



진정원 WWW.WEBOOSTER.CO.KR

JAVASCRIPT

진정원

- ▶ embedded / mobile / server / db / web
- ▶ webooster.co.kr
- ▶ Meteor.js School / Startup
- ▶ 010-6617-6309 / ceo@webooster.co.kr

IT TERMS

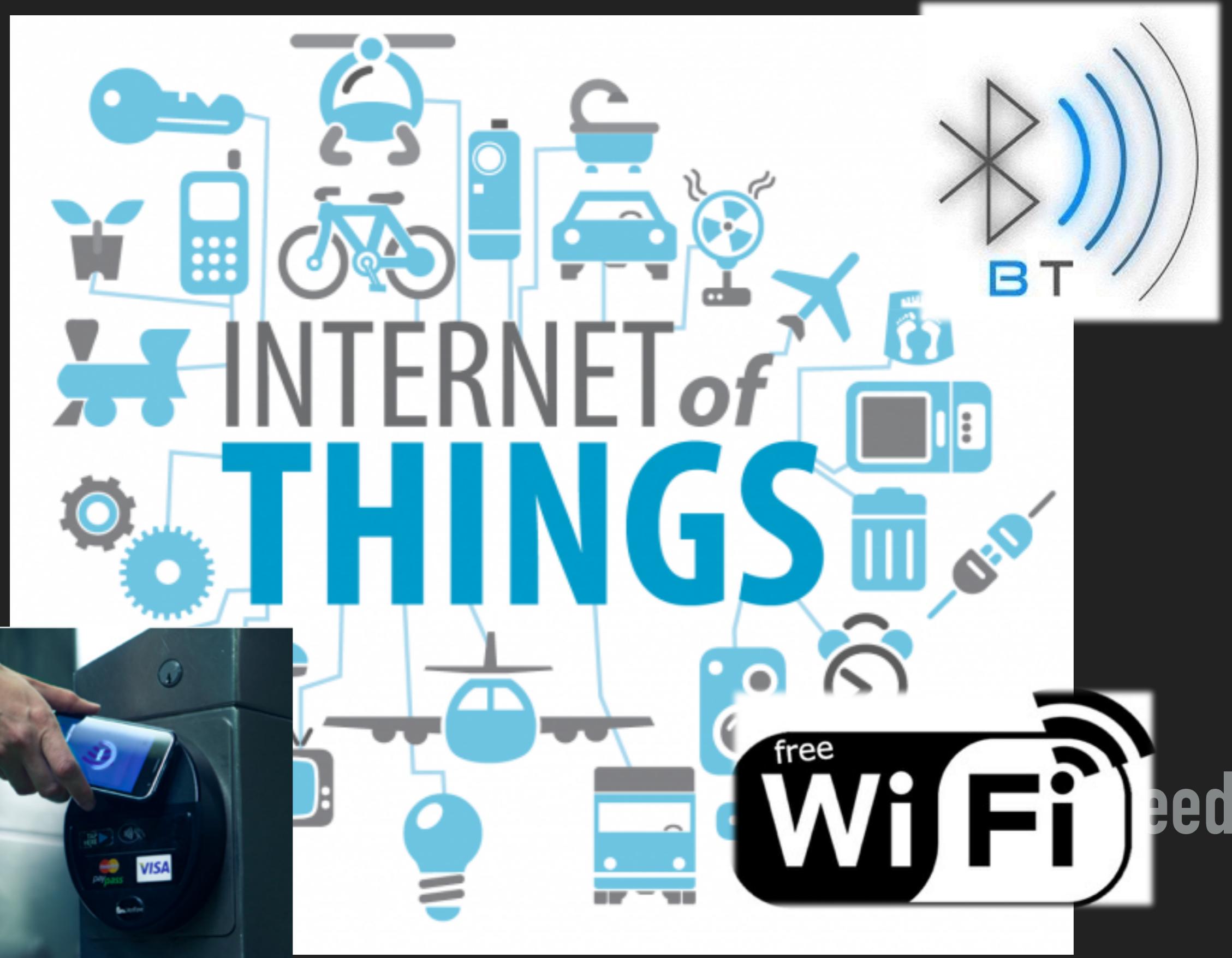


facebook

facebook

facebook map

IOT, INTERNET OF THINGS



IPV4 VS IPV6

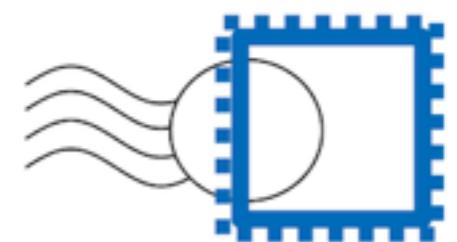
