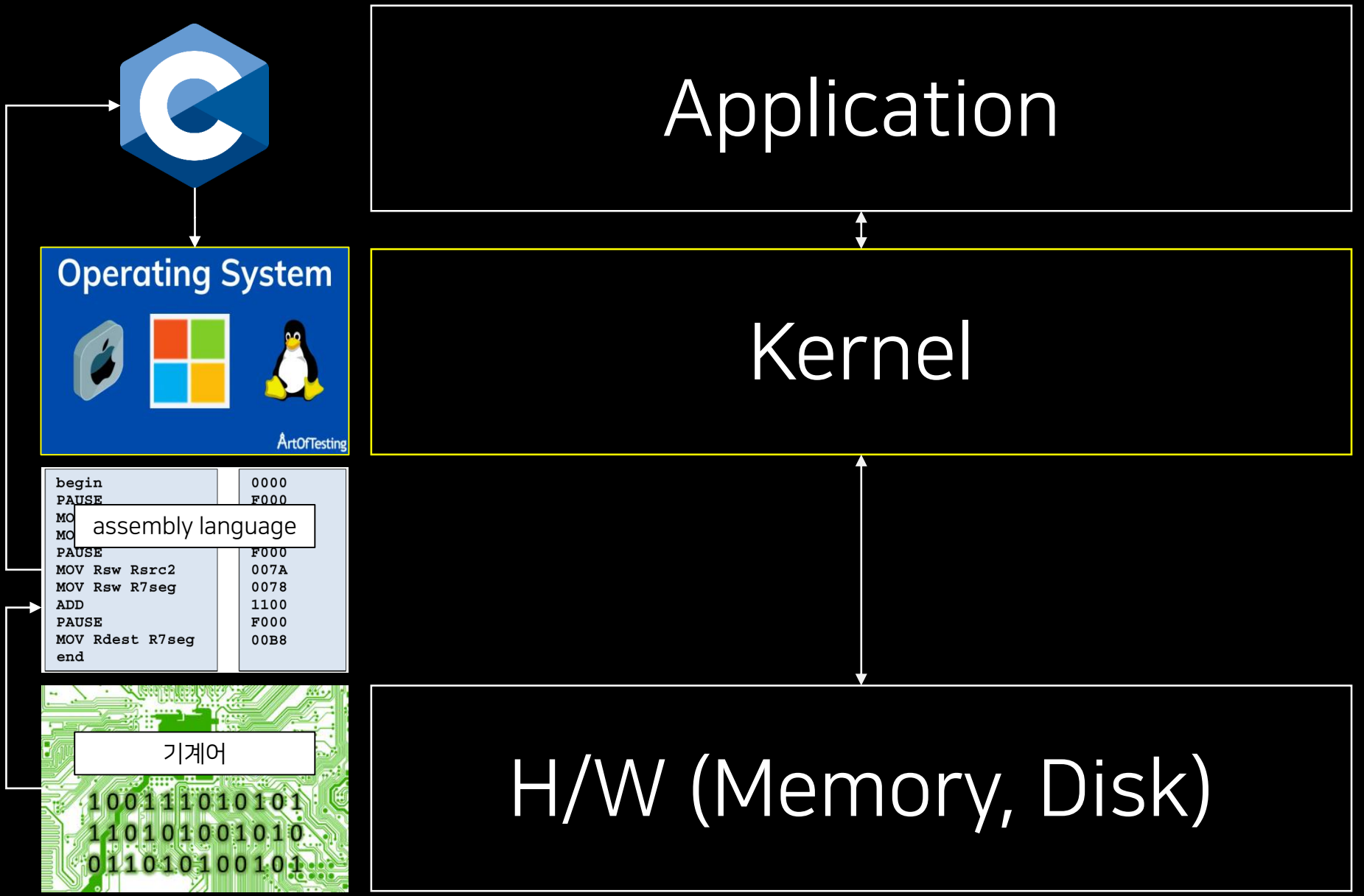
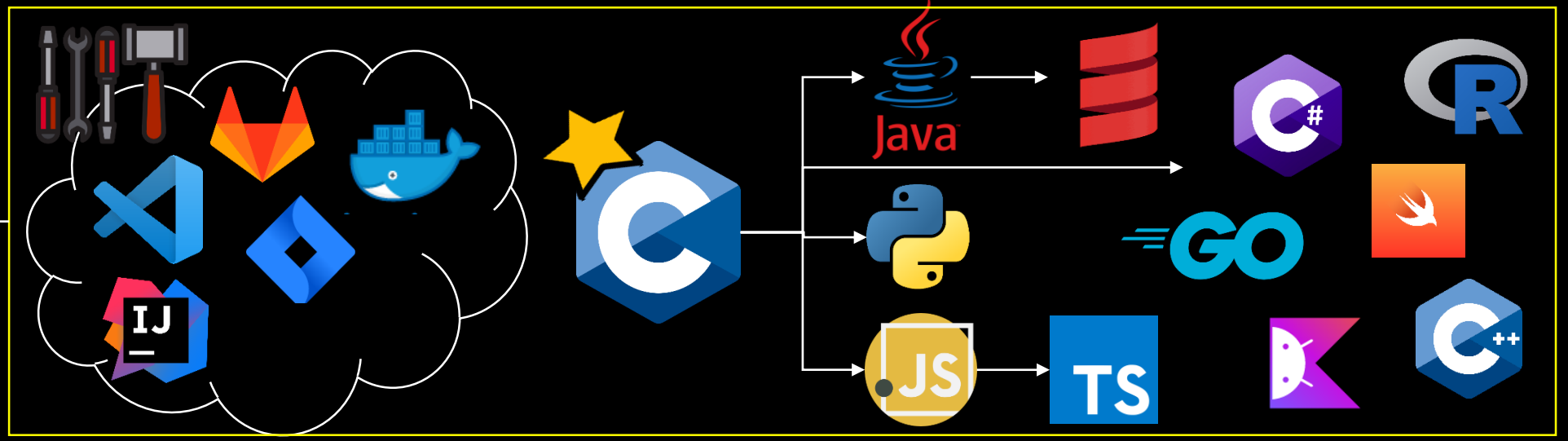
C Language Genealogy : 운영체제 개발 언어



