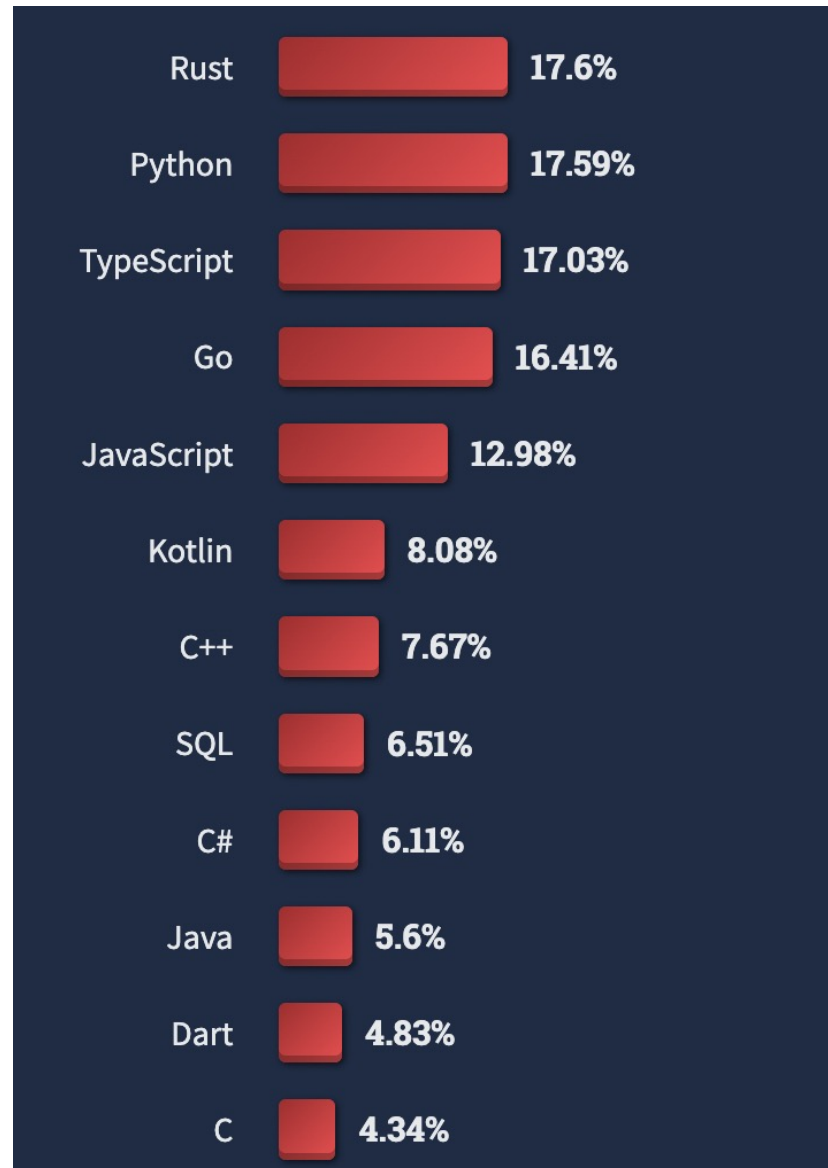


# 이상하고 아름다운 파이썬

서버 파트 코어 멤버 정명진



Worldwide, Jul 2022 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	28.38 %	-2.3 %
2		Java	17.5 %	-0.7 %
3		JavaScript	9.29 %	+0.1 %
4		C#	7.63 %	+0.5 %
5		C/C++	6.48 %	-0.1 %
6		PHP	5.32 %	-1.0 %
7		R	4.13 %	+0.4 %
8	↑↑	TypeScript	2.55 %	+0.8 %
9	↓	Objective-C	2.13 %	+0.3 %
10	↑	Swift	1.95 %	+0.3 %
11	↑↑	Go	1.7 %	+0.2 %
12		Matlab	1.67 %	+0.2 %
13	↓↓↓↓	Kotlin	1.58 %	-0.2 %

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello, World!" << endl;