An IPv6 address

(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

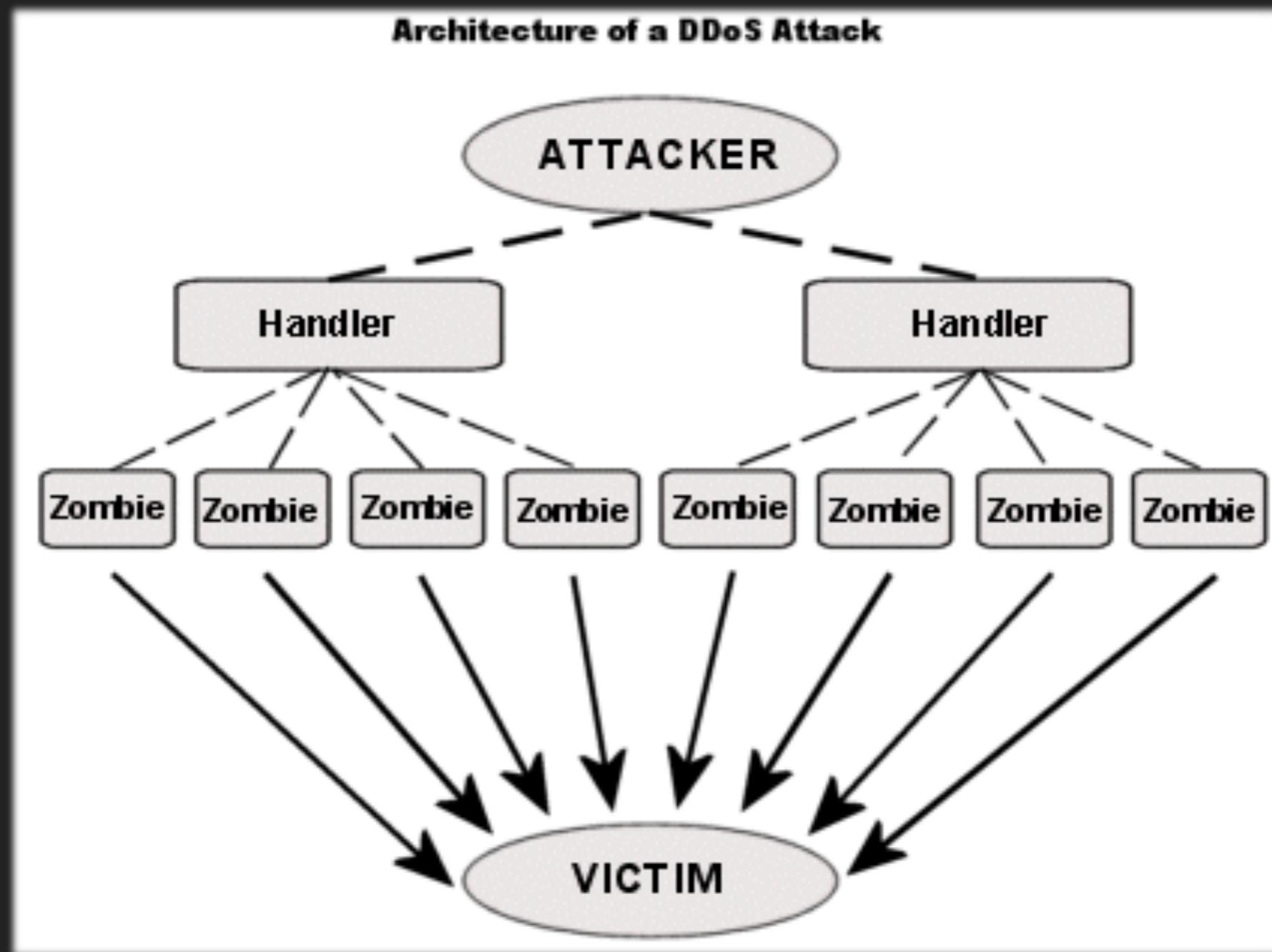
<https://en.wikipedia.org/wiki/IPv6>

I'M BOT!

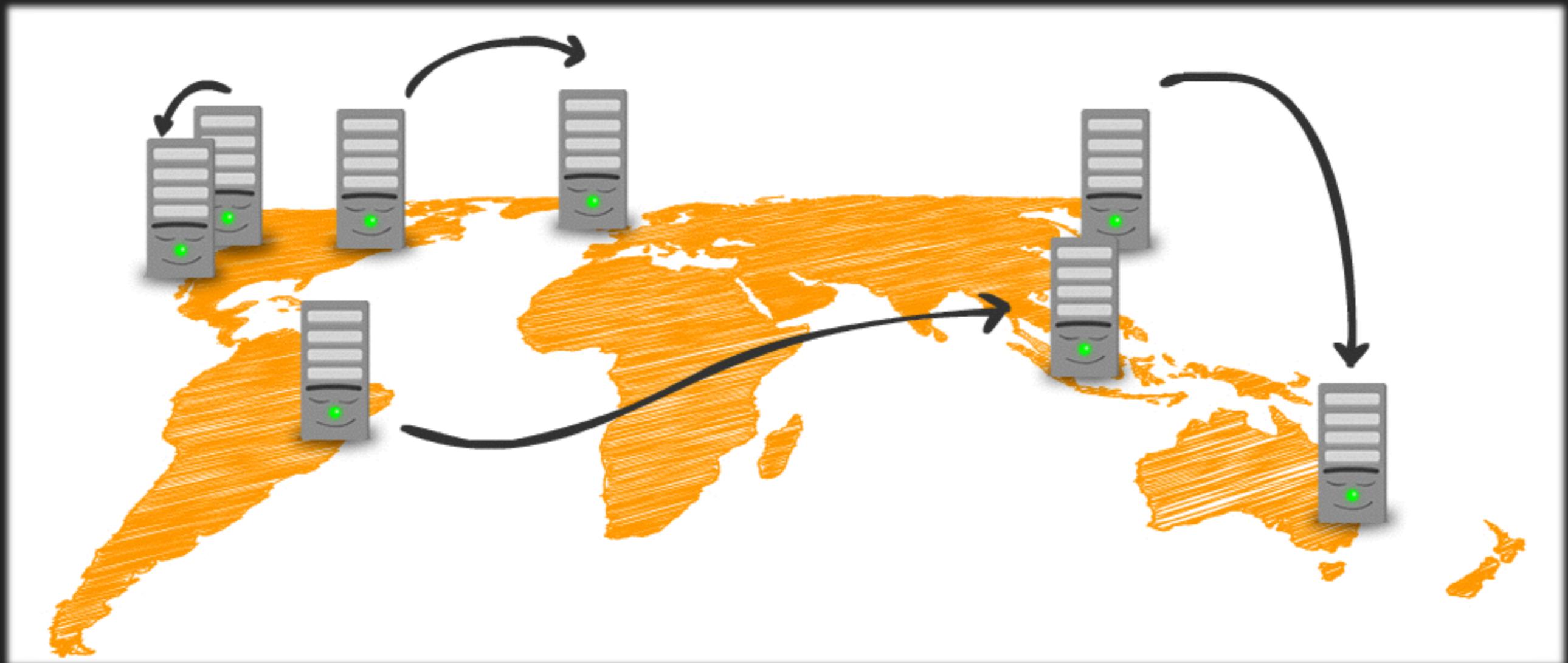


<http://www.filothaea.com/blog/google-tips-how-to-call-googlebot-to-index-your-new-website-blog-quickly/>