C Language Genealogy : 모든 언어의 기초 토대



Build, Test, Release

Application

Operating System

ArtOfTesting

Kernel

여러 언어 비교를 통한 Entry Point 잡기

```c
# include <stdio.h>

int MySum(int num1, int num2):
    int intSum;
    intSum = sum1 + sum2;

    return intSum

void main():
    int n1, n2;

    int s = MySum(n1, n2);
```

```java
public class SumExample{

    public static int mySum(int num1, int num2){
        int intSum;
        intSum = num1 + num2;

        return intSum
    }

    public static void main(String[] args){
        int n1, n2;

        int s = mySum(n1, n2)
    }

}
```

Entry Point “Main”

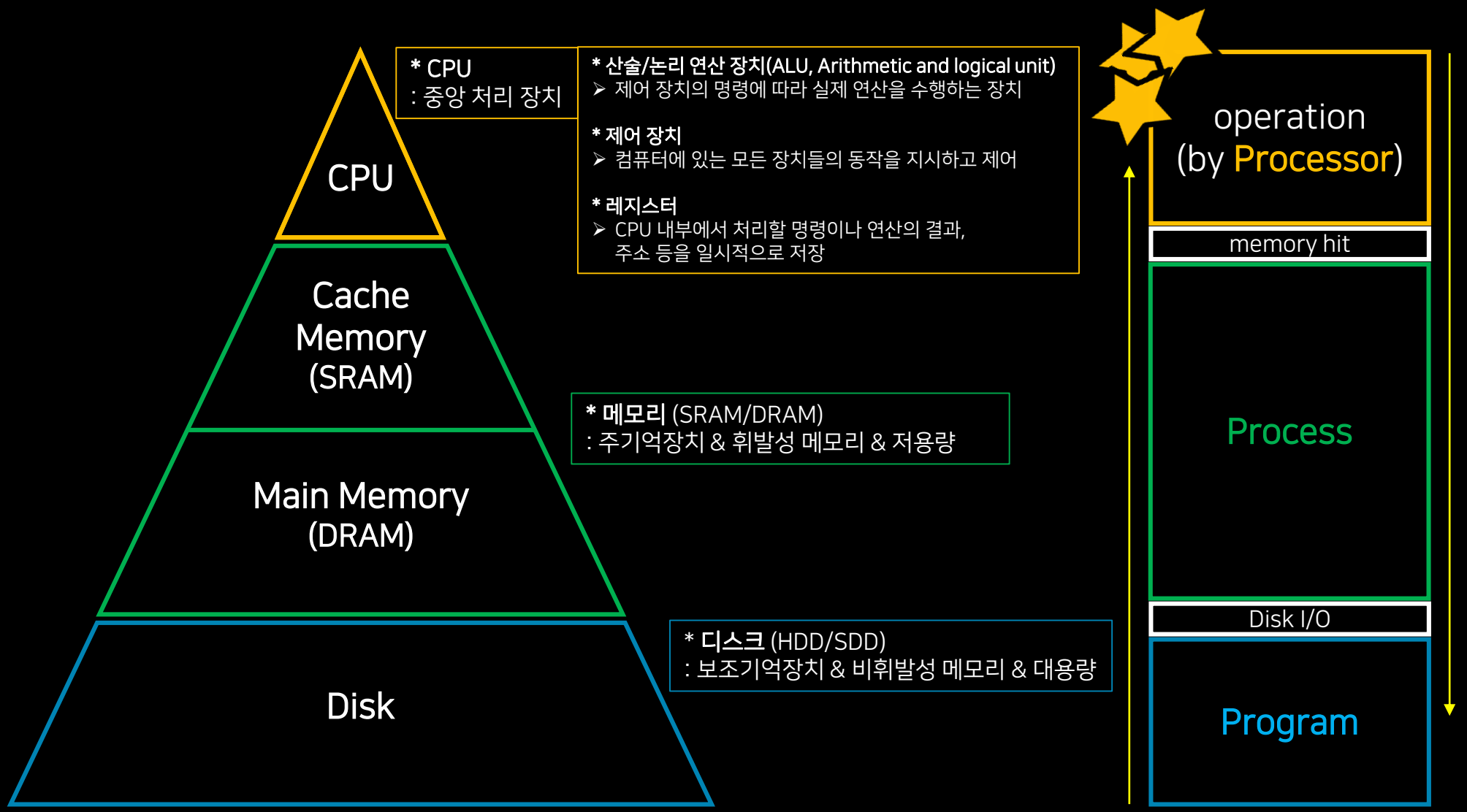
```python
def my_sum(n1: int, n2: int) -> int:
    intSum = n1 + n2

    return intSum

def main():
    n1, n2 = 1, 2

    s = my_sum(n1, n2)

if __name__ == '__main__':
    main()
```

