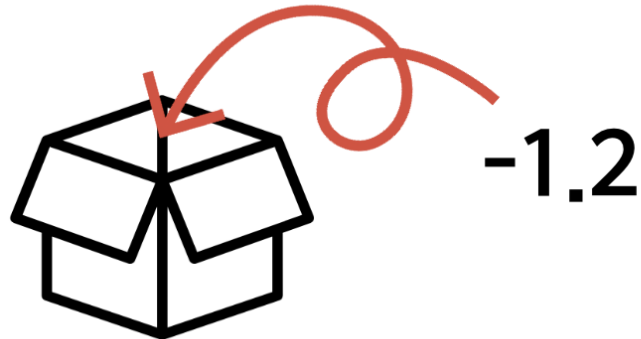


```
num = 3  
print(num)
```

3

- 데이터 저장 : 재사용
- 가독성 : 유지보수 용이
- 동적 처리 가능
- 메모리 관리
- 모듈화

자료형	설명
숫자(정수)	-2, -1, 0, 1, 2
숫자(실수)	-3.2, 3.14, -0.12, 3.0
문자열	'Hello World!', "Hi", "123"
논리	True, False



- num1 변수에 숫자 2을 대입하시오
- num2 변수에 숫자 4를 대입하시오
- num2 변수에 숫자 20을 대입하시오
- num3 변수에 숫자 3.141592를 대입하시오
- num4 변수에 숫자 -1.25를 대입하시오
- num5 변수에 숫자 3.0을 대입하시오

- str1 변수에 문자열 “Hello World!” 를 대입하시오
- str2 변수에 문자열 'Hi Python' 을 대입하시오
- bool1 변수에 True를 대입하시오
- bool2 변수에 False를 대입하시오

1. 영문자, 숫자, Underscore(\_)를 사용할 수 있다

- 단, 영문자는 대소문자를 다르게 인식한다

2. 숫자로 시작할 수 없다

3. 키워드 사용이 불가하다

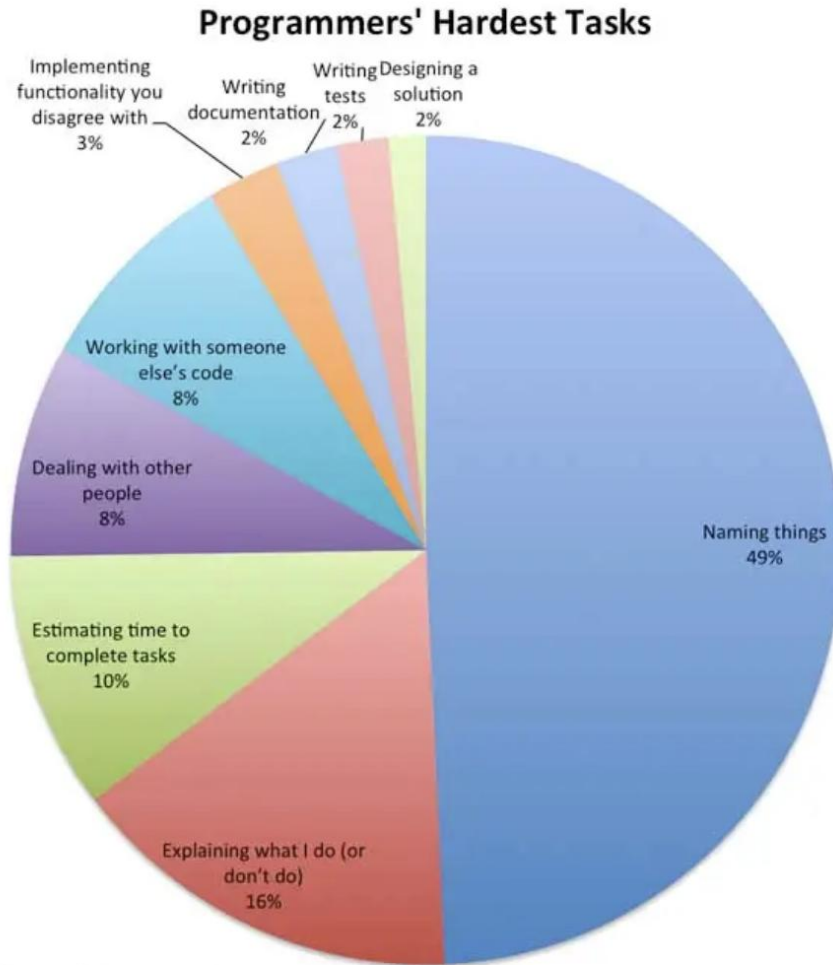
```
from keyword import kwlist
```

```
print(kwlist)
```



필수는 아니지만 **권장**되는 규칙들

1. 변수 이름의 **첫 글자**는 **소문자**로 시작한다
2. 여러 단어를 나열할 경우 표기법을 사용한다
  - Camel 표기법 : todayLunchMenu
  - Snake 표기법 : today\_lunch\_menu