DDOS ATTACKING



Johnny Appleseed



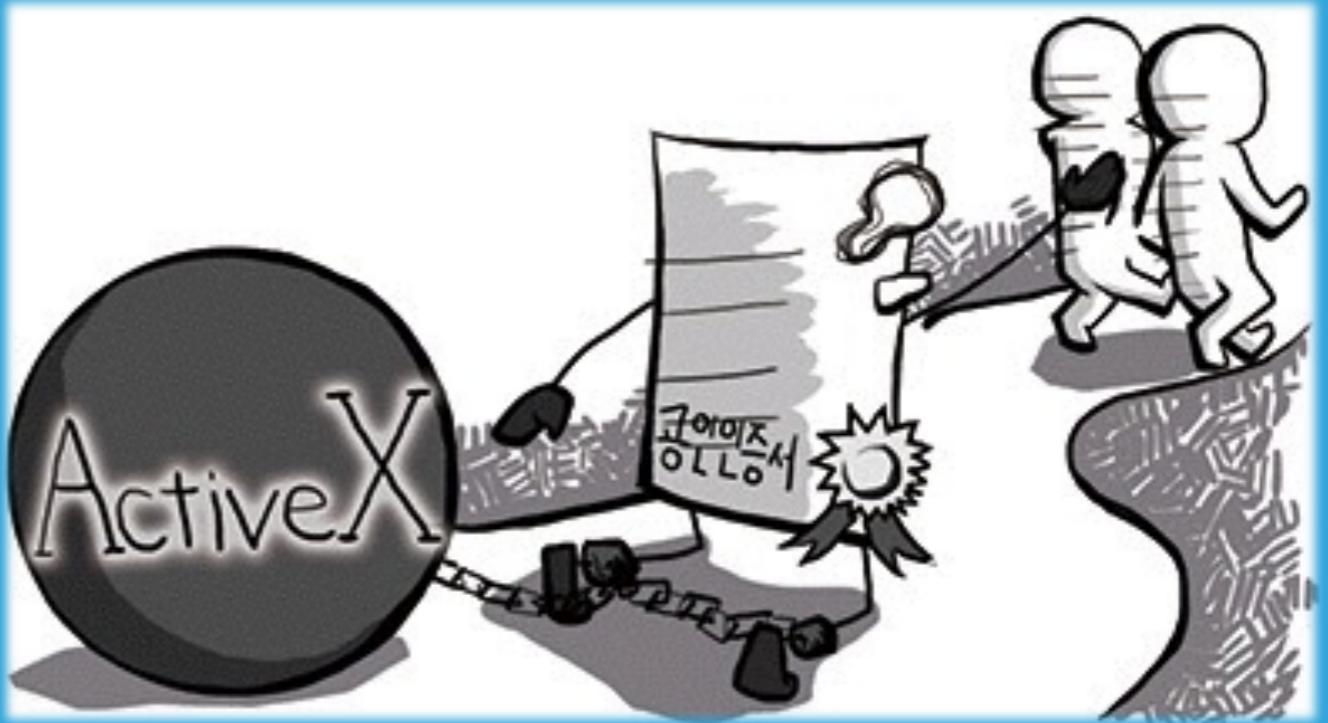
<http://blog.takipi.com/aws-olympics-speed-testing-amazon-ec2-s3-across-regions/>

WINDOWS VS MAC



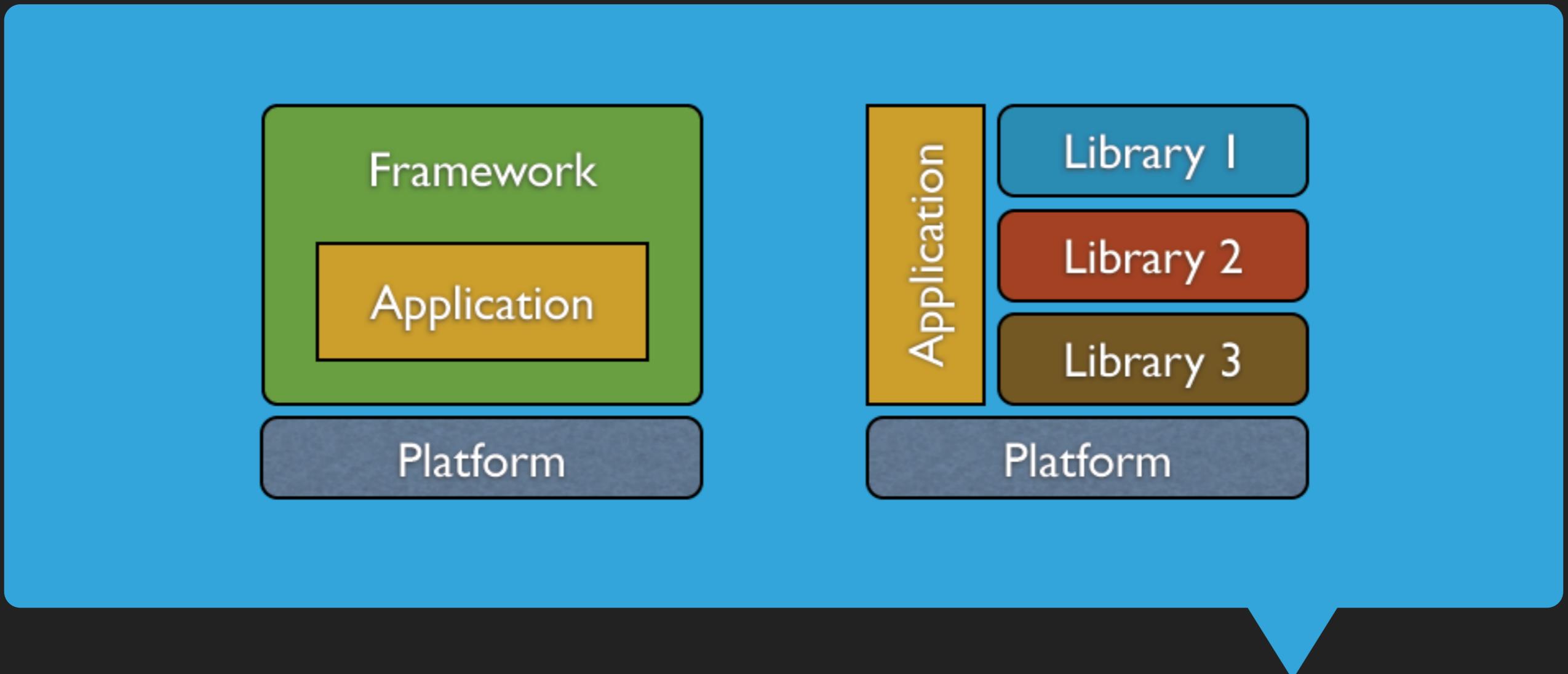
first item of google search: windows vs mac