C language Primitive Type

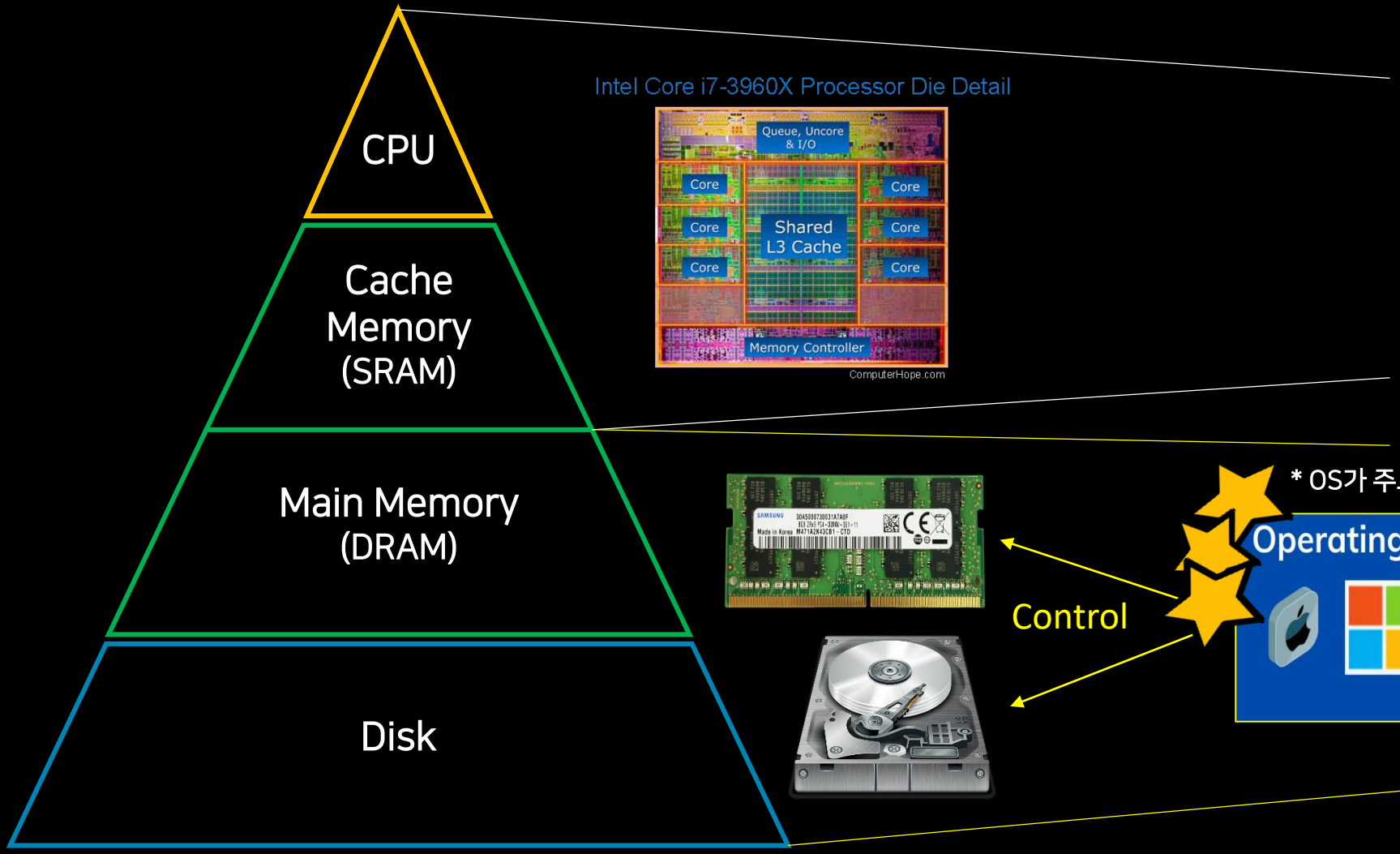
기본 자료형			크기		값의 범위
			bit	Byte	
정수	signed	short	16	2	-32,768 ~ 32,767
		int	32	4	-2,147,483,648 ~ 2,147,483,647
		long	32	4	
		long long	64	8	~
	unsigned	unsigned short	16	2	0 ~ 65,535
		unsigned int	32	4	0 ~ 4,294,967,295
		unsigned long	32	4	
		unsigned long long	64	8	~
실수	X	float	32	4	3.4*10-38 ~ 3.4*1038
		double	64	8	1.7*10-308 ~ 1.7*10308
문자	signed or unsigned	char	8 (한글 : 16)	1 (한글 : 2)	-128 ~ 127 (숫자-문자 매핑)
	signed	char	8	1	-128 ~ 127
	unsigned	unsinged char	8	1	0 ~ 255

