

# Project #2

Survey on  
Programming  
Paradigms



스크립트언어의 급격한 성장 !  
“파이썬” 쉽고 강력한 언어의 등장

# Project #2

Survey on  
Programming  
Paradigms



발표를 시작하기 전에..

# Index

---

## ▶ 개요 – 선정된 주제의 배경 및 특징

- 주제선정의 배경
- 스크립트 언어란 ?
- 스크립트 언어의 폭발적인 성장 Why?
- 스크립트 언어의 사용사례

## ▶ 본론 – 구체적인 내용 및 소스코드예제(언어특징)

- “파이썬” 넌 누구야?!

## ▶ 평가 – 언어 평가

## ▶ 결론 – 정리 및 개인 소견

## ▶ Q&A

# 주제 선정의 배경

---

- ▶ Powerful Compile Language
  - C/C++ , JAVA, C#, Objective - C, ... etc.
- ▶ Spatial expansion of the system. (시스템의 공간 확장)
  - Desktop, mobile ~ IoT with new Web Technology
- ▶ Needs
  - Dynamic Web, Script Engine, Real Time
- ▶ More powerful Web Technology
  - Web 3.0 - 4.0 age.

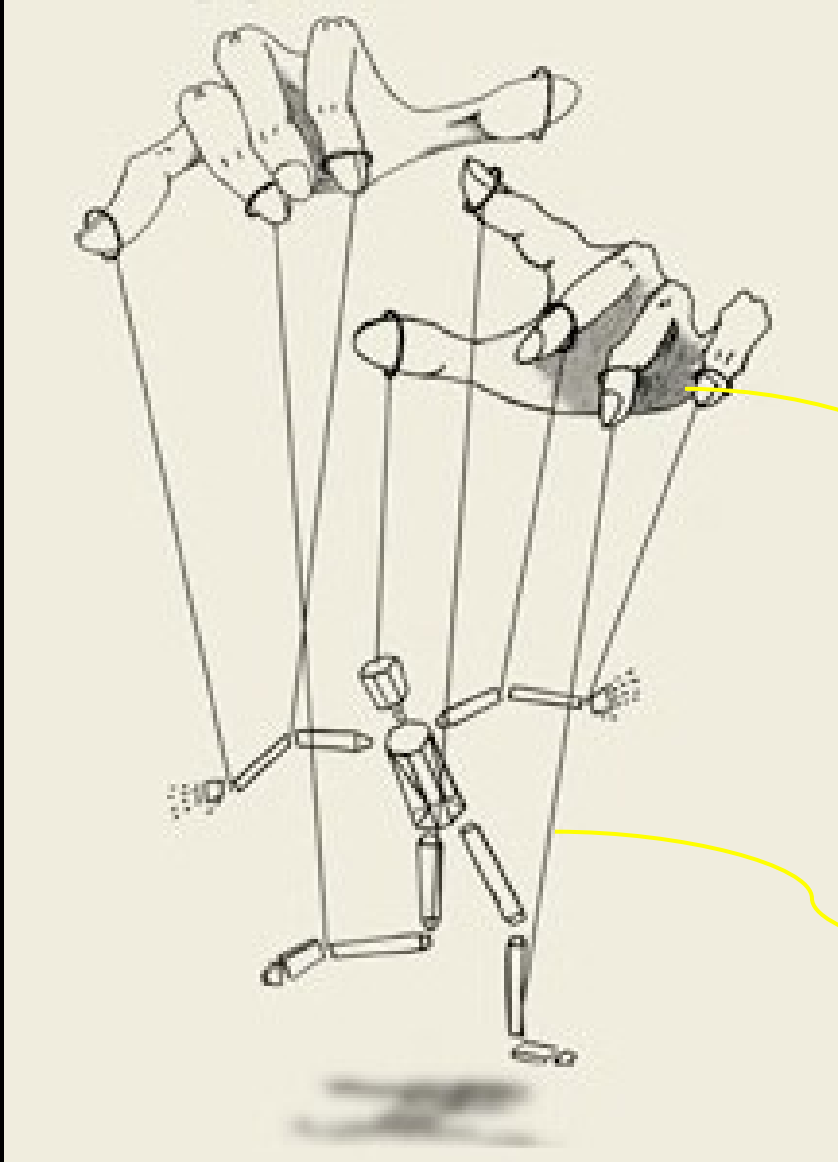
# Project #2

Survey on  
Programming  
Paradigms



스크립트 언어란 ?

# 스크립트 언어란?



- ▶ 응용 소프트웨어와 독립하여 사용됨.
- ▶ 응용 소프트웨어를 제어하는 언어.