    return 0;
}
```



print("Hello world")

포인터, 엄격한 자료 형, 정적 배열

# When you switch from C++ to Python



만나서 드러웠고 다신 보지 말자



이게 된다고..?

맞았습니다!!	263260 KB	2312 ms
맞았습니다!!	263260 KB	2400 ms
시간 초과		
시간 초과		
시간 초과		
시간 초과		
시간 초과		
시간 초과		
시간 초과		
시간 초과		
시간 초과		
시간 초과		
시간 초과		
시간 초과		
시간 초과		

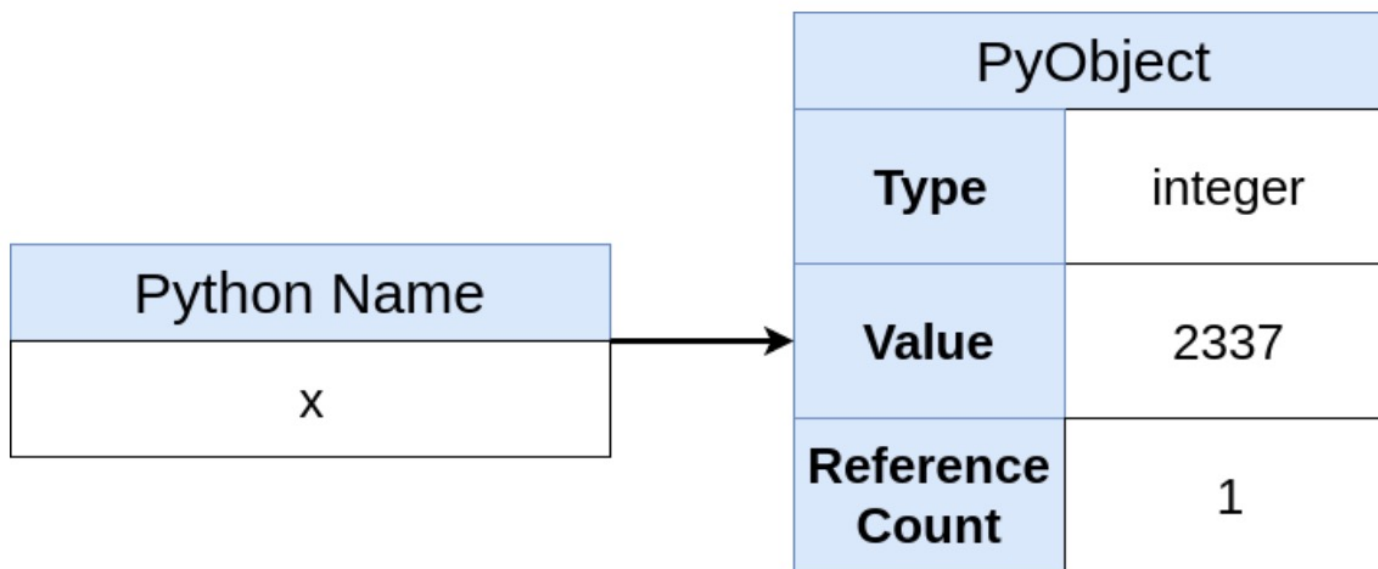
무수한 억까

파이썬은 왜 그럴까?

자료형이 자유로운 이유는 뭘까?

왜 그럴까?

# 파이썬의 모든 것은 객체이다



변수는 객체를 참조하는 별명에 불과하다



```
>>> dir(1)
['__abs__', '__add__', '__and__', '__bool__', '__ceil__', '__class__', '__delattr__', '__dir__', '__divmod__', '__doc__', '__eq__', '__float__', '__floor__', '__floordiv__', '__format__', '__ge__', '__getattr__', '__getnewargs__', '__gt__', '__hash__', '__index__', '__init__', '__init_subclass__', '__int__', '__invert__', '__le__', '__lshift__', '__lt__', '__mod__', '__mul__', '__ne__', '__neg__', '__new__', '__or__', '__pos__', '__pow__', '__radd__', '__rand__', '__rdivmod__', '__reduce__', '__reduce_ex__', '__repr__', '__rfloordiv__', '__rlshift__', '__rmod__', '__rmul__', '__ror__', '__round__', '__rpow__', '__rrshift__', '__rshift__', '__rsub__', '__rtruediv__', '__rxor__', '__setattr__', '__sizeof__', '__str__', '__sub__', '__subclasshook__', '__truediv__', '__trunc__', '__xor__', 'as_integer_ratio', 'bit_count', 'bit_length', 'conjugate', 'denominator', 'from_bytes', 'imag', 'numerator', 'real', 'to_bytes']
```

파이썬의 모든 것들은 객체로서 존재하고 다뤄진다.



샤미드



쥬피썬더



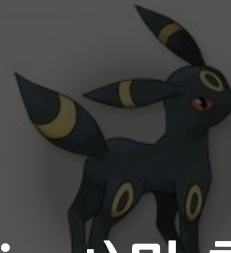
부스터



에브이



이브이



글래기



글레이시아



넴피아



리피아

파이썬의 변수는 사실 하나의 자료 형(PyObject)만 참조한다.

## 파이썬이 느린 이유

```
a = 1
```

1. 인터프리터는 1을 읽고 int 타입인 것을 유추
2. Int 에 적합한 객체 생성
3. 읽은 값 1을 생성한 객체의 값 속성에 저장
4. 생성한 객체를 변수 a가 참조할 수 있게 함

```
int a = 1
```

1. 변수 a에 int 자료형에 알맞는 메모리 영역 할당
2. 해당 메모리에 1을 저장

```
a = 1
b = 2
c = a + b
```

1. 아까의 복잡한 할당 과정이 a,b에 대해 진행
2. 변수 a가 가리키는 객체로 이동 객체의 자료형과 값 파악
3. 변수 b가 가리키는 객체로 이동 객체의 자료형과 값 파악
4. 정수 연산을 위한 `__add__()` 메서드 실행
5. 메서드 연산의 결과를 값으로 갖는 새로운 객체 생성
6. 해당 객체를 c가 참조하게 함

```
a = 1
b = 2
c = a + b
```

1. 값 할당이 a,b에 대해 진행
2. 변수 c의 메모리 영역 할당
3. 변수 a의 메모리 영역의 값 읽음
4. 변수 b의 메모리 영역의 값 읽음
5. 해당 영역에 읽은 값을 더함
6. 결과를 c의 메모리 영역에 저장

만약 이러한 더하기를 연속으로 반복하면 어떤 일이 발생할까..?

Slowest things on earth:



# 정리

"abc", [], 1



전부 객체

변수



별명, 상자 X

파이썬의 모든 것은 객체여서 자유롭지만, 때문에 느리다.



## 썬 결론

파이썬의 모든 것은 객체 여서 자유롭고 편하다.

하지만 그것이 속도 저하에 가장 큰 기여를 한다.

TMI

파이썬은 느린데 연산이 많이 들어가는 인공지능에 어떻게 사용하나요?

인공지능은 파이썬이 아닙니다. Numpy입니다.



When you replace a for loop with a vectorized numpy function and see the speed improvement



대규모 서비스에서는 활용 안하나요?



Spotify®

**Q&A**