※ bool 형 (True, False)는 각각 (1, 0)과 동일

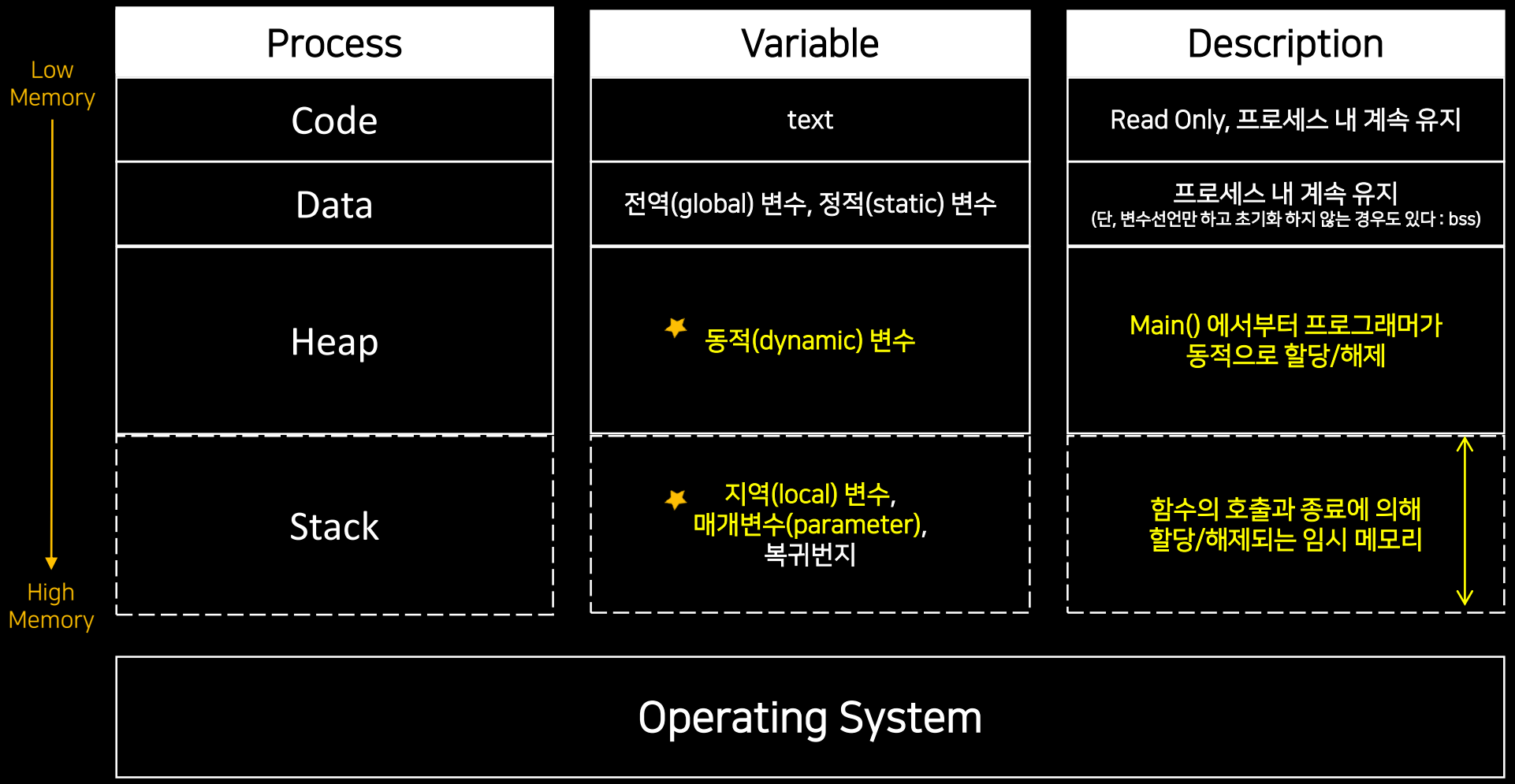
Basic Components of H/W : CPU, RAM, Hard Drive