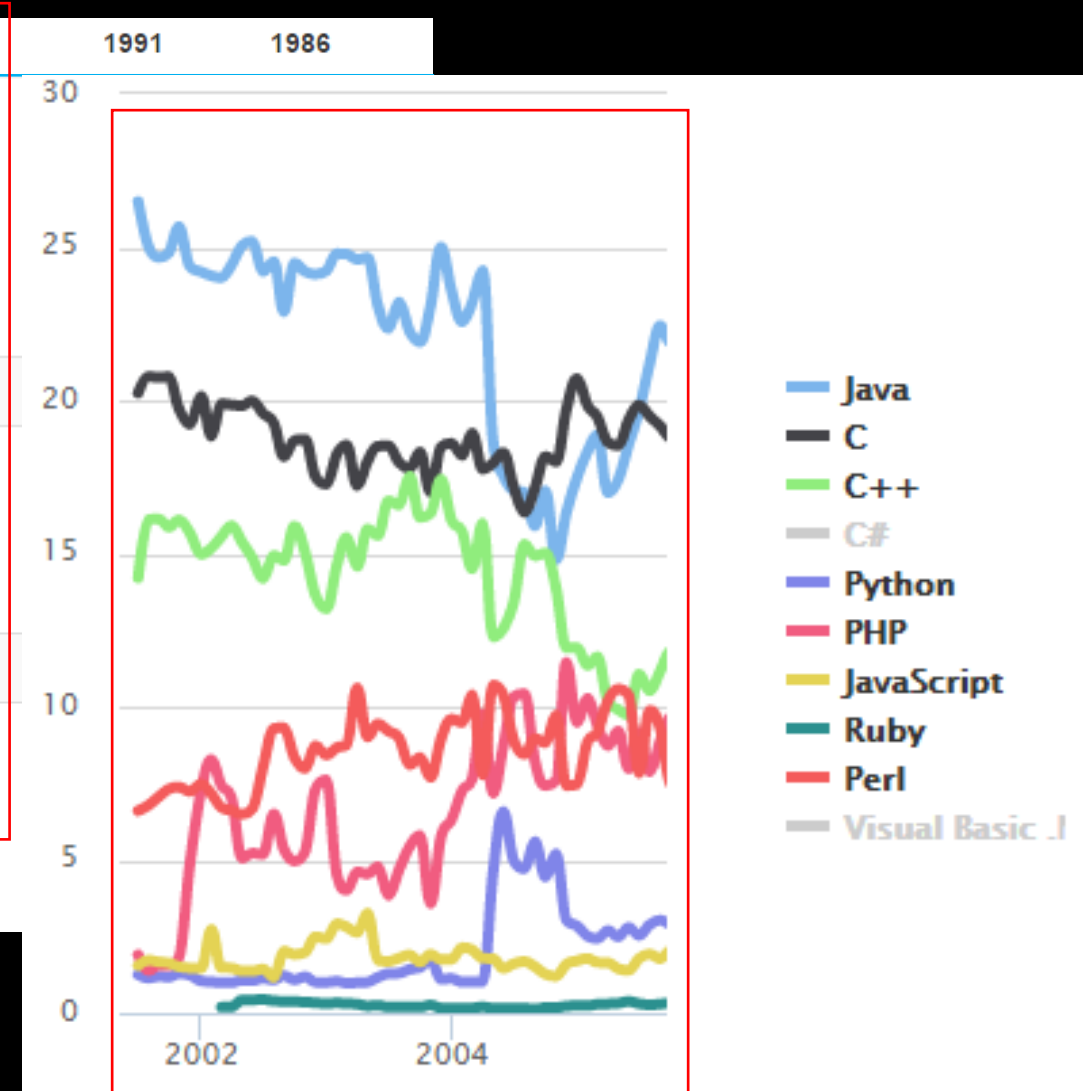
스크립트 언어

응용 소프트웨어

# 스크립트 언어의 폭발적인 성장 Why?

Programming Language	2016	2011	2006	2001	1996
Python	5	6	7	26	18
PHP	6	4	4	10	-
JavaScript	8	9	8	9	31
Perl	9	8	5	4	3
Ruby	10	10	24	31	-

A red arrow points from the Ruby row in the table to the line graph on the right. A blue arrow points from the JavaScript row in the table to the line graph on the right.