Data Source: Quora/Ubuntu Forums  
Total Votes: 4,522



Naming things in code is *really* hard

CREDIT: IMAGE CREDIT: ITWORLD/PHIL JOHNSON

"I do spend a lot of time as well worrying about the names of anything that can be given a name when I am programming."

willcodejavaforfood

"I have to agree that naming is an art." Otávio Décio

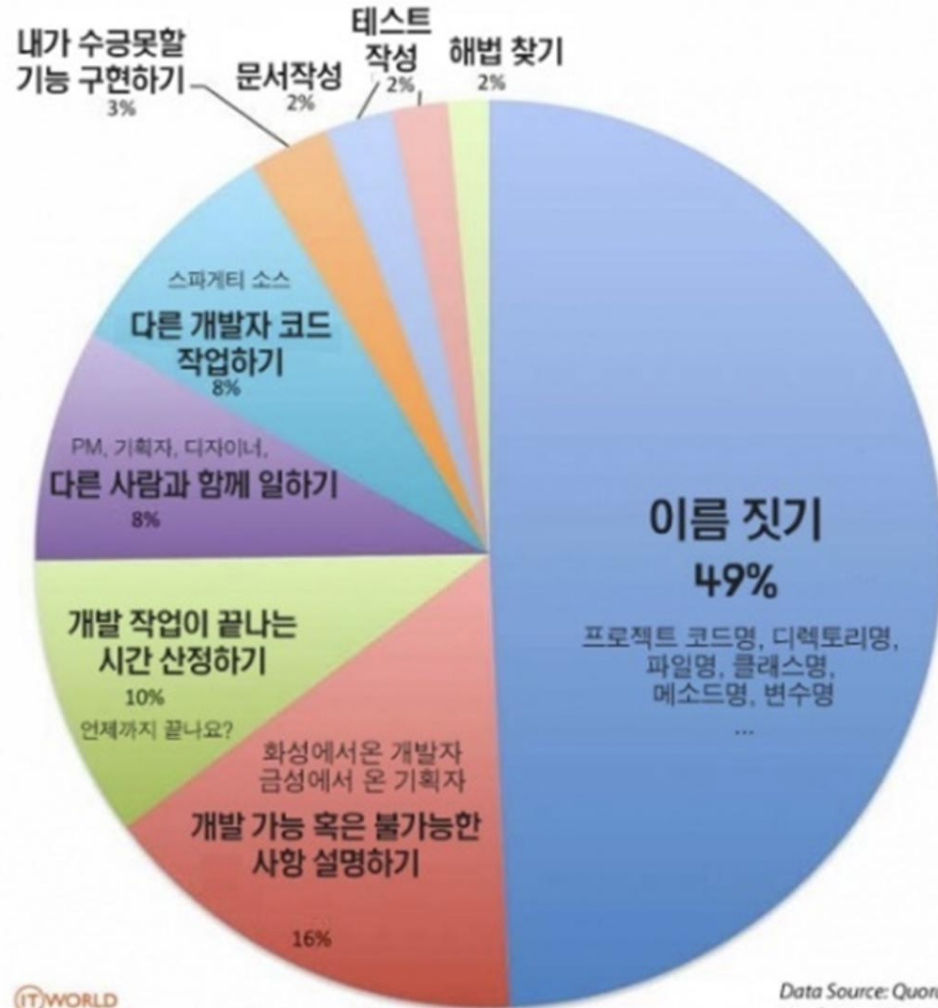
"...its one of the most important things if you want your code to be readable by others." Gerjes Handal

"Often times, however, the inability to come up with a name may be a hint to something wrong with your design. Does your method have too many responsibilities? Does your class encapsulate a coherent idea?" Brad Barker

"One lesson I heave [sic] learned, is that if you can't find a name for a class, there is almost always something wrong with that class: you don't need it, it does too much" Toon Krijthe

"It's good that it's difficult. It's forcing you to think about the problem, and what the class is actually supposed to do." JW.

## 프로그래머가 가장 힘들어하는 일은?



"프로그래밍할 때 이름을 지을 수 있는 모든 것의 이름에 대해 걱정하는 데도 많은 시간을 보냅니다." [willcodejavaforfood](#)

"나는 명명이 예술이라는 데 동의해야 합니다." [오타비오 데시오](#)

"...다른 사람들이 코드를 읽을 수 있게 하려면 가장 중요한 것 중 하나입니다." [Gerjes Handal](#)

"그러나 종종 이름을 생각해 낼 수 없다는 것은 디자인에 문제가 있다는 힌트일 수 있습니다. 메서드에 책임이 너무 많습니까? 클래스가 일관된 아이디어를 캡슐화합니까?" [브래드 바커](#)

"내가 얻은 한 가지 교훈은 클래스 이름을 찾을 수 없다면 거의 항상 그 클래스에 문제가 있다는 것입니다. 필요하지 않은 것이고 너무 많은 일을 하는 것입니다." [Toon Krijthe](#)

"어려운 건 좋은 일이에요. 문제에 대해 생각하게 만들고, 수업이 실제로 무엇을 해야 하는지 생각하게 만드는 거죠." [JW.](#)

