



진정원 [WWW.WEBOOSTER.CO.KR](http://WWW.WEBOOSTER.CO.KR)

---

# JAVASCRIPT

## 진정원

- ▶ embedded / mobile / server / db / web
- ▶ [webooster.co.kr](http://webooster.co.kr)
- ▶ Meteor.js School / Startup
- ▶ 010-6617-6309 / [ceo@webooster.co.kr](mailto:ceo@webooster.co.kr)

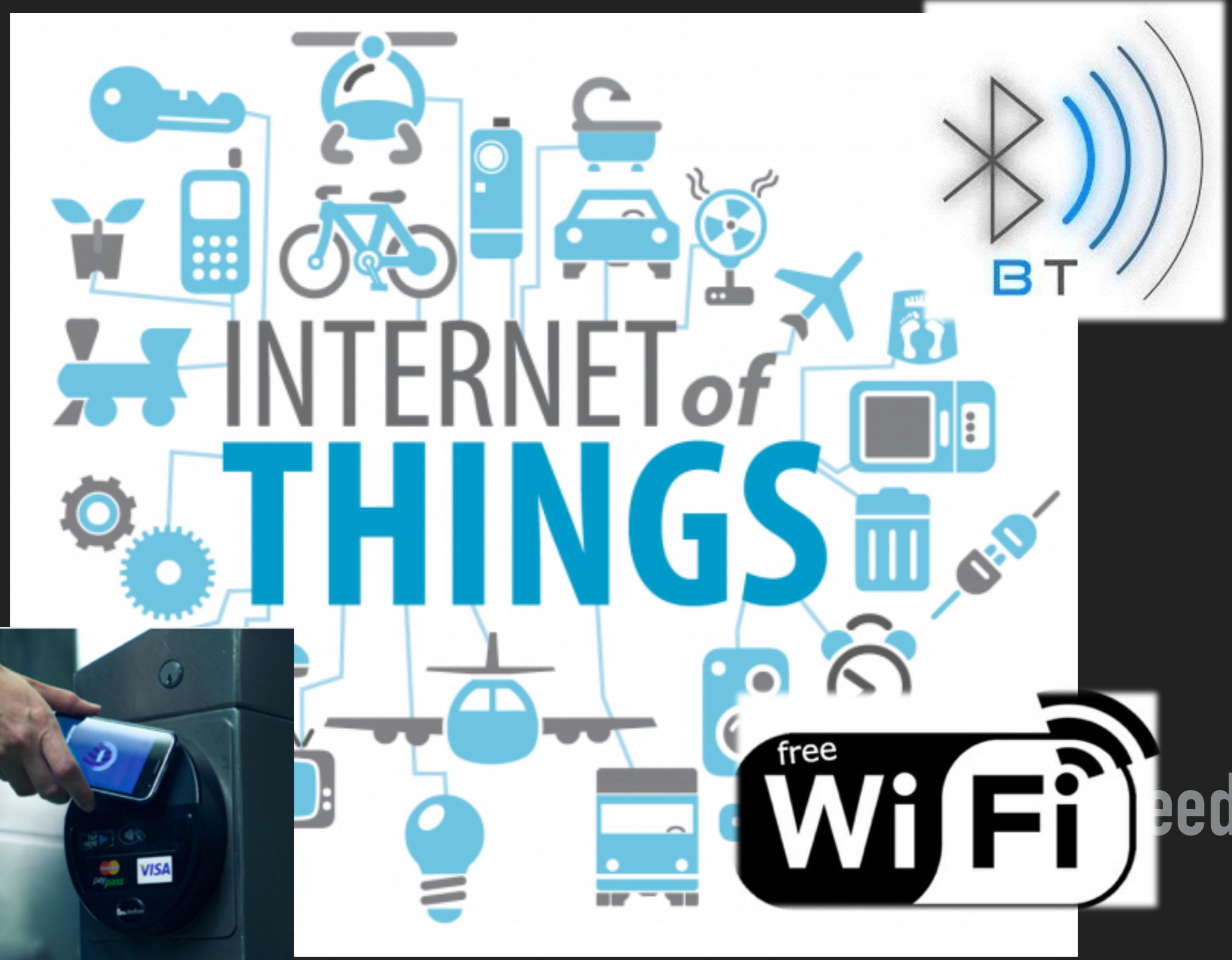
# TERMS



facebook

facebook

facebook map



## IPV4 VS IPV6

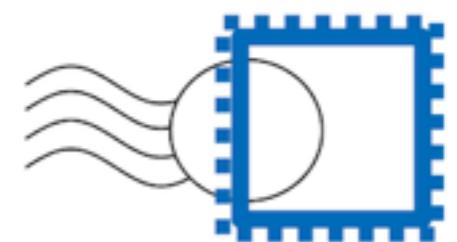
An IPv6 address

(in hexadecimal)

2001:0DB8:AC10:FE01:0000:0000:0000:0000

↓      ↓      ↓      ↓      [ ]  
2001:0DB8:AC10:FE01:: Zeroes can be omitted

00100000000001:0000110110111000:1010110000010000:1111111000000001:  
0000000000000000:0000000000000000:0000000000000000:0000000000000000



My IP Address  
**192.168.1.1**

<https://en.wikipedia.org/wiki/IPv6>

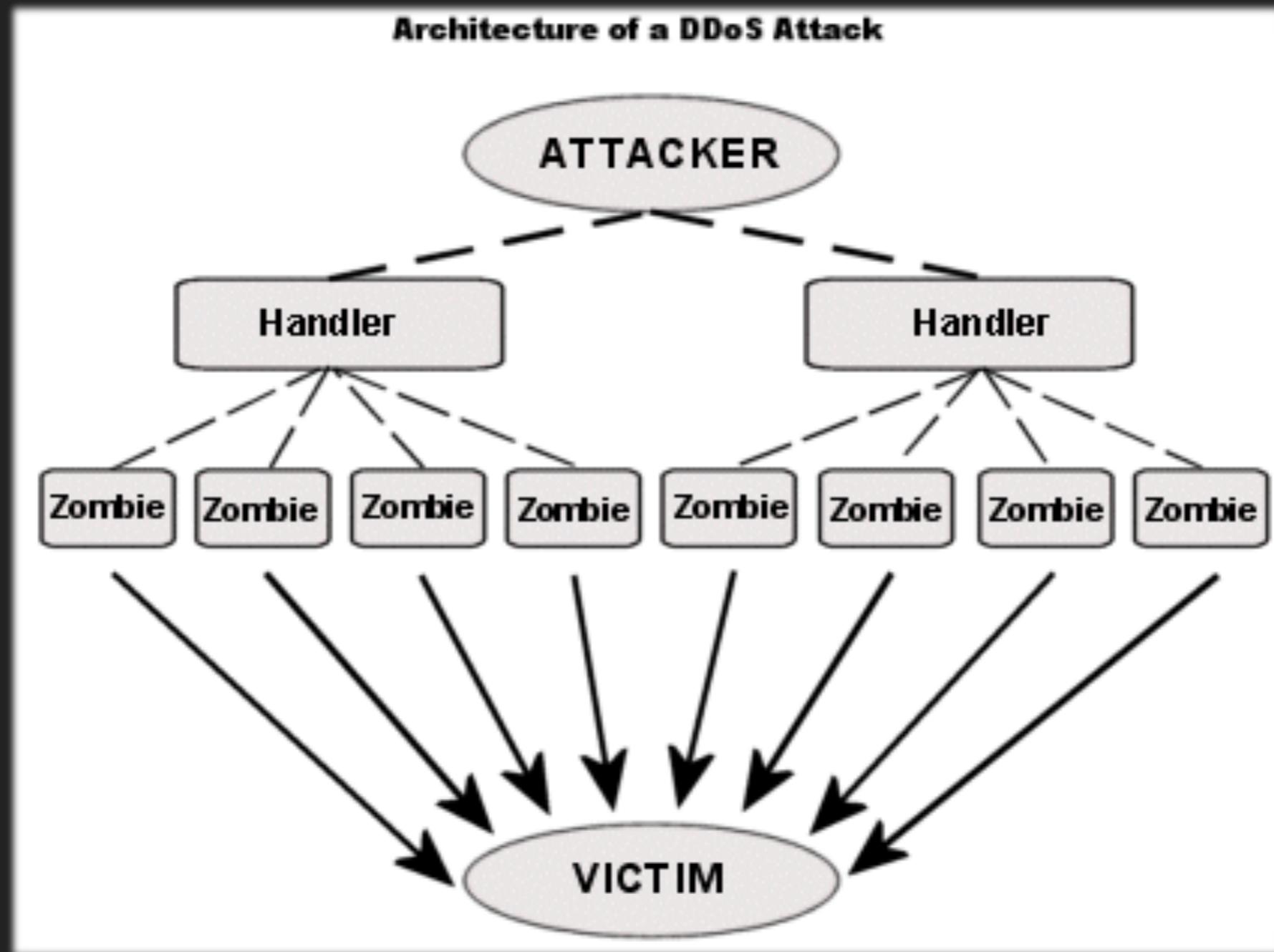
I'M BOT!