# 스크립트 언어의 폭발적인 성장 Why?





# Project #2

Survey on  
Programming  
Paradigms



스크립트 언어 (파이썬)의  
사용 사례

# 스크립트 언어의 (파이썬) 사용사례

## ▶ Server Side – backend

- 트위터, gmail, 스냅샷

## ▶ 스크래핑, 크롤링

- 스카이스캐너

## ▶ DB 프로그래밍

- 인터넷쇼핑몰 DB연결

## ▶ 웹 프레임워크

- 홈페이지 제작

## ▶ 소켓 프로그래밍, API

- 웹 어플리케이션, API서버, 카카오톡

## ▶ 문자열 처리, 수치 연산

- 빅 데이터처리, 기계학습

## ▶ 시스템유틸리티 제작

- 해킹 프로그램

## ▶ 각종 언어와 결합

- C/C++ , JAVA, PHP, ruby, Ada, ... 등

# 별첨 - 파이썬으로 할 수 있는 일



# 별첨 - 파이썬으로 할 수 없는 일

---

- ▶ 파이썬으로 도스나 리눅스 같은 운영체제, 엄청난 횟수의 반복과 연산을 필요로 하는 프로그램 또는 데이터 압축 알고리즘 개발 프로그램등을 만들기에는 부적합하다.
- ▶ 즉, 대단히 빠른 속도를 요구하거나 하드웨어를 직접 핸들링 하는 프로그램에는 어울리지 않는다.

# Project #2

Survey on  
Programming  
Paradigms



“파이썬” 넌 누구야 ?!

“파이썬” 넌 누구야?!