EXPLORER & ACTIVE-X & 공인인증서

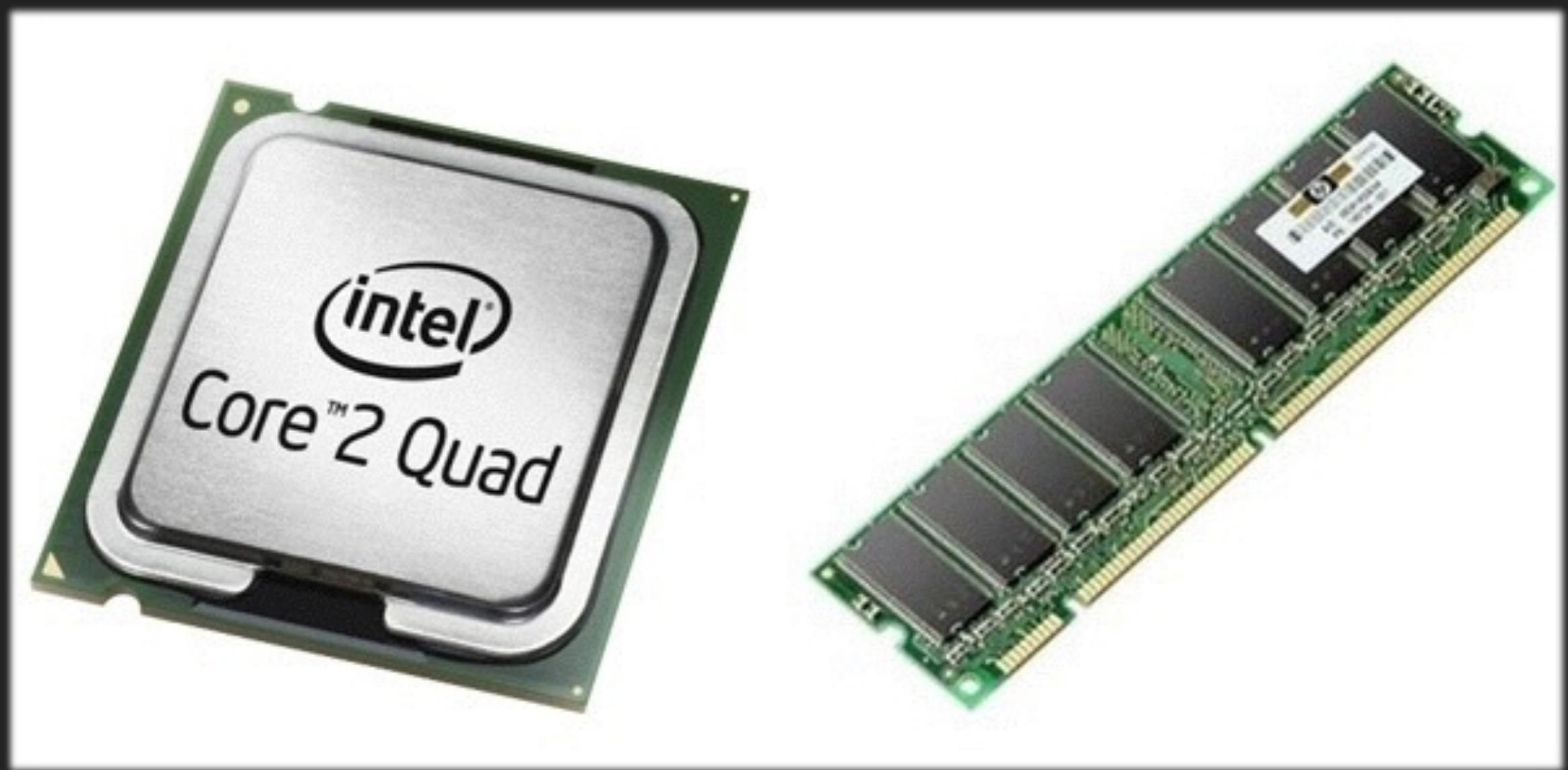


<http://uncyclopedia.kr/wiki/%EC%9C%A0%EC%8B%9D>
<http://times.kaist.ac.kr/news/articleView.html?idxno=2628>