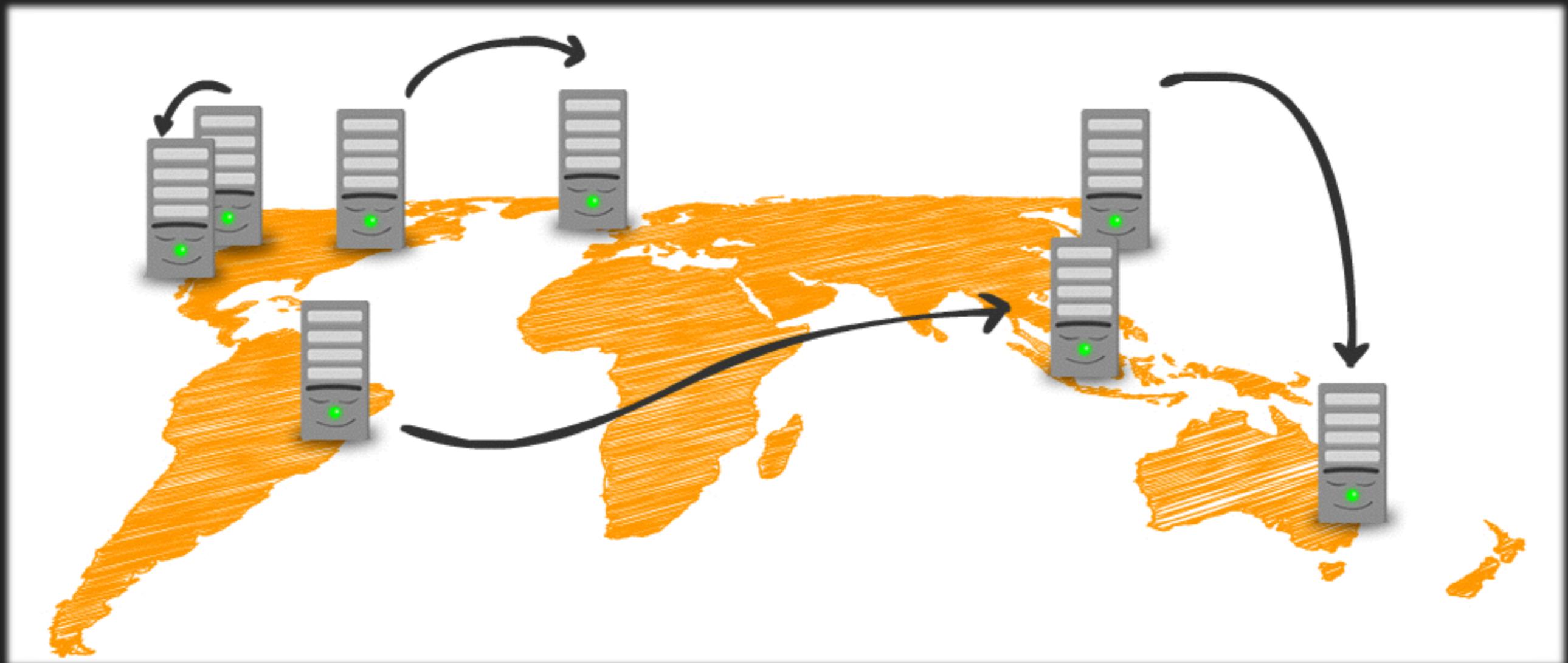
<http://www.filothaea.com/blog/google-tips-how-to-call-googlebot-to-index-your-new-website-blog-quickly/>

# DDOS ATTACKING

---



Johnny Appleseed



<http://blog.takipi.com/aws-olympics-speed-testing-amazon-ec2-s3-across-regions/>