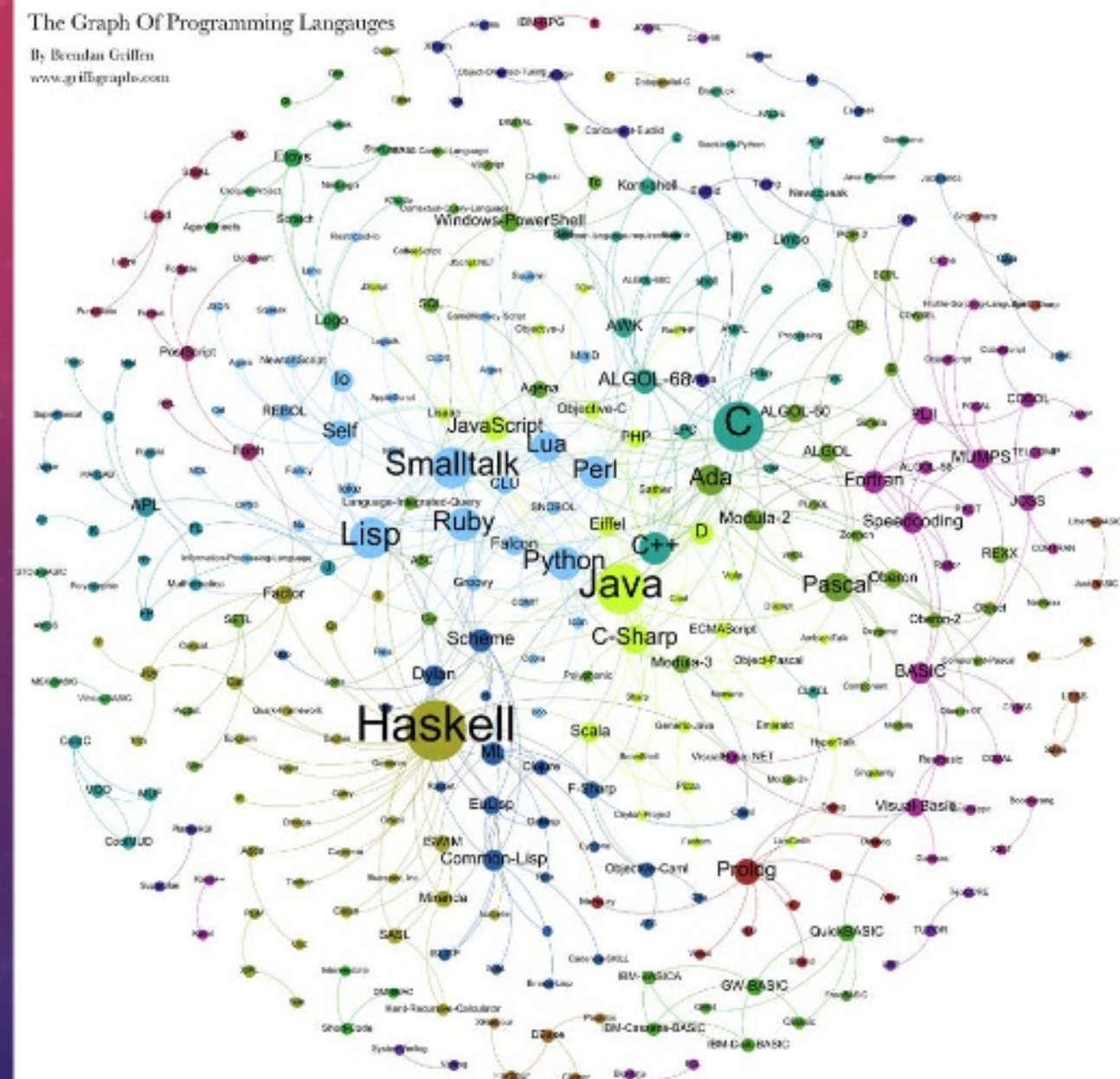
python™





# The Graph Of Programming Languages

By Brendan Grifflin  
www.grifflin.com



# “파이썬” 패러다임



- ▶ 함수형
- ▶ 명령형
- ▶ 객체지향형



# “파이썬” 함수형 프로그래밍

---

$Y = f(x) * f(x);$

$z = f(x);$

$Y = z * z;$

# “파이썬” 함수형 프로그래밍

특징	명령형 방법	함수형 방법
프로그래머가 중점을 두는 부분	작업을 수행하는 방법(알고리즘)과 상태의 변경을 추적하는 방법	원하는 정보와 필요한 변환
상태 변경	중요	존재하지 않음
실행 순서	중요	중요도가 낮음
주요 흐름 제어	루프, 조건 및 함수(메서드) 호출	재귀를 비롯한 함수 호출
주요 조작 단위	클래스나 구조체의 인스턴스	1급(first-class) 개체와 데이터 컬렉션인 함수

# “파이썬” 특징



- ▶ Dynamic
- ▶ Interpreted
- ▶ Multi Paradigm
- ▶ Cross Platform
- ▶ Readability
- ▶ Unicode
- ▶ Free
- ▶ Library
- ▶ Glue-tic
- ▶ Writability

# “파이썬” 특징 - Unicode

```
이름 = "어쩌고"
```

```
나이 = 17
```

```
보여줘 = print
```

```
보여줘('{} {}'.format(이름, 나이))
```

# “파이썬” 특징 - Writeability

```
#include <iostream>
using namespace std;
int main(int argc, const char * argv[])
{
    cout << "Hello World!" << endl;
    return 0;
}
```

```
>>> print ("Hello, World!") # 3.X