- 외부 모듈, 라이브러리를 불러오는 기능

Python **생태계 확장**의 1등 공신

연산자	기호
산술 연산자	+ - * / // %
지수 연산자	**
대입(복합) 연산자	= += -= *= /= //= %=
관계(비교) 연산자	> >= < <= == !=
논리 연산자	not and or
삼항 연산자	a if 조건식 else b

## 나누기, 나머지, 나누기(몫) 구하기

```
num1 = 10  
num2 = 7  
  
print(num1/num2)  
print(num1%num2)  
print(num1//num2)
```

실행결과 =>

## 문자열 더하기

```
str1 = "안녕"  
str2 = "하세요"  
  
print(str1 + str2)
```

안녕하세요

```
str1 = "10"  
str2 = "7"  
  
print(str1 + str2)
```

107



## 숫자, 문자열 더하기

```
num1 = 10  
str2 = "7"  
  
print(num1 + str2)
```

```
num1 = 10  
str2 = "7"
```

```
print(str(num1) + str2)  
print(num1 + int(str2))
```

```
107  
17
```

다음 코드를 완성하여 다음과 같은 결과를 출력하시오.

```
num1 = 23  
num2 = 3  
?
```

더하기 결과 : 26

빼기 결과 : 20

곱하기 결과 : 69

나누기 결과 : 7.666666666666667

다음 코드에서 변수 num1과 num2를 **키보드로 입력** 받아서 결과를 계산하시오.

```
num1 = #키보드 입력  
num2 = #키보드 입력  
?
```

더하기 결과 : 26

빼기 결과 : 20

곱하기 결과 : 69

나누기 결과 : 7.6666666666666667

1

```
num = input("정수를 입력하세요 >> ")
```

정수를 입력하세요 >>

2

```
num ← input("정수를 입력하세요 >> ")
```

정수를 입력하세요 >>

3

num

'133'

← 문자열

## 문자를 숫자로 바꾸는 방법

- `int(문자열)` : 문자열을 정수로 변환
- `float(문자열)` : 문자열을 실수로 변환

```
num = int(input("정수를 입력하세요 >> "))
```

```
num
```

```
정수를 입력하세요 >> 123
```

```
123
```

다음 코드에서 변수 num1과 num2를 **키보드로 입력** 받아서 결과를 계산하시오.

```
num1 = ? #키보드 입력  
num2 = ? #키보드 입력  
?
```

정수를 입력하세요 >> 3

정수를 입력하세요 >> 7

더하기 결과 : 10

빼기 결과 : -4

곱하기 결과 : 21

나누기 결과 : 0.42857142857142855

Python, 머신러닝, 딥러닝 점수를 키보드로 입력 받아 합계와 평균을 출력하시오.

```
print("합계 : {}".format(?))  
print("평균 : {}".format(?))
```

```
python 점수 입력 >> 100  
머신러닝 점수 입력 >> 80  
딥러닝 점수 입력 >> 60  
합계 : 240  
평균 : 80.0
```

<= 입력 값



초를 입력 받아 “00시간 00분 00초” 형태로 출력하시오.

```
time = int(input("시간 입력 >> "))  
hour = ?  
minute = ?  
second = ?  
print("{}시간 {}분 {}초".format(hour, minute, second))
```

시간 입력 >> 7533  
2시간 5분 33초

시간 입력 >> 1123  
0시간 18분 43초

시간 입력 >> 3723  
1시간 2분 3초

## 문자열 곱하기

```
s = "x"  
print(s*10)
```

XXXXXXXXXX

```
s = "안녕하세요"  
print(s*2)
```

안녕하세요안녕하세요

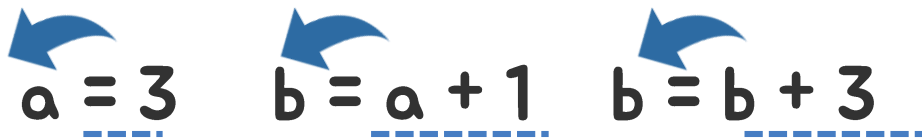
## 지수 연산자 (\*\*)

```
num = int(input("정수 입력 >>"))  
power = int(input("지수 입력 >>"))  
print("{}의 {}승은 {}입니다.".format(num, power, num**power))
```

정수 입력 >>2

지수 입력 >>3

2의 3승은 8입니다.

<p>= (대입 연산자)</p>	<p></p>
<p>+=, -=, *=, /=, %= (복합 대입 연산자)</p>	<p><math>a += b \rightarrow a = a + b</math> <math>a -= 3 \rightarrow a = a - 3</math></p>

## 대입(복합) 연산자 실습

```
num = 27
```

```
num += 3
```

```
num
```

```
num = 27
```

```
num = num + 3
```

```
num  
30
```

## 대입(복합) 연산자 실습

```
num = 27
```

```
num += 3
```

```
num += 3
```

```
num += 3
```

```
num
```

```
num = 27
```

```
num = num + 3 30
```

```
num = num + 3 33
```

```
num = num + 3 36
```

```
num
```

36