## WINDOWS VS MAC

---



first item of google search: windows vs mac

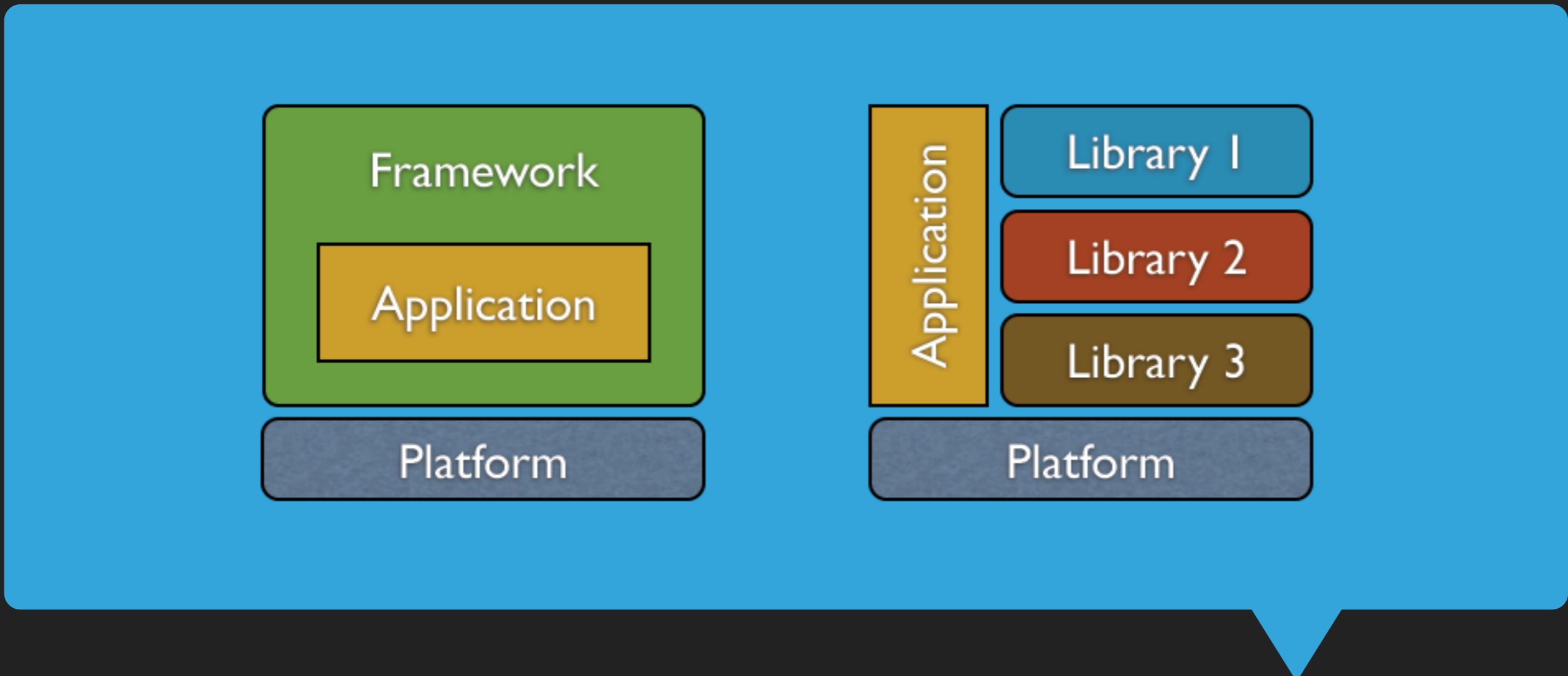
## EXPLORER & ACTIVE-X & 공인인증서



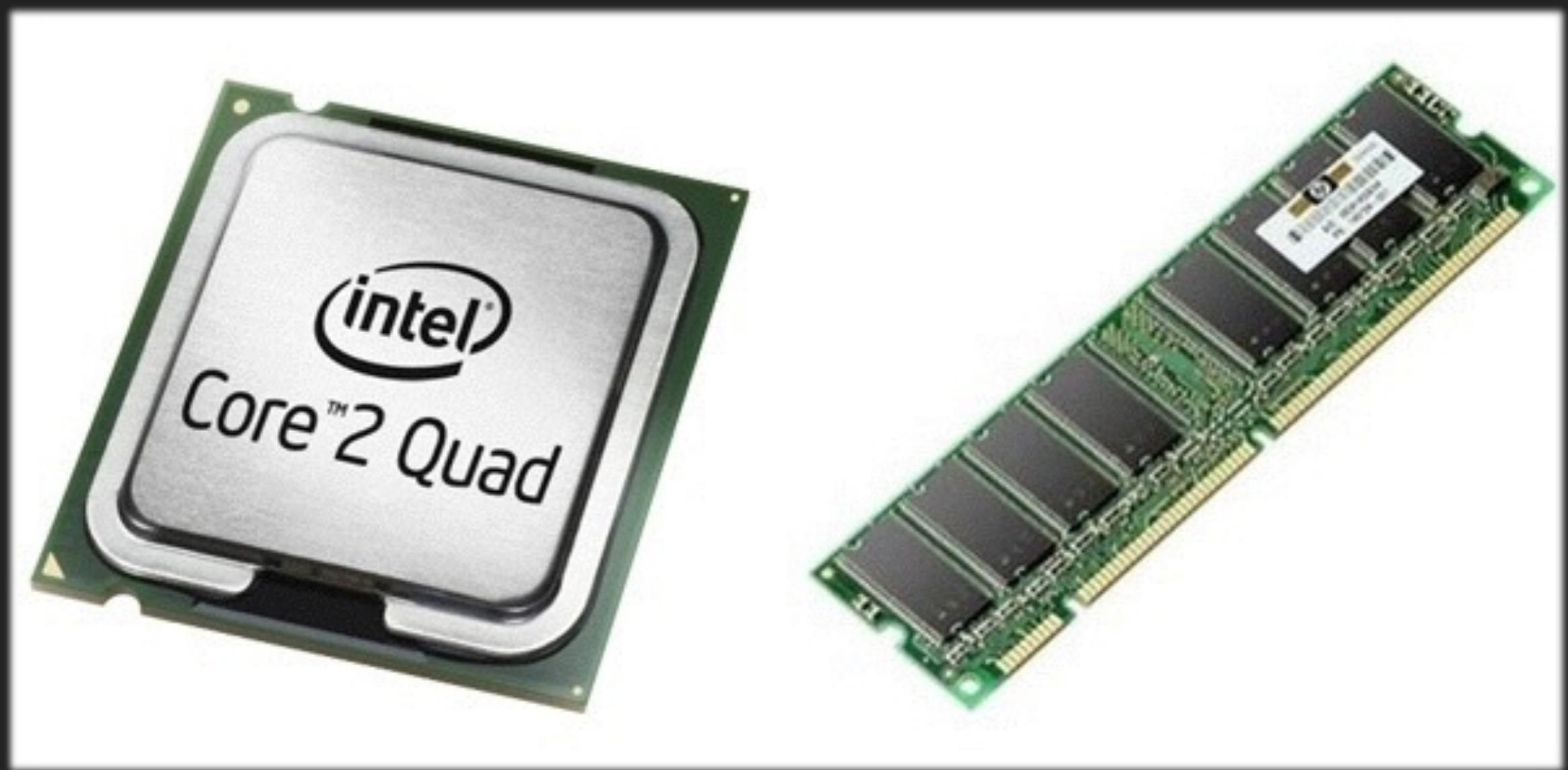
<http://uncyclopedia.kr/wiki/%EC%9C%A0%EC%8B%9D>  
<http://times.kaist.ac.kr/news/articleView.html?idxno=2628>