Hello, Wolrd!
```

# “파이썬” 특징 - Readability

```
int factorial(int x) {  
    if(x == 0) {return 1;} else  
    {return x * factorial(x - 1); } }
```

```
def factorial(x):  
    if x == 0:  
        return 1  
    else:  
        return x * factorial(x - 1)
```

# “파이썬” 특징 – 동적 타이핑



```
>>> from types import *
>>>
>>> def what (x):
...     if type(x) == IntType:
...         print "This is an int."
...     else:
...         print "This is something else."
...
>>> what(4)
This is an int.
>>>
>>> what("4")
This is something else.
```

# “파이썬” 특징 - Interpreted

---

```
>>>a=4 # 변수
```

```
>>>b=3
```

```
>>>a+b
```

```
7
```



# “파이썬” 특징 – Glue-tic

```
#include <python.h>

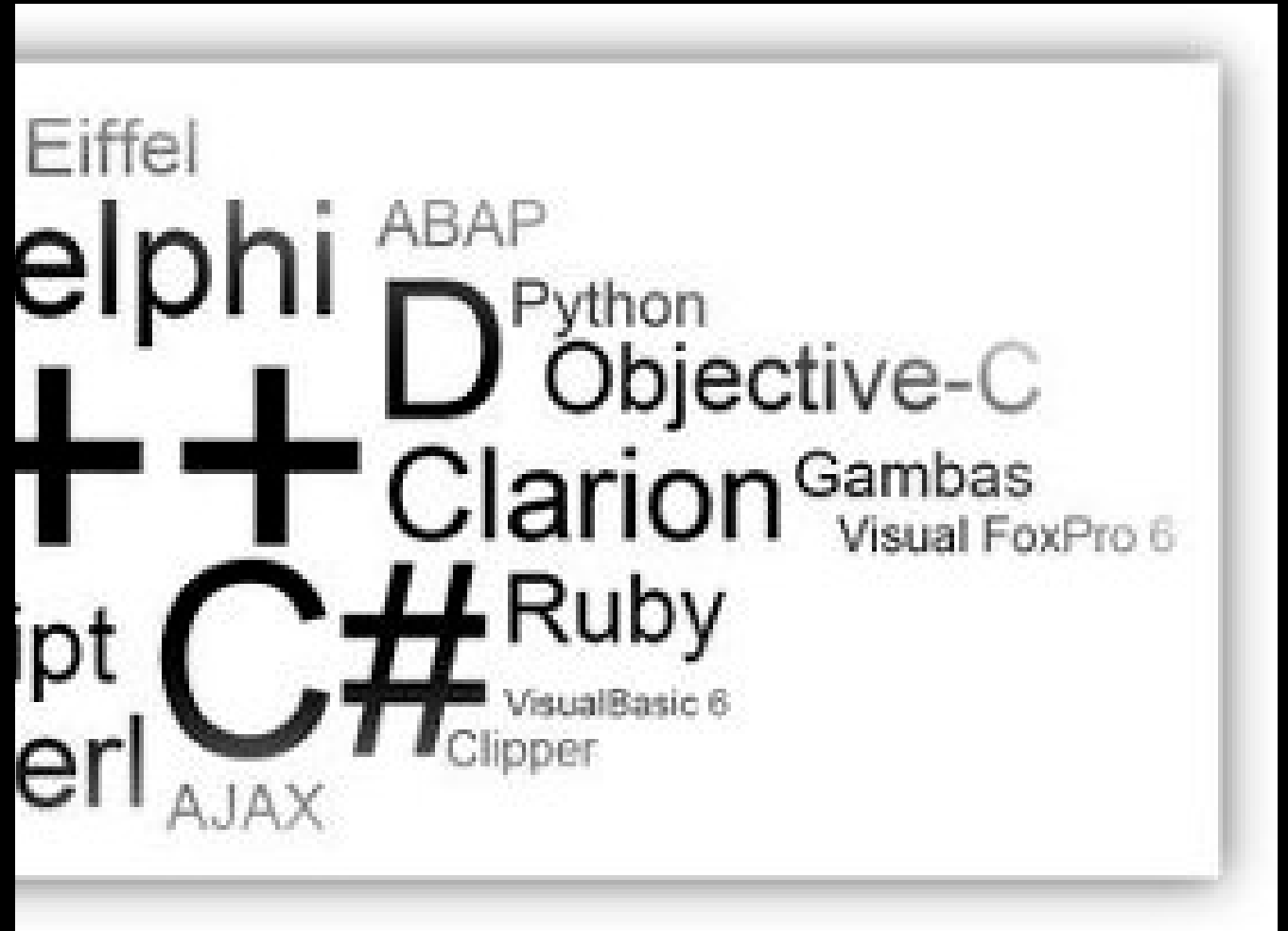
static PyObject *

spam_strlen(PyObject *self, PyObject *args)
{
    char* str;
    int len;

    if(!PyArg_ParseTuple(args, "s", &str))
        return NULL;

    len = strlen(str);

    return Py_BuildValue("i", len);
}
```