LIBRARY & FRAMEWORK & PLATFORM



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

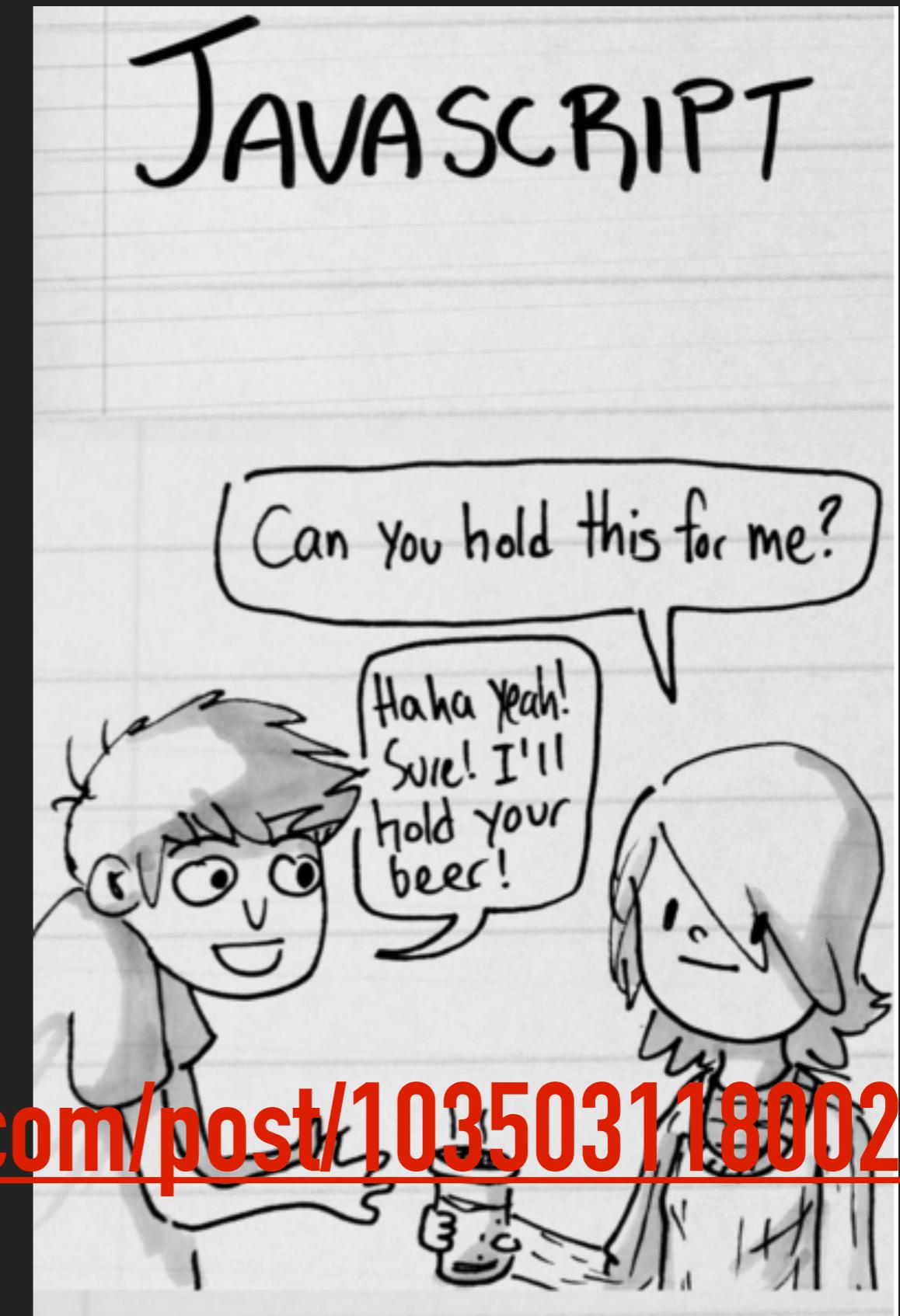
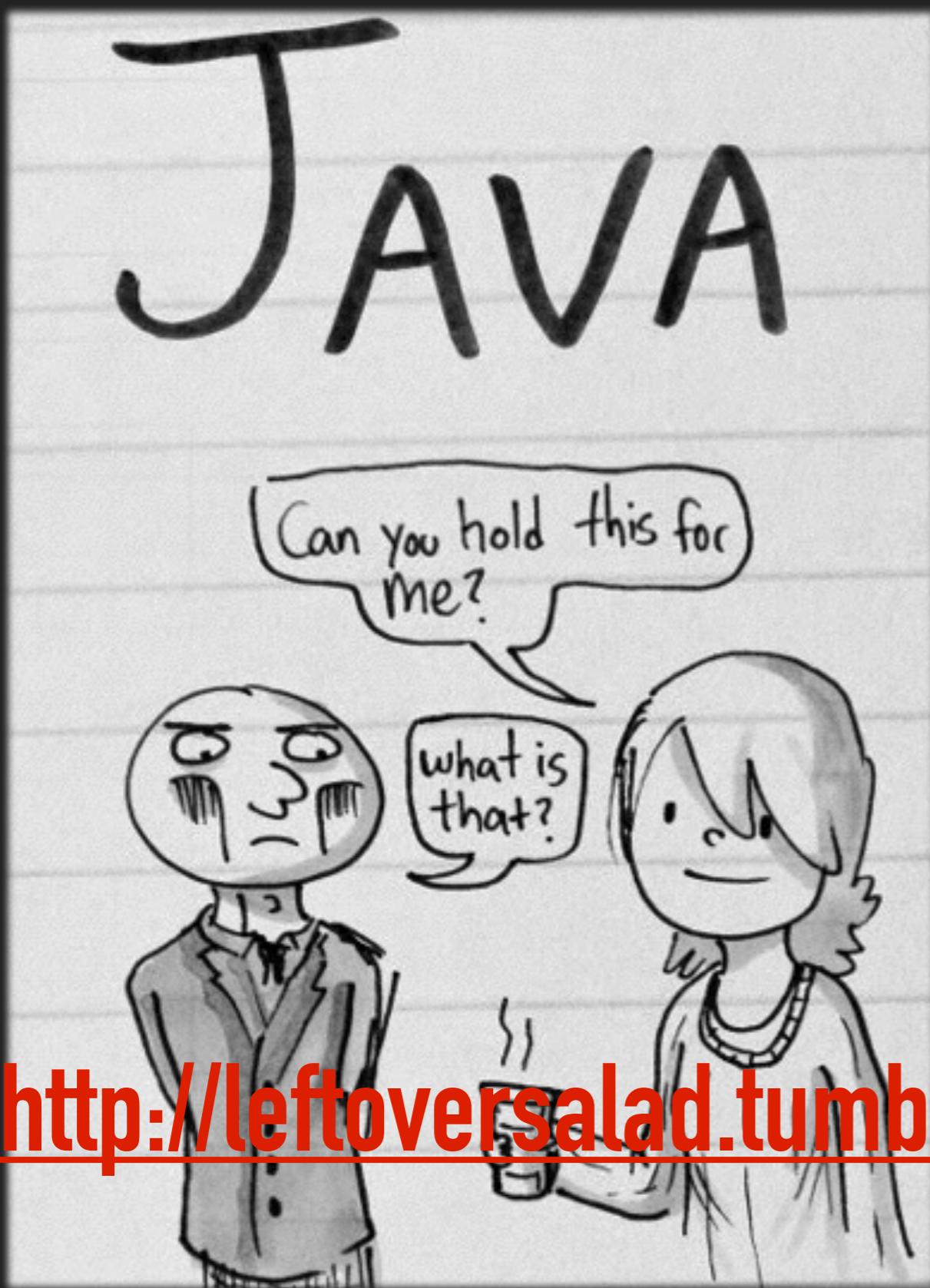