## LIBRARY & FRAMEWORK & PLATFORM

---



<http://stackoverflow.com/a/30744433/2762215>

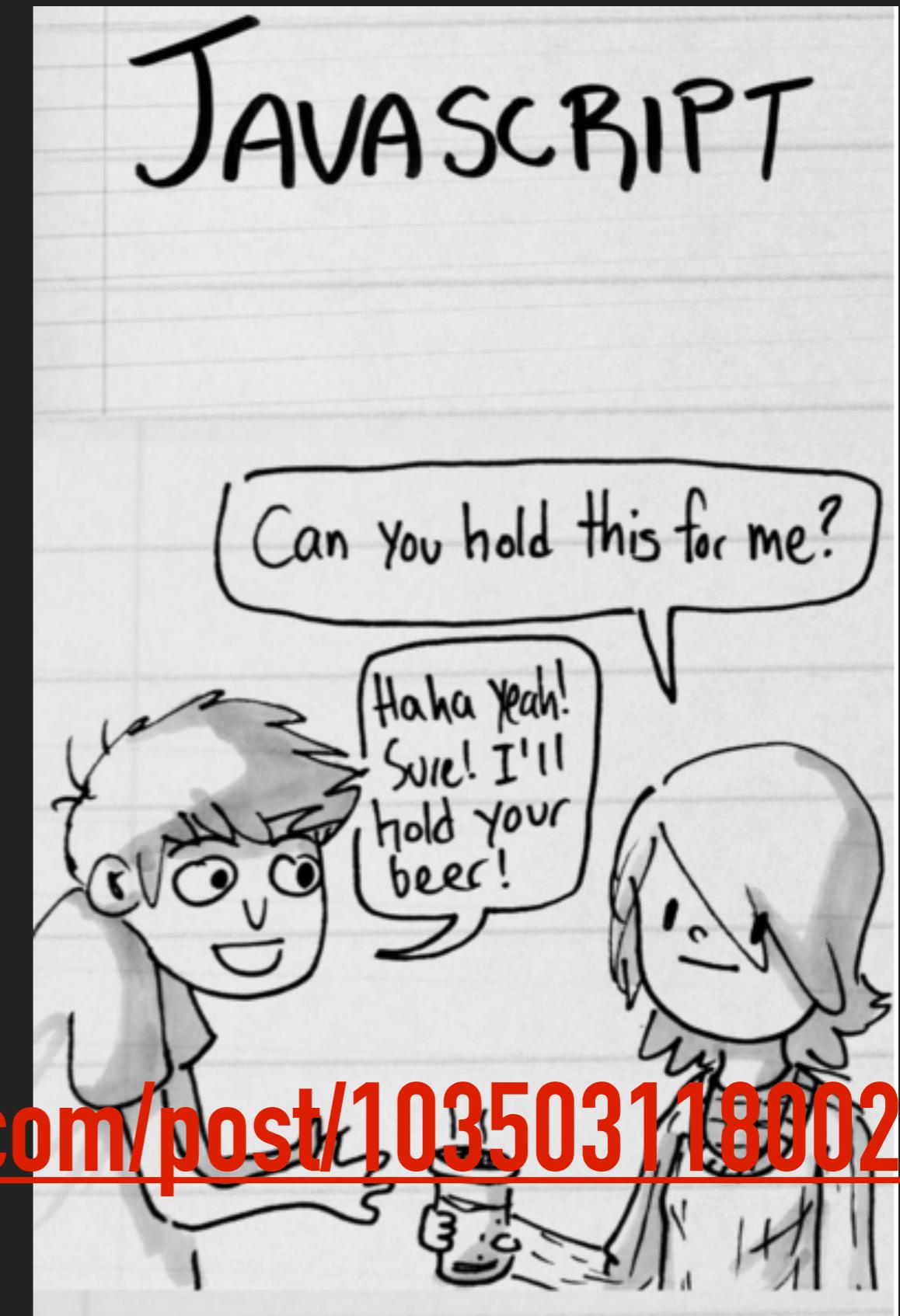
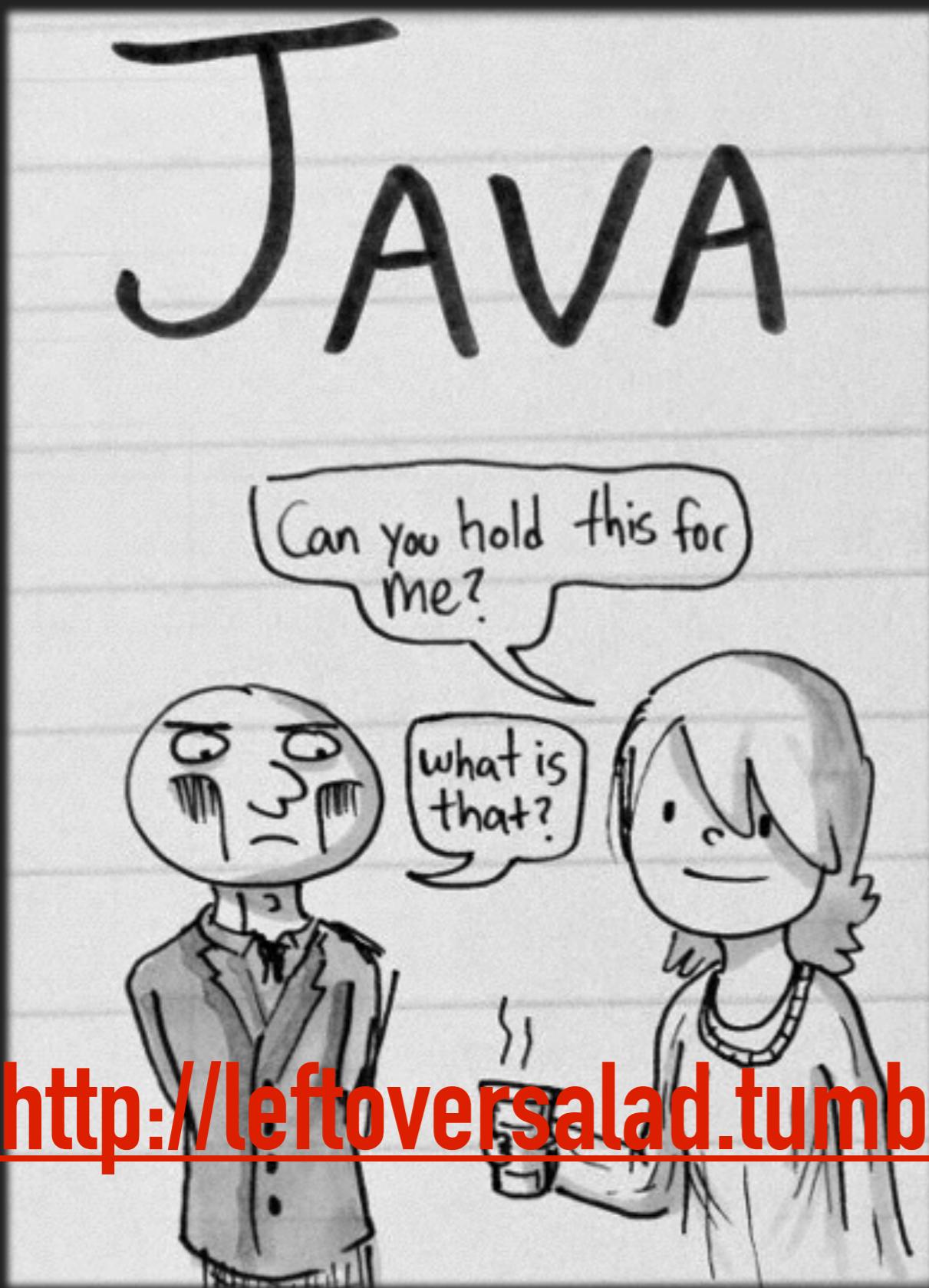


[http://vsphere-land.com/news/understanding-cpu-memory-  
management-in-vsphere.html](http://vsphere-land.com/news/understanding-cpu-memory-management-in-vsphere.html)

# PROGRAMMING LANGUAGES



<http://blog.gaerae.com/2015/06/website-run-execute-code-online.html>



<http://leftoversalad.tumblr.com/post/103503118002>

# COFFEESCRIPT

```
class Animal
  constructor: (@name) ->
    move: (meters) ->
      alert @name + " moved #{meters}m."
    class Snake extends Animal
      move: ->
        alert "Slithering..."
        super 5
    class Horse extends Animal
      move: ->
        alert "Galloping..."
        super 45
    sam = new Snake "Sammy the Python"
    tom = new Horse "Tommy the Palomino"
    sam.move()
    tom.move()
```

```
var Animal, Horse, Snake, sam, tom,
  __hasProp = {}.hasOwnProperty,
  __extends = function(child, parent) { for (var key in parent) { if (__hasProp.call(parent, key)) child[key] = parent[key]; } }
  function ctor() { this.constructor = child; } ctor.prototype =
  parent.prototype; child.prototype = new ctor(); child.__super__ =
  parent.prototype; return child; }

Animal = (function() {
  function Animal(name) {
    this.name = name;
  }

  Animal.prototype.move = function(meters) {
    return alert(this.name + (" moved " + meters + "m."));
  };

  return Animal;
})();

Snake = (function(_super) {
  __extends(Snake, _super);
  function Snake() {
    return _super.__super__.constructor.apply(this, arguments);
  }

  Snake.prototype.move = function() {
    alert("Slithering...");
    return Snake.__super__.move.call(this, 5);
  };

  return Snake;
})();

Horse = (function(_super) {
  __extends(Horse, _super);

  function Horse() {
    return Horse.__super__.constructor.apply(this, arguments);
  }

  return Horse;
})()



GAHHHH! MY EYES!



CONTINUES FOR ANOTHER PAGE...