# “파이썬” 특징 – 풍부한 library



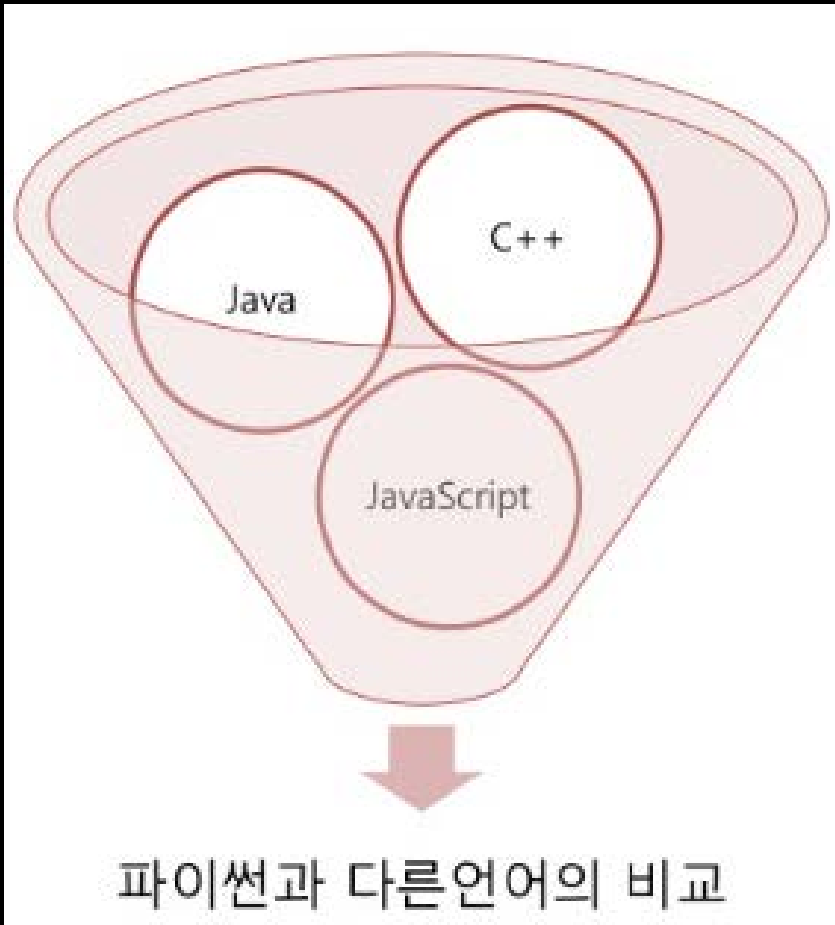
# Project #2

Survey on  
Programming  
Paradigms



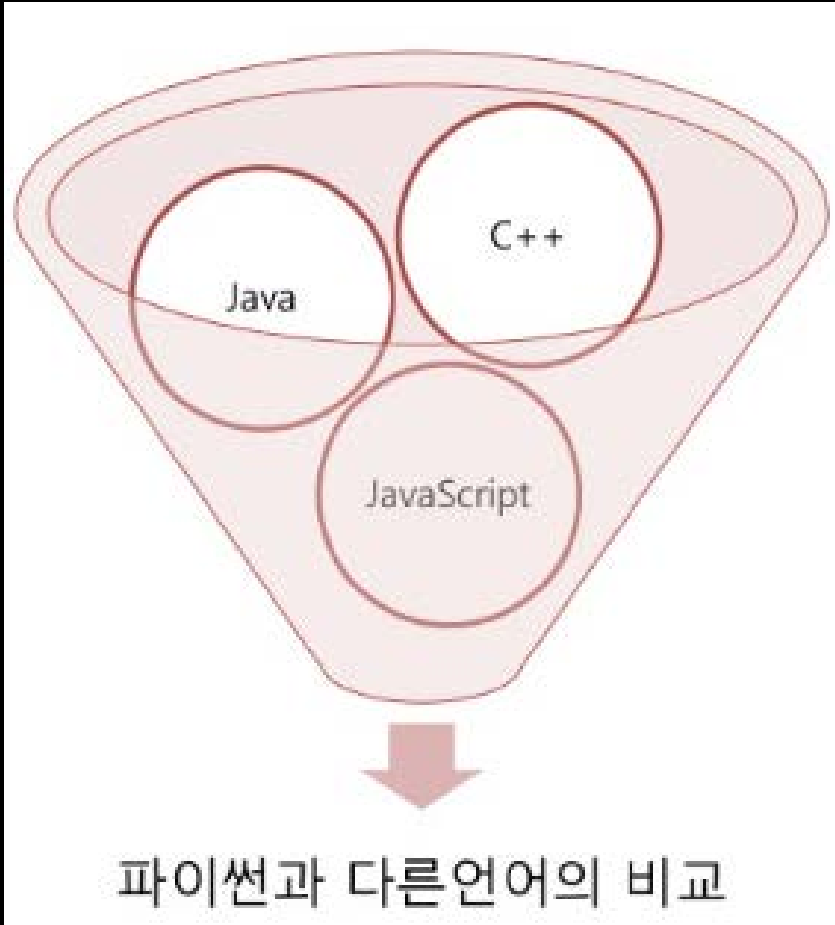
“파이썬” 타 언어와의 비교

# “파이썬” 언어 비교



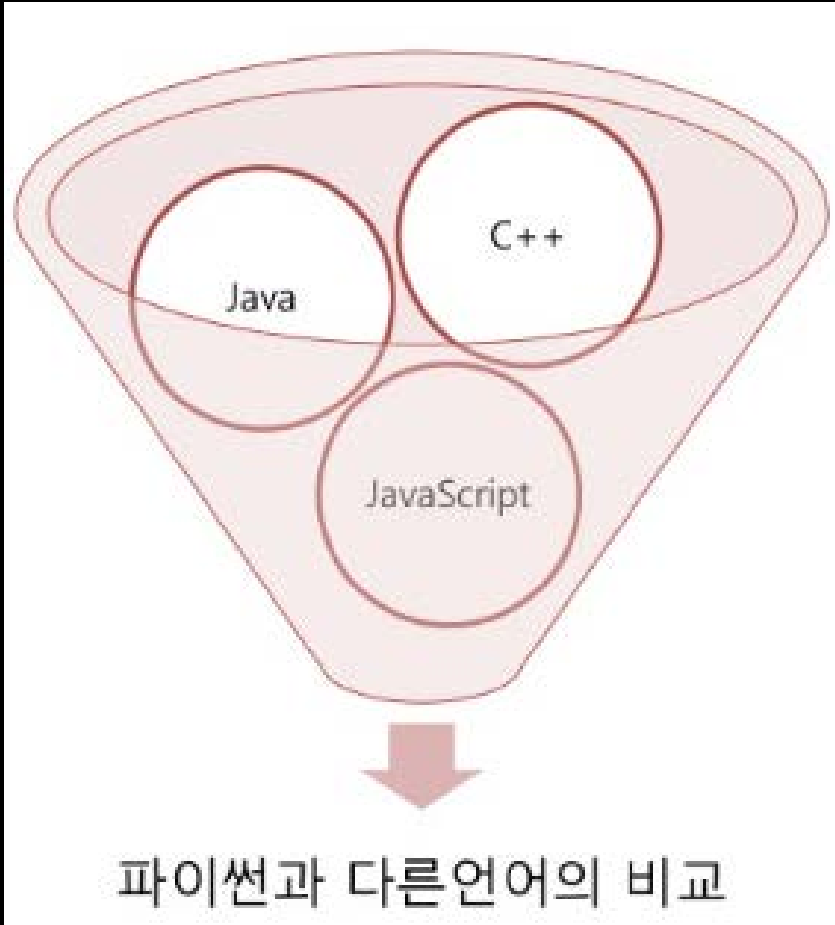
비교 언어	설명
Java	<p>일반적으로 Python 프로그램은 Java 프로그램 보다는 느리게 수행된다. 하지만 Python은 개발하는 시간이 훨씬 적게 걸린다.</p> <p>Python 은 Java 보다도 3~5배 정도 코드가 짧으며, 이 차이는 Python의 내장 고수준 데이터 형과 동적인 형결정 기능에서 기인한다고 생각한다. 같은 실행시간에 좀더 많은 일은 한다는 뜻이다.</p> <p>이러한 이유들로, Python은 '접착' 언어로서 아주 적당한 반면, Java는 저수준 구현 언어로 특성화 지을 수 있다. Python이 Java로 구현 되기 전에 그 프로토 타입을 정하는데 활용된다.</p>