What the OS manages

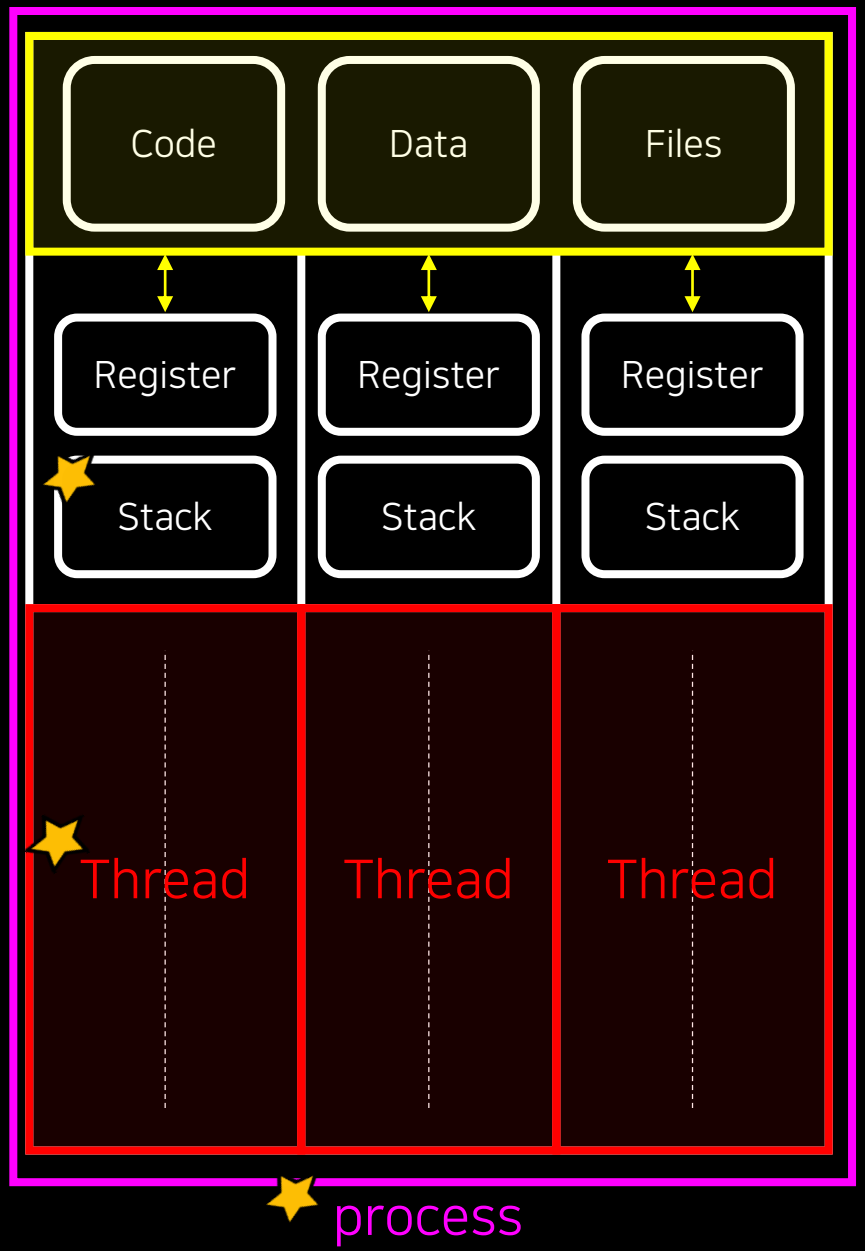


Memory Layout : Code, Data, Heap, Stack



※ 매개변수(parameter) vs 아규먼트(argument) 용어 구분

Multi-Threading



Thread Synchronization

쓰레드(Thread)는 컴퓨터 프로그램의 실행 흐름의 단위로, 프로세스 내에서 독립적으로 실행되는 작업의 일부

일반적으로 한 프로세스는 하나의 실행 흐름인 메인 쓰레드(Main Thread)를 가지며, 메인 쓰레드는 프로그램의 시작부터 종료까지 실행

프로세스(Process)는 실행 중인 프로그램의 인스턴스

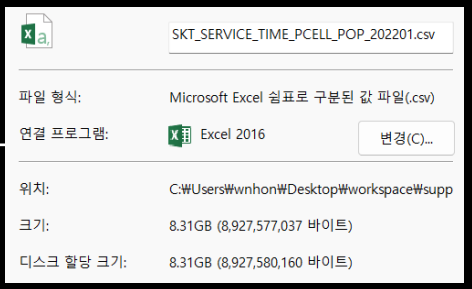
실행 가능한 파일로서 정적인 상태 (binary file)인 프로그램에서 운영 체제에 의해 프로세스로 메모리에 로드

How to execute all process at the same time?

△ 8 GB ~ 32 GB Memory

Load / transform

Is it possible?