```

<http://blog.gruffdavies.com/>



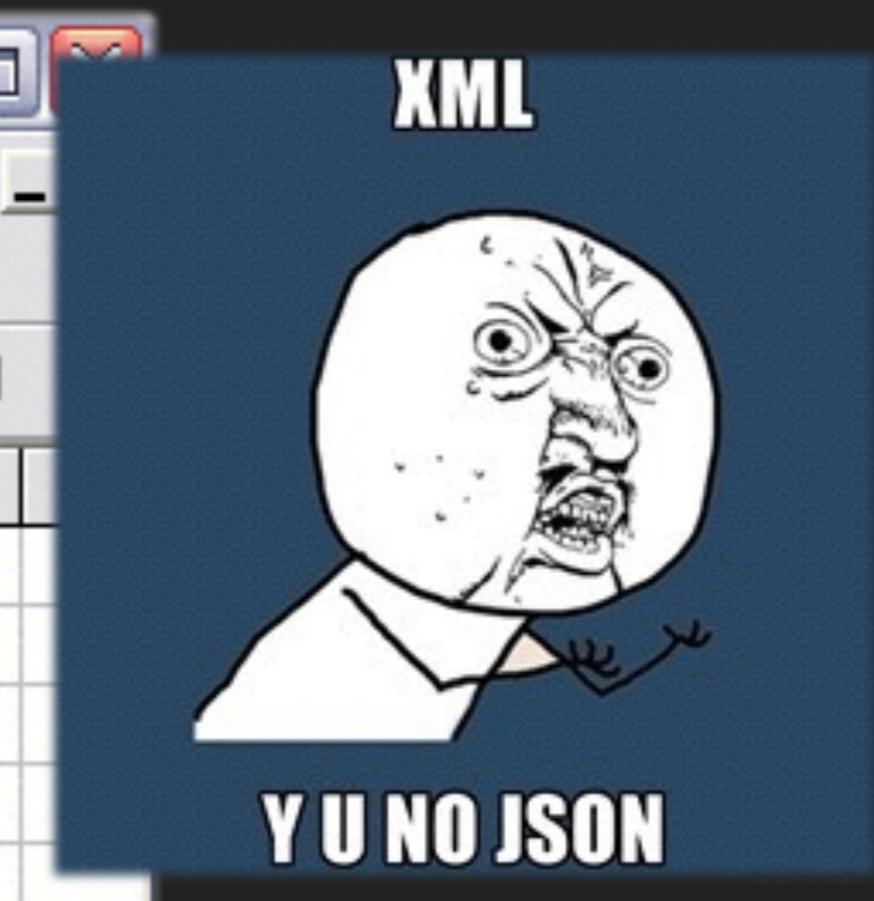
<http://headwayscareer.com/what-is-open-source-technology/>

## DATABASE, RDBMS VS NOSQL(JSON) VS XML

Microsoft Access - [Individual : Table]

File Edit View Insert Format Records Tools Window Help

XML



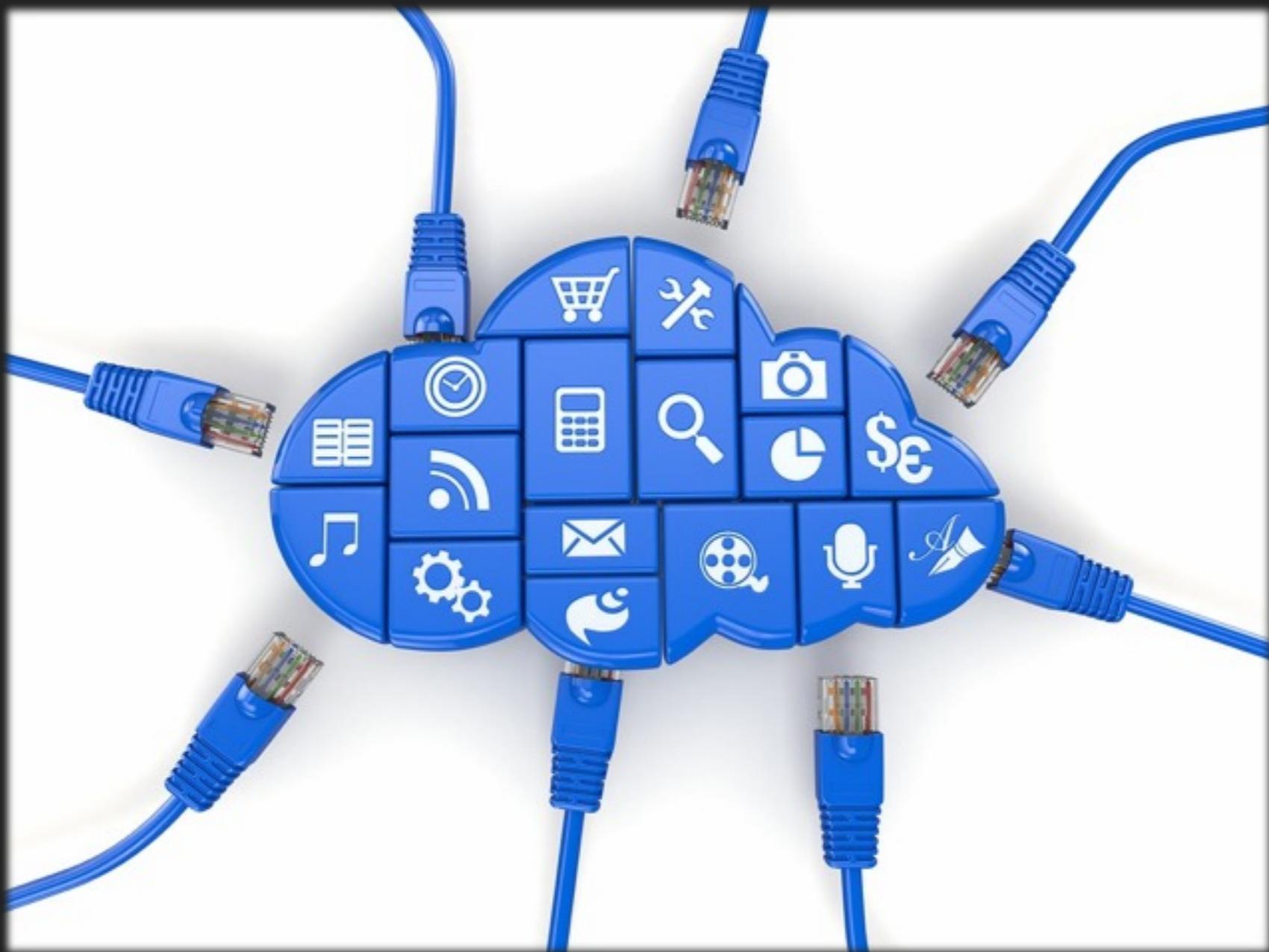
	IndividualId	FirstName	EmailAddress
	1	Homer	homer@quackit.com
▶	2	Barney	barney@quackit.com
	3	Ozzy	ozzy@quackit.com
	4	Fred	fred@quackit.com
*	(AutoNumber)		

```
1 {  
2   "key라부르고": "value라읽지요",  
3   "키는열쇠": "밸류는내용물",  
4   "저장하고": "꺼내씁니다"  
5 };
```

**BIGDATA & DATA MINING, DATA ANALYST**



<http://www.rsc.org/chemistryworld/News/2011/May/23051103.asp>



<http://www.clicdata.com/blog/automate-your-reporting-and-data-imports-with-an-api/>