[http://vsphere-land.com/news/understanding-cpu-memory-
management-in-vsphere.html](http://vsphere-land.com/news/understanding-cpu-memory-management-in-vsphere.html)

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...

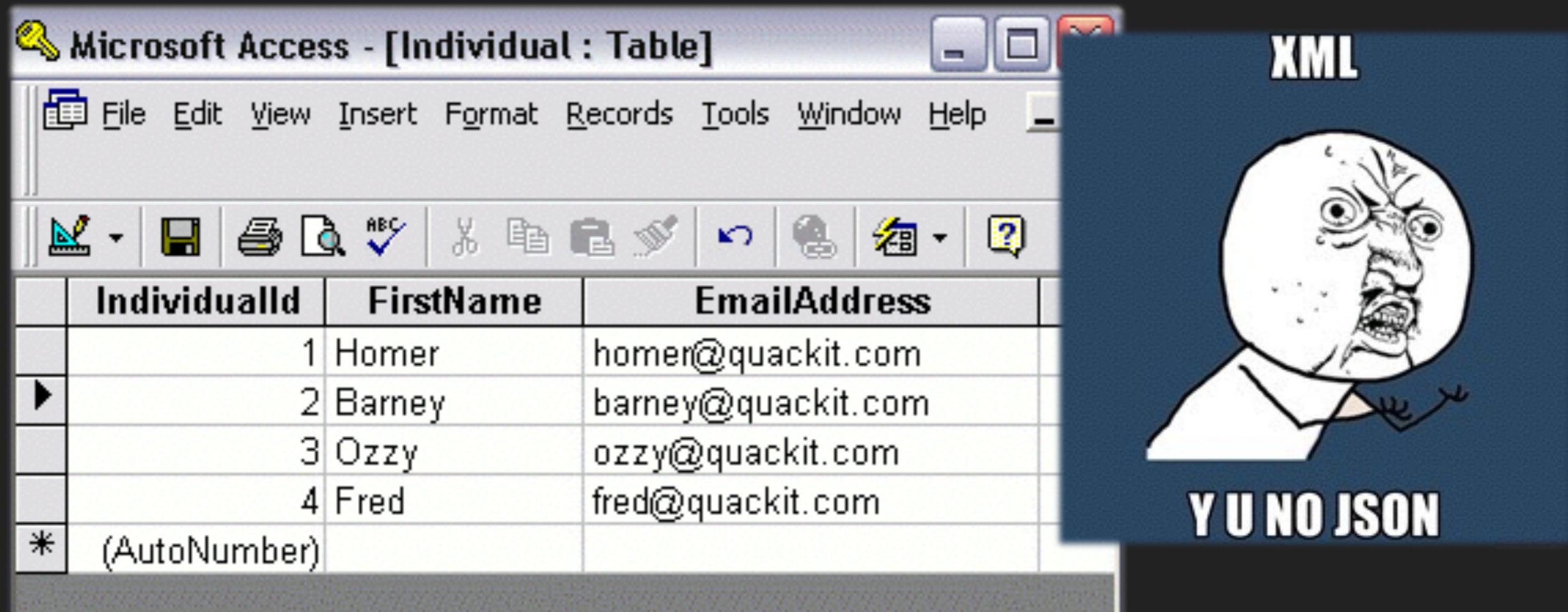

```

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



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

DATABASE, RDBMS VS NOSQL(JSON) VS XML

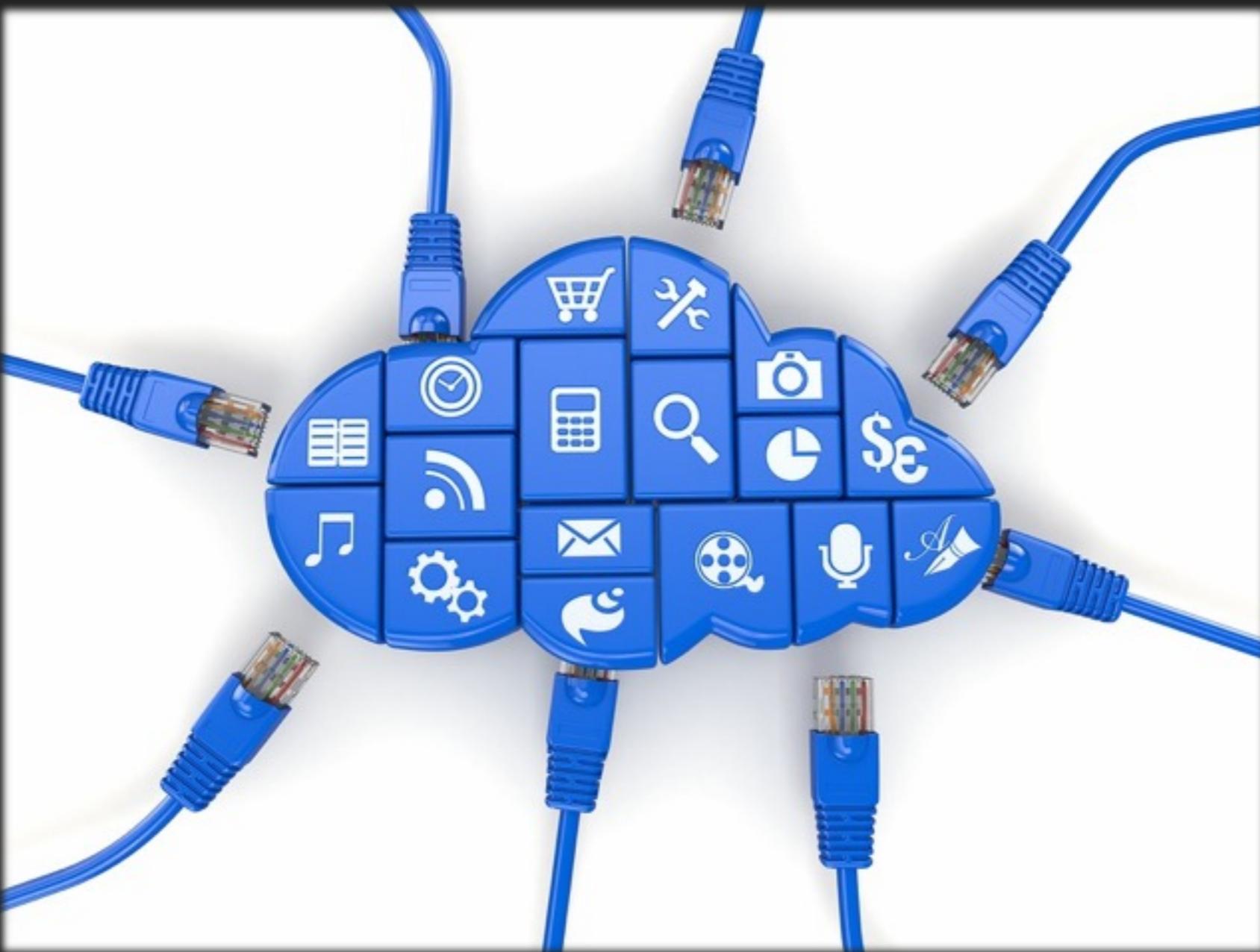