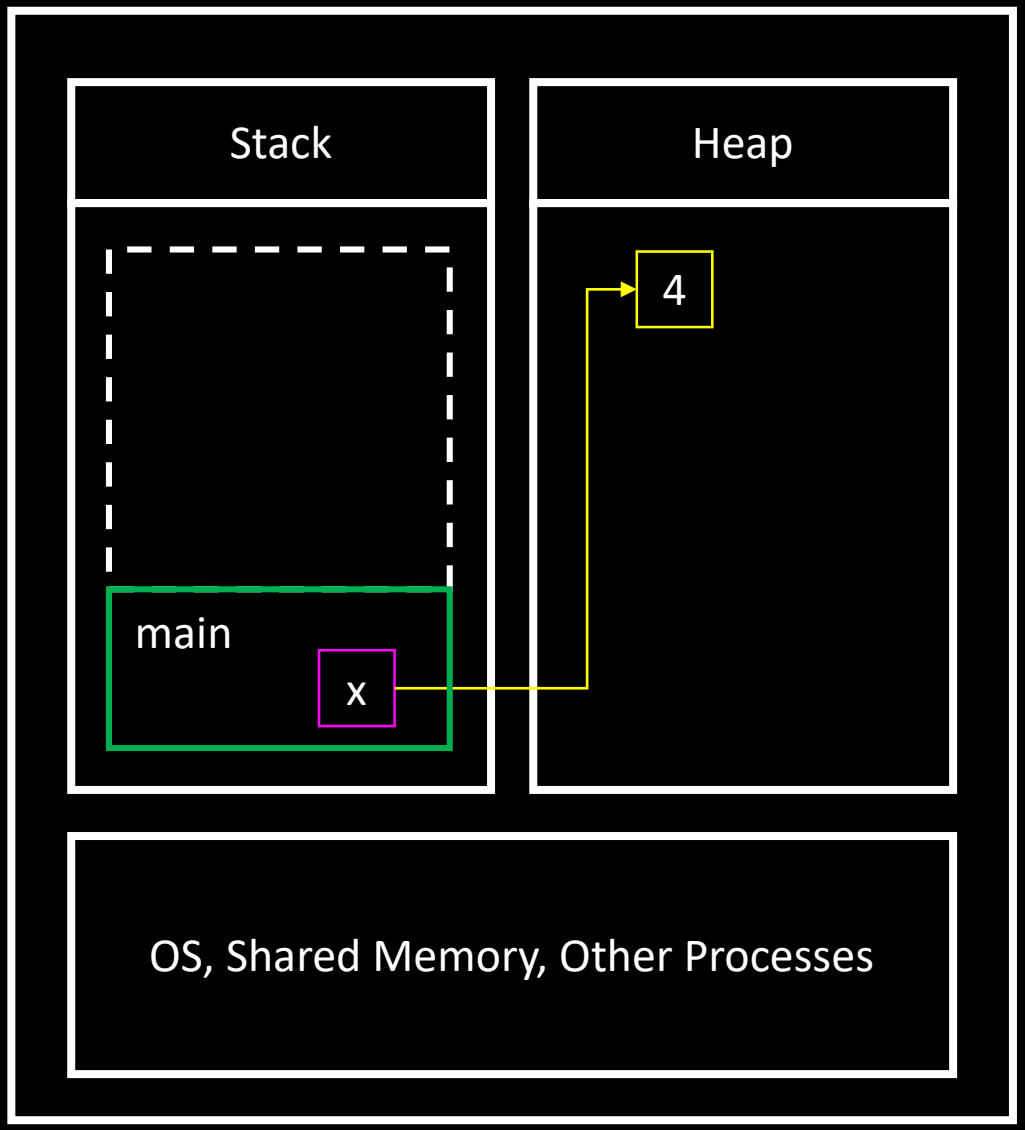
8 GB Disk File

exist

```
top - 11:02:38 up 38 min,  1 user,  load average: 0.34, 0.49, 0.77
Tasks:  255 total,    1 running,  254 sleeping,    0 stopped,    0 zombie
%Cpu(s):  1.6 us,  0.6 sy,  0.0 ni, 96.5 id,  1.3 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  8069036 total,   682140 free,  4108228 used,  3278668 buff/cache
KiB Swap:  3906556 total,  3906556 free,        0 used.  3389920 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 4446 tecmint   20   0 1647364 288376  53268 S   4.0  3.6   1:00.82 cinnamon
 5161 tecmint   20   0 2447576 468128 111768 S   1.7  5.8   1:02.30 Web Content
 2141 root      20   0  442288 101188  88744 S   1.0  1.3   1:11.92 Xorg
    7 root      20   0       0      0      0 S   0.3  0.0   0:04.30 rcu_sched
 2029 root      39  19 4666272 451468  21056 S   0.3  5.6   0:47.52 java
 4468 tecmint   20   0 1179392  70096  40428 S   0.3  0.9   0:04.62 nemo
 5253 tecmint   20   0 2365636 479080 107280 S   0.3  5.9   1:02.48 Web Content
 5451 tecmint   20   0 2446420 420728 116488 S   0.3  5.2   1:33.00 Web Content
 5528 tecmint   20   0  509204  42564  30632 S   0.3  0.5   0:03.77 gnome-terminal-
 5549 tecmint   20   0 1726164 533396 228764 S   0.3  6.6   1:24.22 chrome
 5938 tecmint   20   0 1503040 296364  57556 S   0.3  3.7   0:10.52 chrome
 7452 tecmint   20   0 1194392 330084  87076 S   0.3  4.1   0:28.61 chrome
```

△ Daemon Process (background)

Memory Layout : Code, Data, Heap, Stack




```python
def main():
    x = 4

if __name__ == '__main__':
    main()
```

변수(variable)은 값이 아니다.
값의 위치를 가르키는 주소값이다.

```
x = 4
```

값을 비교하는 연산자는 "=="
x == 4 # True

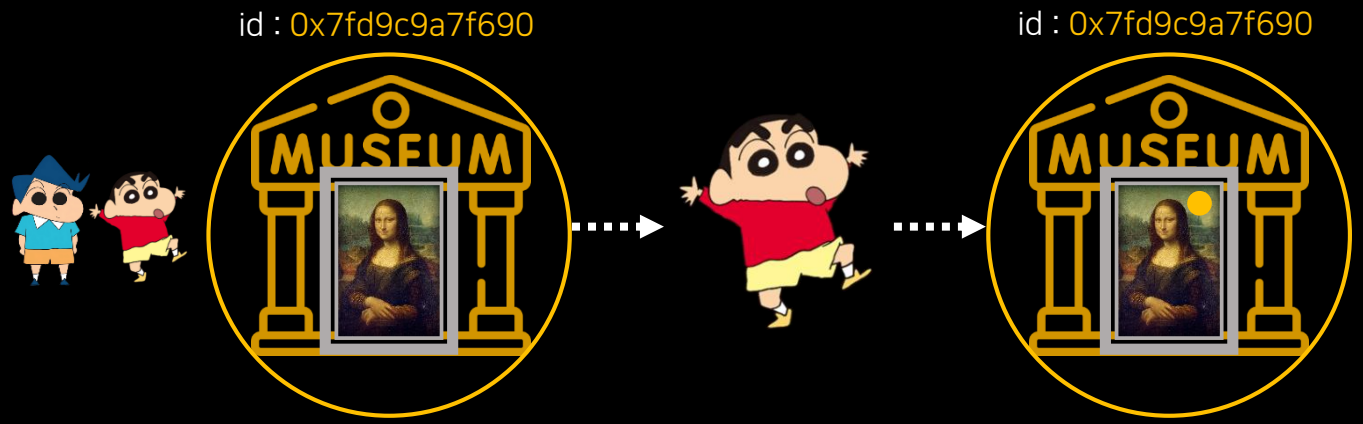
주소값을 비교하는 연산자는 "is"
x is 4 # True
id(x) # 2282602955088
id(4) # 2282602955088

단, cpython -5 ~ 256 는 자주 사용하는 숫자로 지정하여,
프로세스 초기화시 캐시를 만들어 고정된 주소값을 사용.
(Object Interning : Immutable한 객체 재사용)