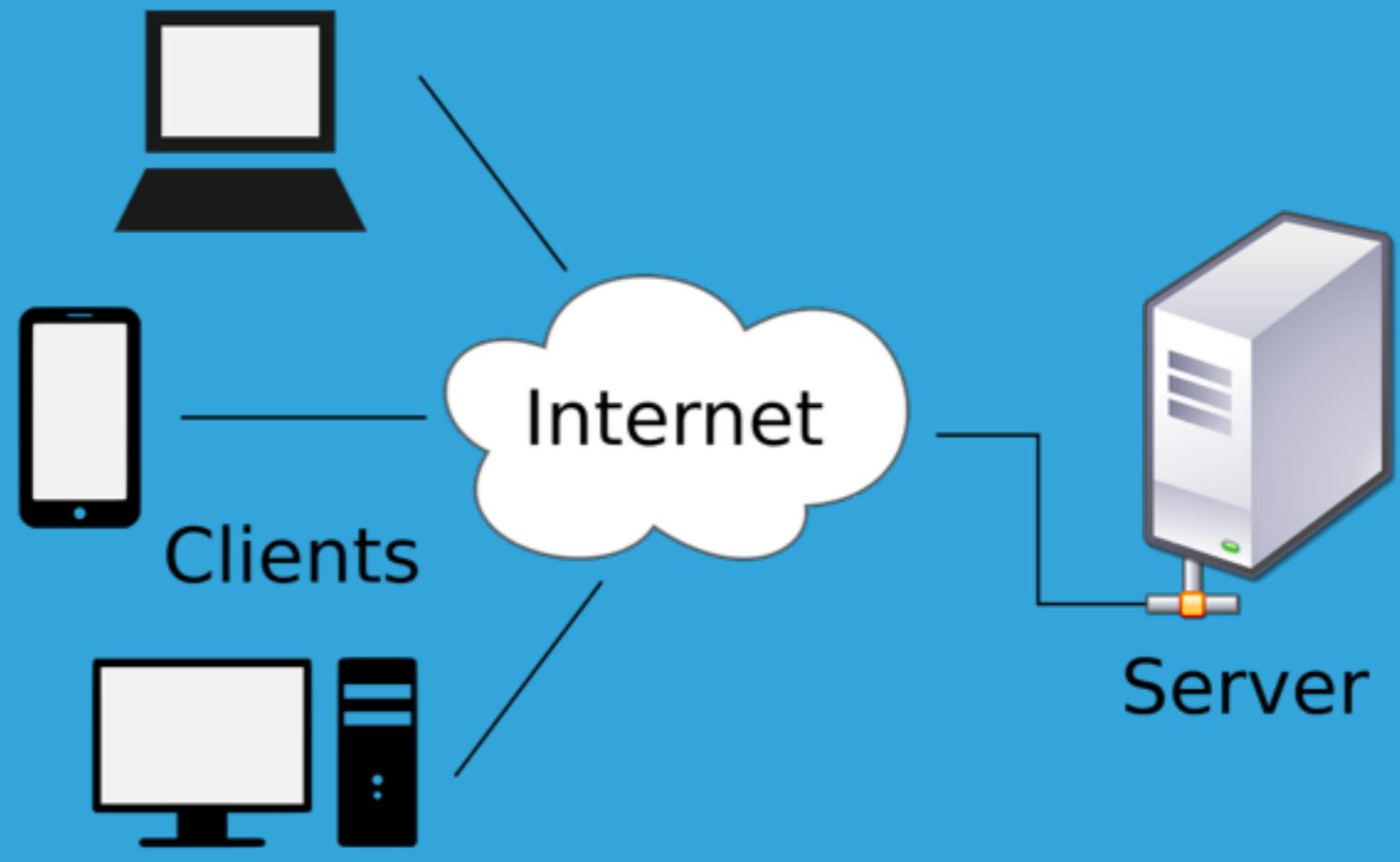


# DO EVERYTHING YOU WANT



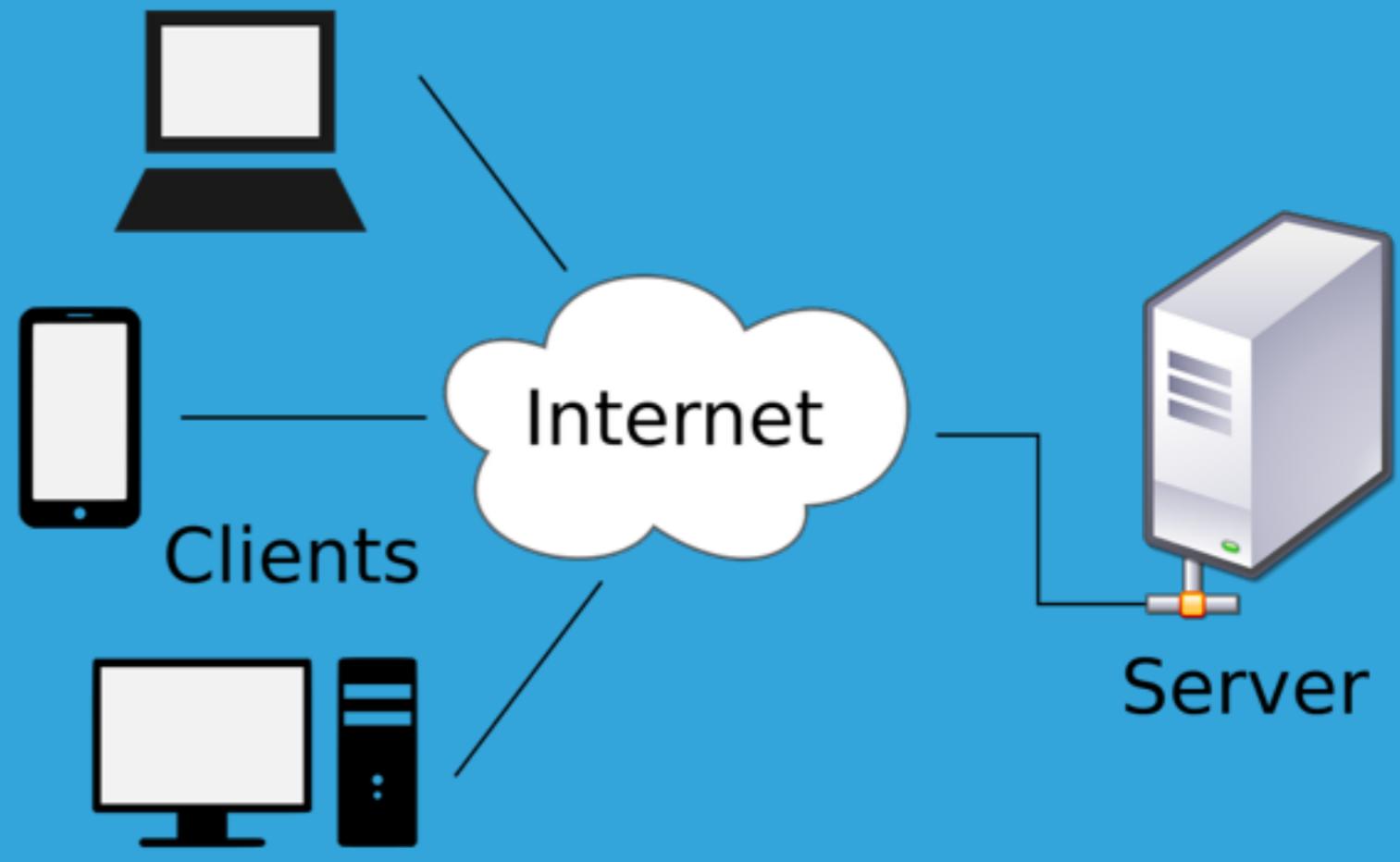
believe or not

# CLIENT TO SERVER



web page with html, css, **JAVASCRIPT**

# CLIENT TO SERVER



web page with html, css, **JAVASCRIPT**

# JAVASCRIPT

# JAVASCRIPT HISTORY

---

- ▶ JavaScript는 넷스케이프 커뮤니케이션즈 코퍼레이션의 브렌던 아이크(Brendan Eich)가 처음에는 모카(Mocha)라는 이름으로, 나중에는 라이브스크립트(LiveScript)라는 이름으로 개발하였으며, 최종적으로 JavaScript라는 이름으로 발표되었다.
- ▶ JavaScript는 객체 기반의 스크립트 프로그래밍 언어이다. 이 언어는 웹브라우저 내에서 주로 사용한다.
- ▶ 프로그래밍 언어로서 저평가 받는 시기도 있었으나 리치 콘텐츠(Rich Content)를 작성할 수 있는 AJAX(Asynchronous JavaScript + XML)의 등장으로 인해 JavaScript의 가치는 재검토되었다.
- ▶ HTML5에서 HTML5의 API로 JavaScript를 공식 채택함으로써 JavaScript는 세계에서 가장 인기 있는 프로그래밍 언어 중 하나로 자리 잡아가고 있다.