```
1 {  
2   "key라부르고": "value라읽지요",  
3   "키는열쇠": "밸류는내용물",  
4   "저장하고": "꺼내씁니다"  
5 };
```

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/>

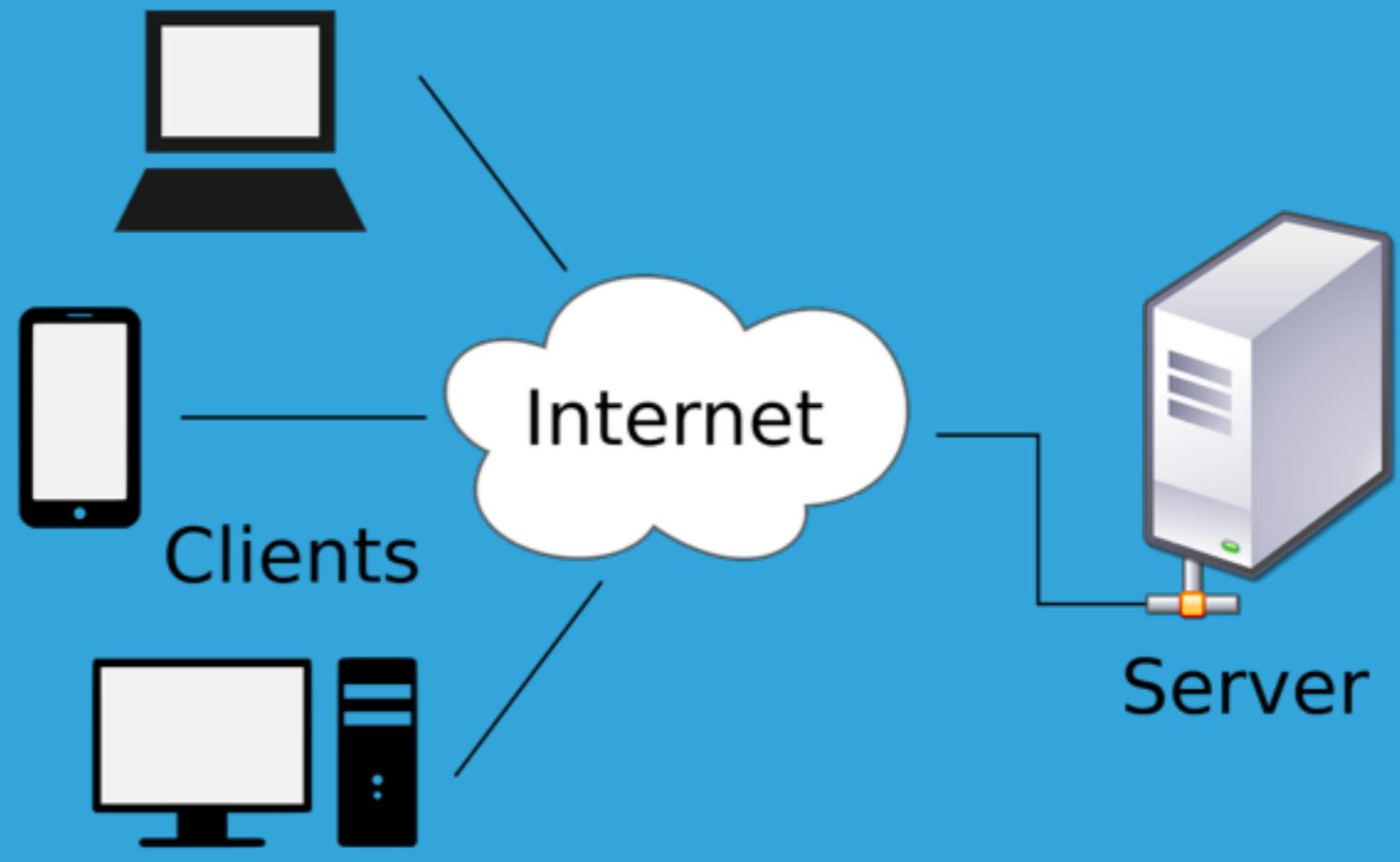


DO EVERYTHING YOU WANT



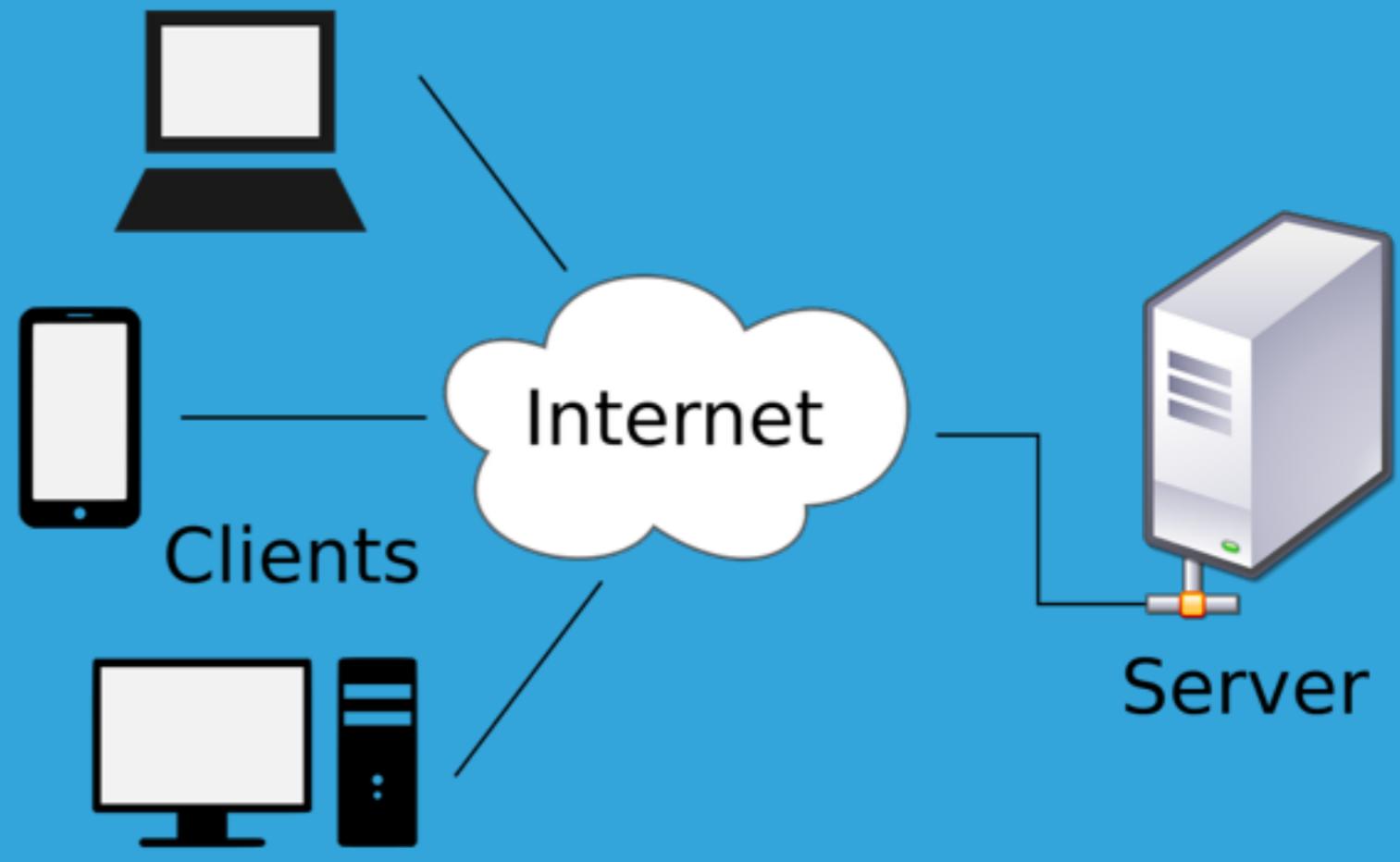
believe or not

CLIENT TO SERVER



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

CLIENT TO SERVER



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

JAVASCRIPT

JAVASCRIPT HISTORY

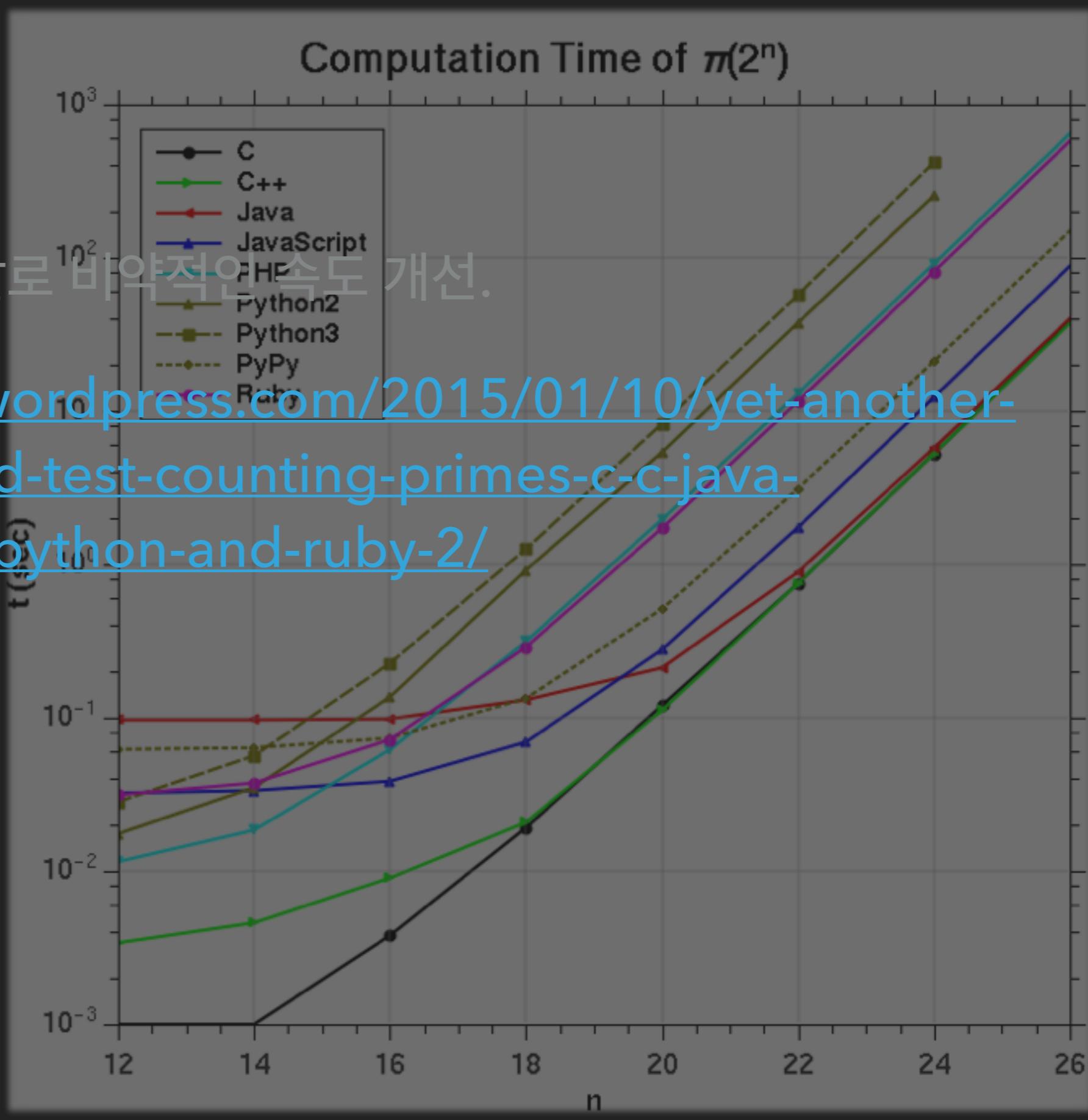