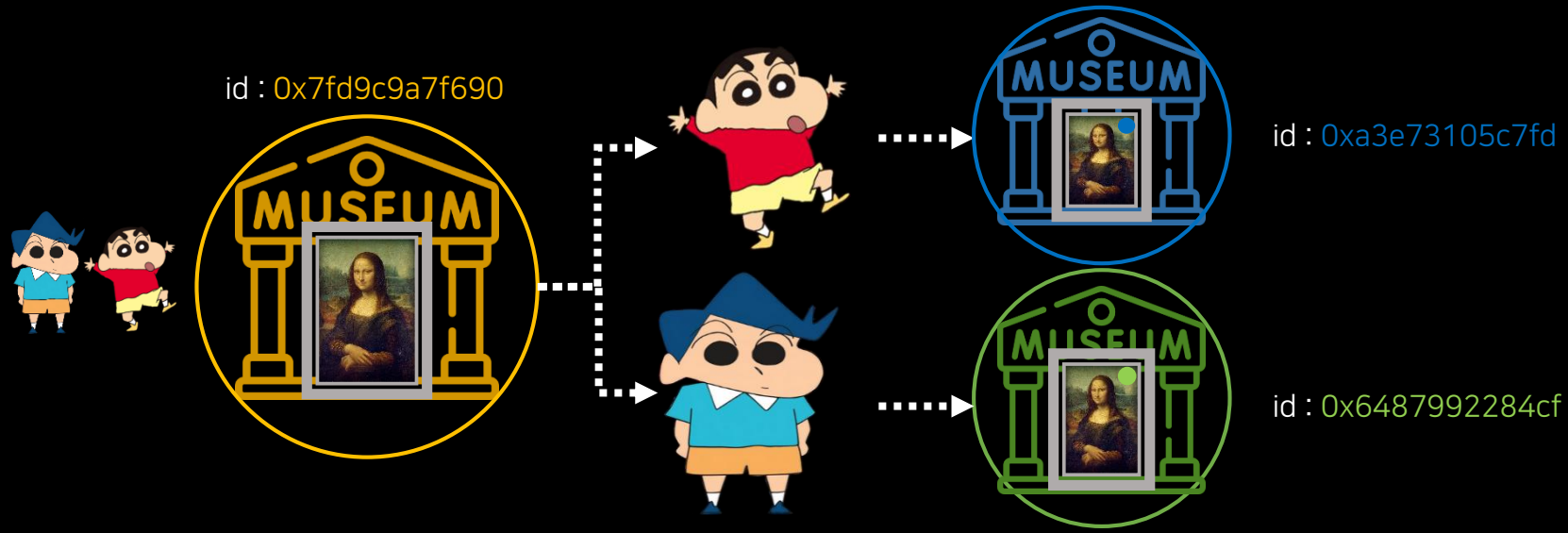
https://www.codesansar.com/python-programming/integer-interning.htm