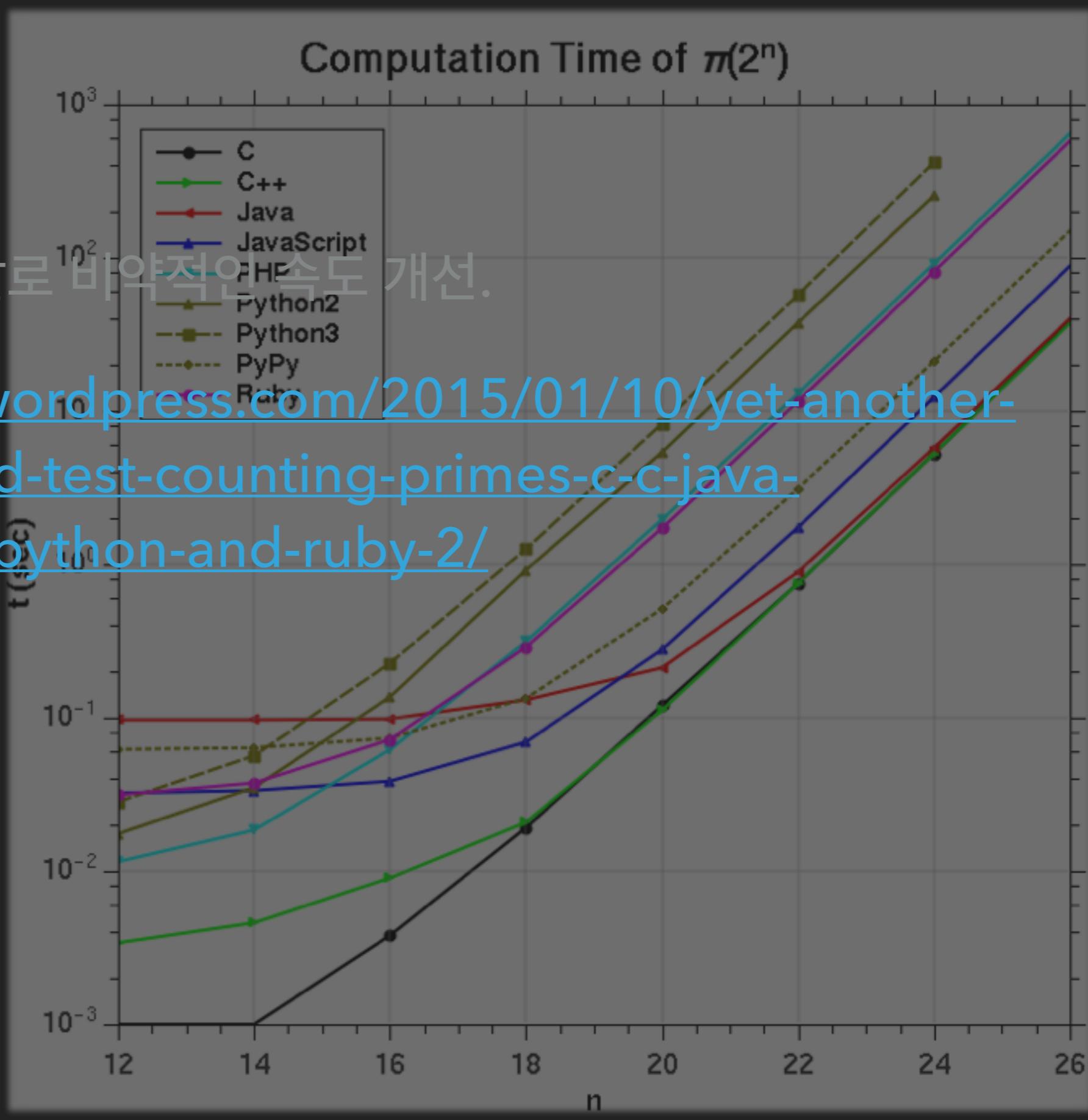
# JAVASCRIPT의 오해

---

- ▶ javascript는 단순하다.
- ▶ 웹 브라우저 상의 간단한 기능을 위해 심플한 코드들이 주로 사용되면서 생긴 오해. 자유로운 문법의 js는 Server/DB 등에서 이미 널리 사용되고 있다.
- ▶ 런타임(Run-time) 실행 환경은 버그가 많다.
- ▶ 모든 프로그램은 개발 시 무수한 테스트를 요한다. 느슨한 타입 정책과 런타임 실행으로 인해 예상 가능한 기능의 오류나 버그는 여타의 언어와 마찬가지로 테스트를 통해 극복 할 수 있다.

# JAVASCRIPT의 오해

- ▶ js는 느리다.
- ▶ V8 엔진 등의 개발로 ~~비약적인 속도 개선~~ 개선.
- ▶ <https://bjpelc.wordpress.com/2015/01/10/yet-another-language-speed-test-counting-primes-c-c-javascript-php-python-and-ruby-2/>



# JAVASCRIPT의 오해

- ▶ 스크립트(script) 언어는 진정한 언어가 아니다.
  - ▶ 대표적인 스크립트 언어 Python, Perl, Ruby
  - ▶ 프로그래밍의 논리적 흐름을 서술 할 수 있다면 그 자체로 이미 언어



# JAVASCRIPT RANK BY GITHUT

<http://www.sitepoint.com/whats-best-programming-language-learn-2015/>

- ▶ GitHub is a relatively new resource which analyzes 2.2 million active repositories on GitHub.

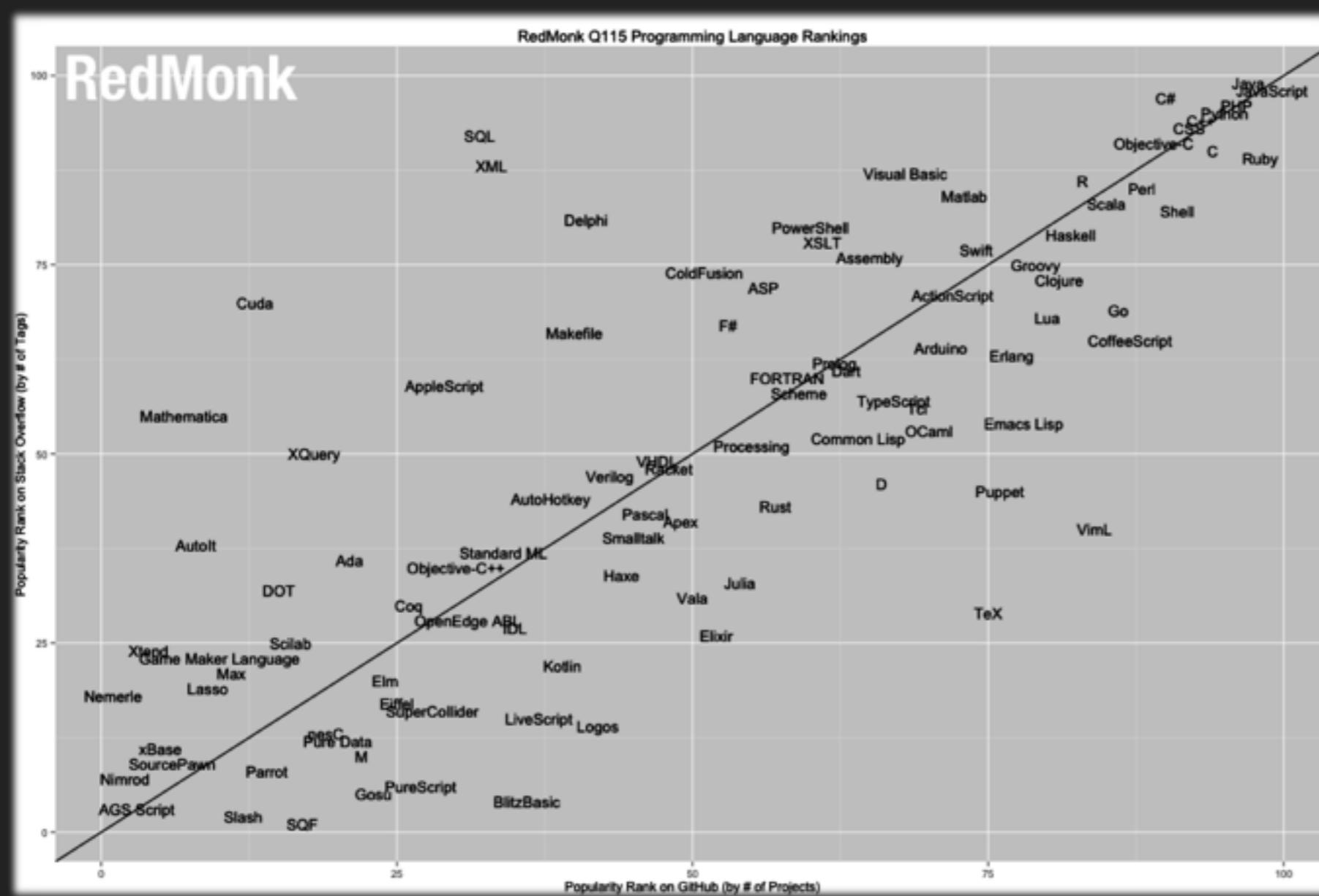
JAVASCRIPT  
JAVA  
PYTHON  
CSS  
PHP  
RUBY  
C++  
C  
SHELL  
C#