# “파이썬” 언어 비교



비교 언어	설명
JavaScript	<p>Python 의 '객체 기반' 부분의 집합이 대략 JavaScript와 동일하다.</p> <p>JavaScript와 같이 (그러나 Java와는 다르게), Python은 클래스 안에 정의하지 않아도 되는, 단순한 함수와 변수를 사용하는 프로그래밍 스타일을 지원한다. 하지만, JavaScript는 이것이 지원하는 전부이다.</p> <p>반면에, Python은 클래스와 상속이 중요한 역할을 하는 진정한 객체 지향프로그래밍 스타일을 통하여 더 좋은 코드 재사용을 하도록 지원한다.</p>

# “파이썬” 언어 비교



비교 언어	설명
C++	<p>Java에 대해서 이야기한 대부분이 c++에도 적용된다. Python코드가 Java코드보다 3-5배 짧으며, C++코드에 비해 5-10배 짧다 !!</p> <p>일 예로 C++프로그래머 두명이 1년에 끝낼 수 없는 일을 두 달만에 끝낸 일도 있었다고 한다.</p> <p>Python은 C++로 쓰여진 코드를 사용하는 접착언어로 빛을 발한다.</p>

# Project #2

Survey on  
Programming  
Paradigms



“파이썬” 언어 평가

# “파이썬” 언어 평가

---

- ▶ Readability : 쓰는 사람에 관계 없이 통일성을 유지하게 하며, 코드가 간결하고 이해하기 쉬움.
- ▶ Writability : 코드가 간결하고 사용하기 쉽다는 이유로, 많은 기업들이 파이썬을 사용하고 분석한다. 쉽게 배울 수 있고 생산성이 좋아, 프로토 타입 개발에도 많이 사용됨.



# “파이썬” 언어 평가

---

- ▶ Reliability : 파이썬은 인터프리터를 통해 실행시간에 자료형을 실시간으로 검사하고 정의합니다.
- ▶ Cost : 전반적으로 가독성과 생산성이 좋아 유지보수가 쉽고, 빠르게 배울 수 있기 때문에, 프로토타입 제작에도 유용하다.

# “파이썬” 언어 평가

---

- ▶ Performance: 인터프리터형 언어 특징상 강력한 퍼포먼스를 발휘 하진 않지만, 퍼포먼스가 강력한 언어들 사이에서 “접착 언어”로 성능을 보완할 수 있음.
- ▶ 보안: 지난 수년 동안 Java에서만 수백 개의 취약성이 발견됐었지만, 파이썬에서 발견된 보안 취약성은 소수에 불과하다.

# Project #2

Survey on  
Programming  
Paradigms



결론 및 개인 소견

# “파이썬” 결론 – 정리 및 개인 소견

---

- ▶ 파이썬은 초보자부터 전문가까지 다양한 층을 보유하고 있는 언어이다. (쉽고 강력하다.)
- ▶ 그 자체로는 보완해야할 점이 있지만, 가독성, 생산성, 유지보수, 총 비용, 퍼포먼스, 보안 부분에서 상당히 좋은 점수를 받고 있기 때문에 프로토타입과 “접착 언어”로 사용하기에 적합하다.