Mutable vs Immutable

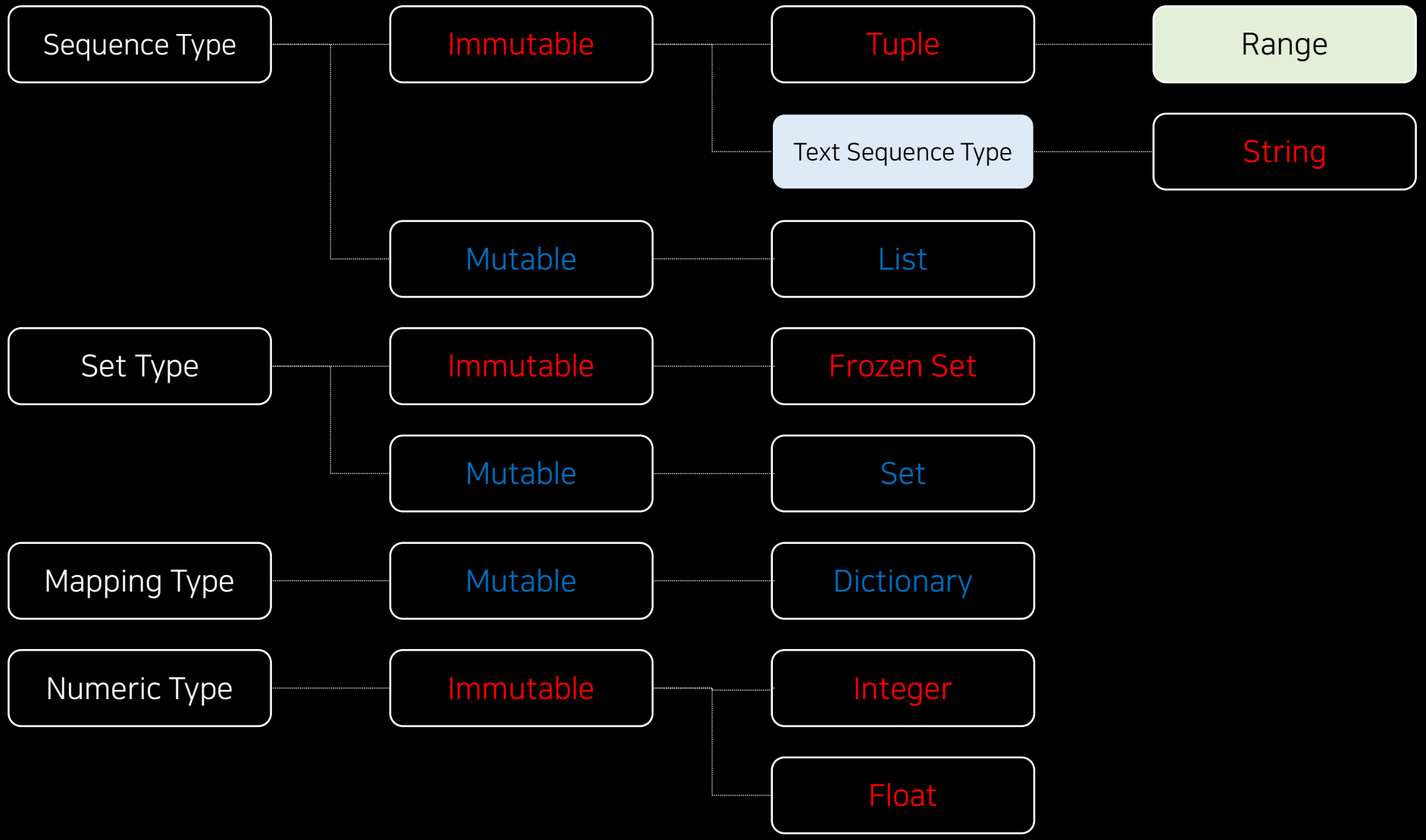
- 가변(mutable): 객체 생성 후 변경 가능



- 불변(Immutable): 객체 생성 후 변경 불가(객체를 변형하여 새로운 객체 할당)



파이썬 가변/불변 (Mutable / Immutable) 자료형



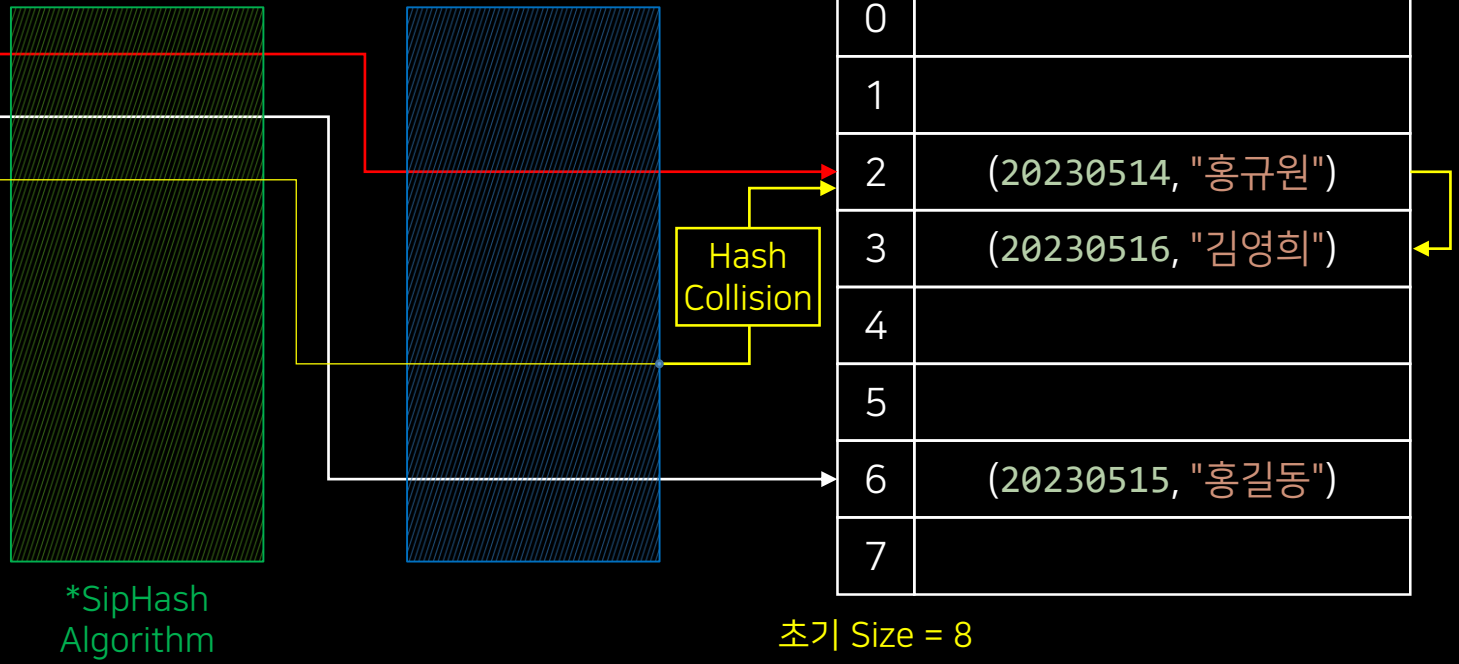
Dictionary (HashTable)

CPython

```
# empty dict
d = dict()
d[20230514] = "홍규원"
d[20230515] = "홍길동"
d[20230516] = "김영희"
```

set(insertion)
remove
search
} 평균 O(1) 시간복잡도

```
# Sudo Code

perturb =  hash(key)

i = hash(key) % size

if H[i].key == key:
  # found
else:

  While H[i] is not empty:

  i = (5 * i + 1 + perturb) % size
  perturb = perturb >> 5
```



* CPython의 기본 해시 알고리즘 : SipHash (https://peps.python.org/pep-0456/)

Everything is Object

클래스(variable), 함수(function), 리터럴(literal) 모두 객체다.

모든 클래스의 최상위 클래스는 object이다.

함수 또한 객체로 넘길 수 있다. 흔히 보는 map 함수..

Cpython에서의 Built-in Functions는 C언어로 구현이 된 것이 많기에, 코드를 직접 확인하기는 어렵지만 document를 보면 알 수 있다.

https://docs.python.org/3/library/functions.html

map(*function, iterable, *iterables*)

Return an iterator that applies *function* to every item of *iterable*, yielding the results. If additional *iterables* arguments are passed, *function* must take that many arguments and is applied to the items from all iterables in parallel. With multiple iterables, the iterator stops when the shortest iterable is exhausted. For cases where the function inputs are already arranged into argument tuples, see `itertools.starmap()`.

ndarray vs list