# JAVASCRIPT RANK BY REDMONK

<http://www.sitepoint.com/whats-best-programming-language-learn-2015/>

- ▶ RedMonk's language ranking for 2015 determines popularity by analyzing activity on both GitHub and StackOverflow.

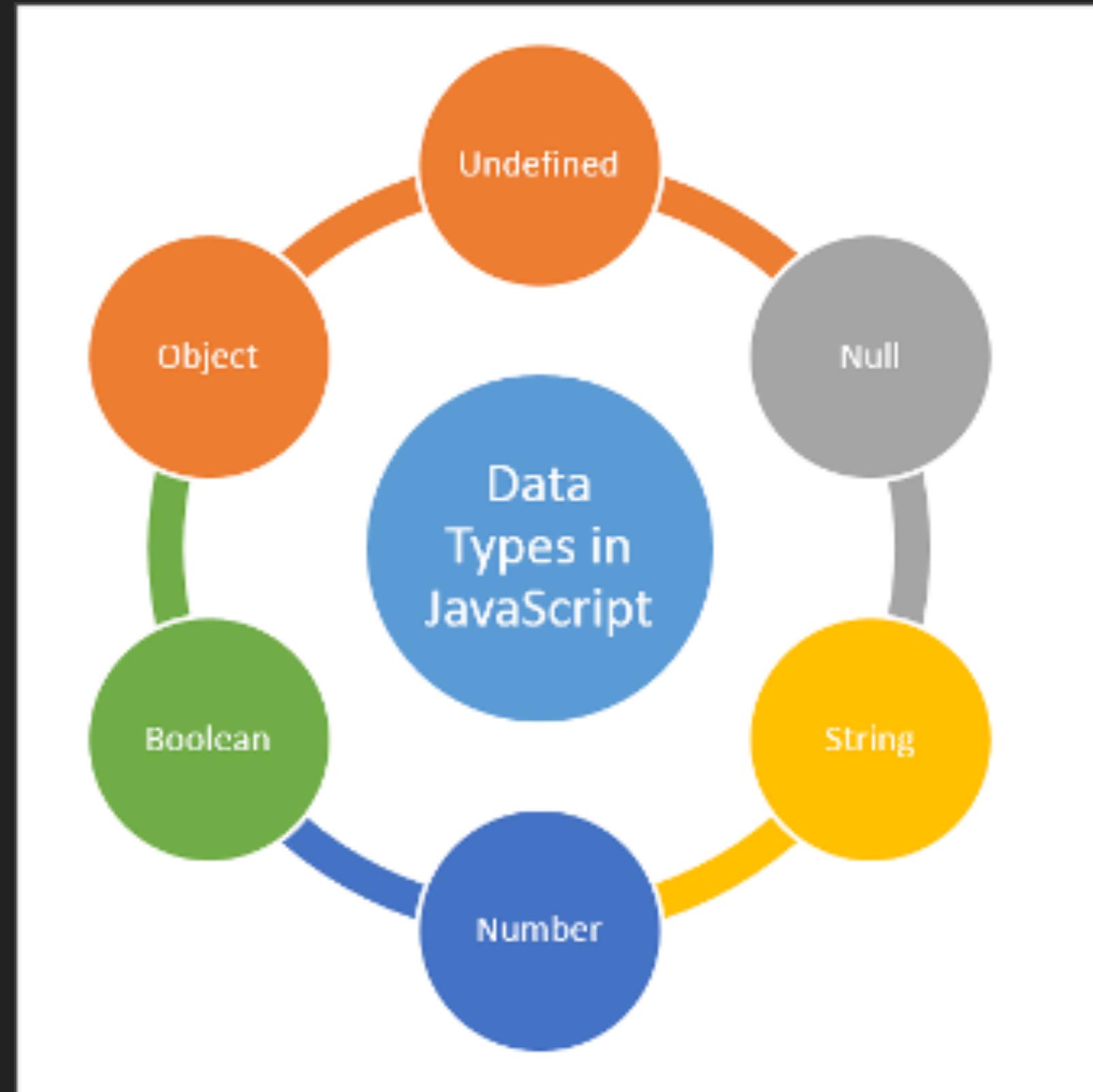


JAVASCRIPT  
JAVA  
PHP  
PYTHON  
C#  
C++  
RUBY  
CSS  
C  
OBJECTIVE-C

# TYPE

---

- ▶ Object
- ▶ String
- ▶ Number
- ▶ Boolean
- ▶ Null
- ▶ Undefined



# OPERATION

---

- ▶ `+, -, *, /, %, ++, --`
- ▶ `<, >, <=, >=, ==, ===, !=, !==`
- ▶ `=, +=, -=, *=, /=, %=`
- ▶ `&&, ||, !, ?(3항연산자)`
- ▶ `&, |, ^, <<, >>`

# STATEMENTS

---

- ▶ for
- ▶ if-else
- ▶ switch
- ▶ while / do-while
- ▶ try-catch-finally
- ▶ break
- ▶ continue
- ▶ var
- ▶ function
- ▶ debugger
- ▶ return
- ▶ delete

# ARRAY

---

- ▶ `var arr = [];` or `var arr = new Array();`
- ▶ 오브젝트의 리스트로 구성된 객체
- ▶ 가변적 크기
- ▶ Methods
  - ▶ `push()` / `pop()` / `shift()` / `unshift()`
  - ▶ `slice(start, end)` / `splice(position, number, [items...])`
  - ▶ `length`
  - ▶ `toString()` / `join()` / `concat()`
  - ▶ `arr[index]`
  - ▶ `delete`
  - ▶ `sort()` / `reverse()`

# FUNCTION

---

- ▶ `function name(parameter1, parameter2, ... ) {  
 code to be executed  
 return;  
}`
- ▶ `var name = function (parameter1, parameter2, ... ) {  
 code to be executed  
 return;  
}`
- ▶ `name();`

# OBJECT

---

- ▶ 

```
var person = {  
    firstName:"John", lastName:"Doe", age:50,  
    fullName: function() {return this.firstName + " " +  
        this.lastName;}  
};
```
- ▶ `person.firstName;`
- ▶ `person["firstName"];`
- ▶ `person.fullName();`