# “파이썬” 결론 - 정리 및 개인 소견

---

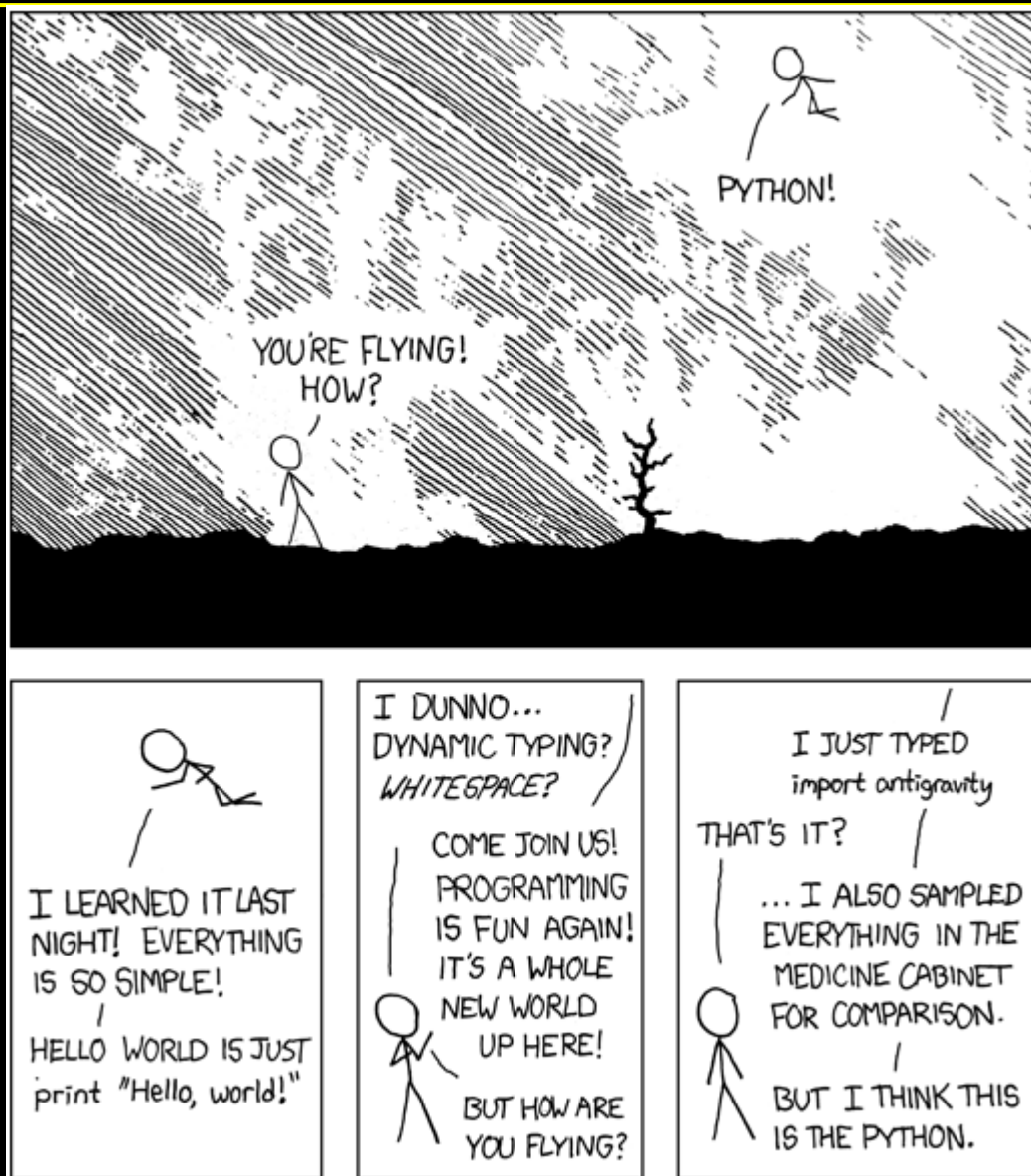
▶ 처음 프로그래밍을 배우거나 첫 번째 server-side 언어, 프로토 타입을 목적으로 한 언어, “접착 언어”로 사용하기에 정말 괜찮은 언어라고 생각된다. 아쉬운 점은, 프로그래밍적 사고를 키우기에는 너무 함축된 프로그래밍 방식을 지원하기 때문에 프로그래밍을 전문적으로 공부하기에는 메인 언어로 다루기에는 부족하다고 생각한다.

# Project #2

Survey on  
Programming  
Paradigms



# 한 장 요약



Q&A

---

Q&A



# 출처

---

1. [https://ko.wikipedia.org/wiki/함수형\\_프로그래밍](https://ko.wikipedia.org/wiki/함수형_프로그래밍)
2. <https://ko.wikipedia.org/wiki/파이썬>
3. [https://ko.wikipedia.org/wiki/스크립트\\_언어](https://ko.wikipedia.org/wiki/스크립트_언어)
4. [https://ko.wikipedia.org/wiki/다중\\_패러다임\\_프로그래밍\\_언어](https://ko.wikipedia.org/wiki/다중_패러다임_프로그래밍_언어)
5. [http://www.tiobe.com/tiobe\\_index?page=Python](http://www.tiobe.com/tiobe_index?page=Python)
6. 파이썬3 바이블
7. JavaScript: The Definitive Guide 5/E