- ▶ JavaScript는 넷스케이프 커뮤니케이션즈 코퍼레이션의 브렌던 아이크(Brendan Eich)가 처음에는 모카(Mocha)라는 이름으로, 나중에는 라이브스크립트(LiveScript)라는 이름으로 개발하였으며, 최종적으로 JavaScript라는 이름으로 발표되었다.
- ▶ JavaScript는 객체 기반의 스크립트 프로그래밍 언어이다. 이 언어는 웹브라우저 내에서 주로 사용한다.
- ▶ 프로그래밍 언어로서 저평가 받는 시기도 있었으나 리치 콘텐츠(Rich Content)를 작성할 수 있는 AJAX(Asynchronous JavaScript + 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-java-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.

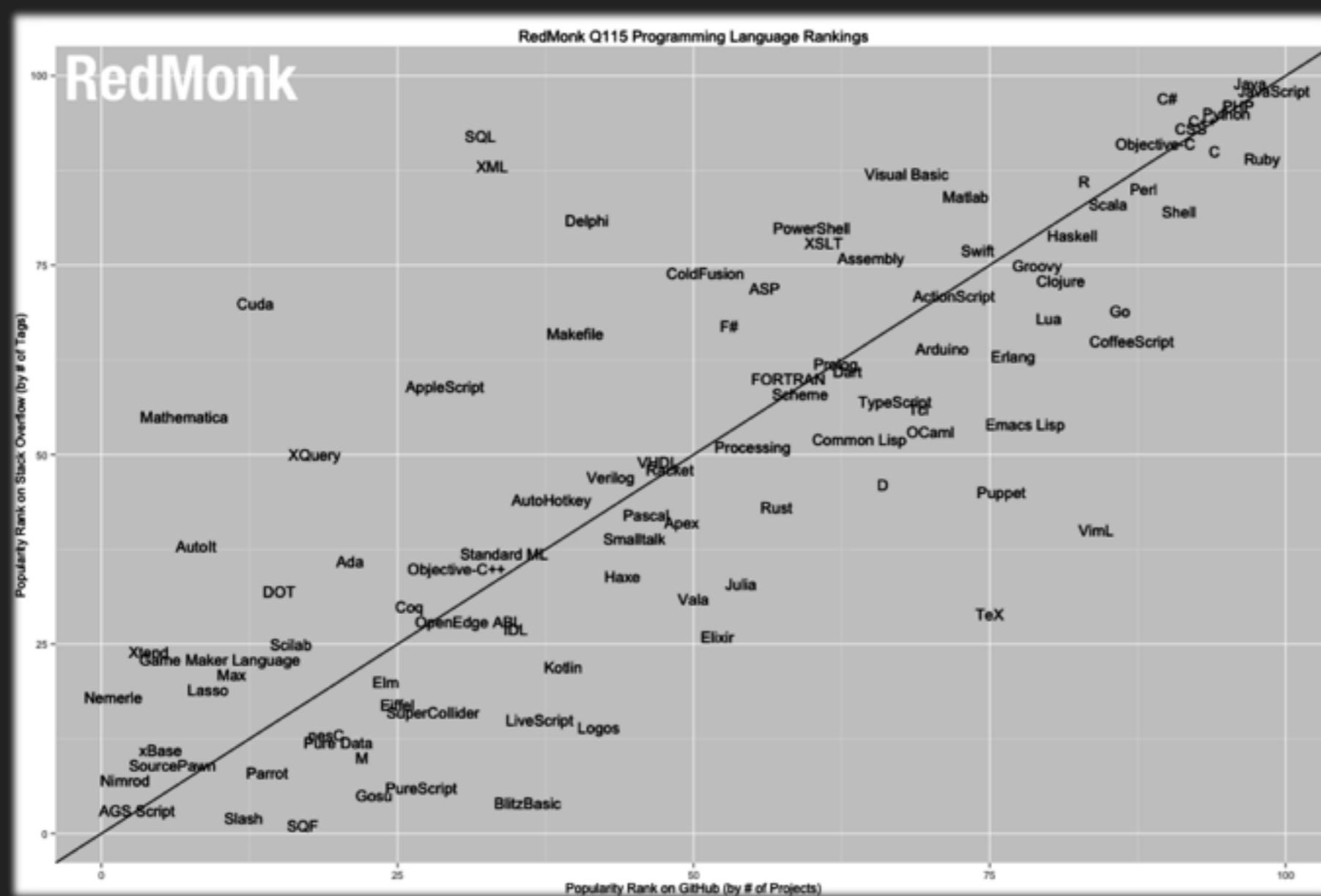
JAVASCRIPT
JAVA
PYTHON
CSS
PHP
RUBY
C++
C
SHELL
C#



JAVASCRIPT RANK BY REDMONK

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

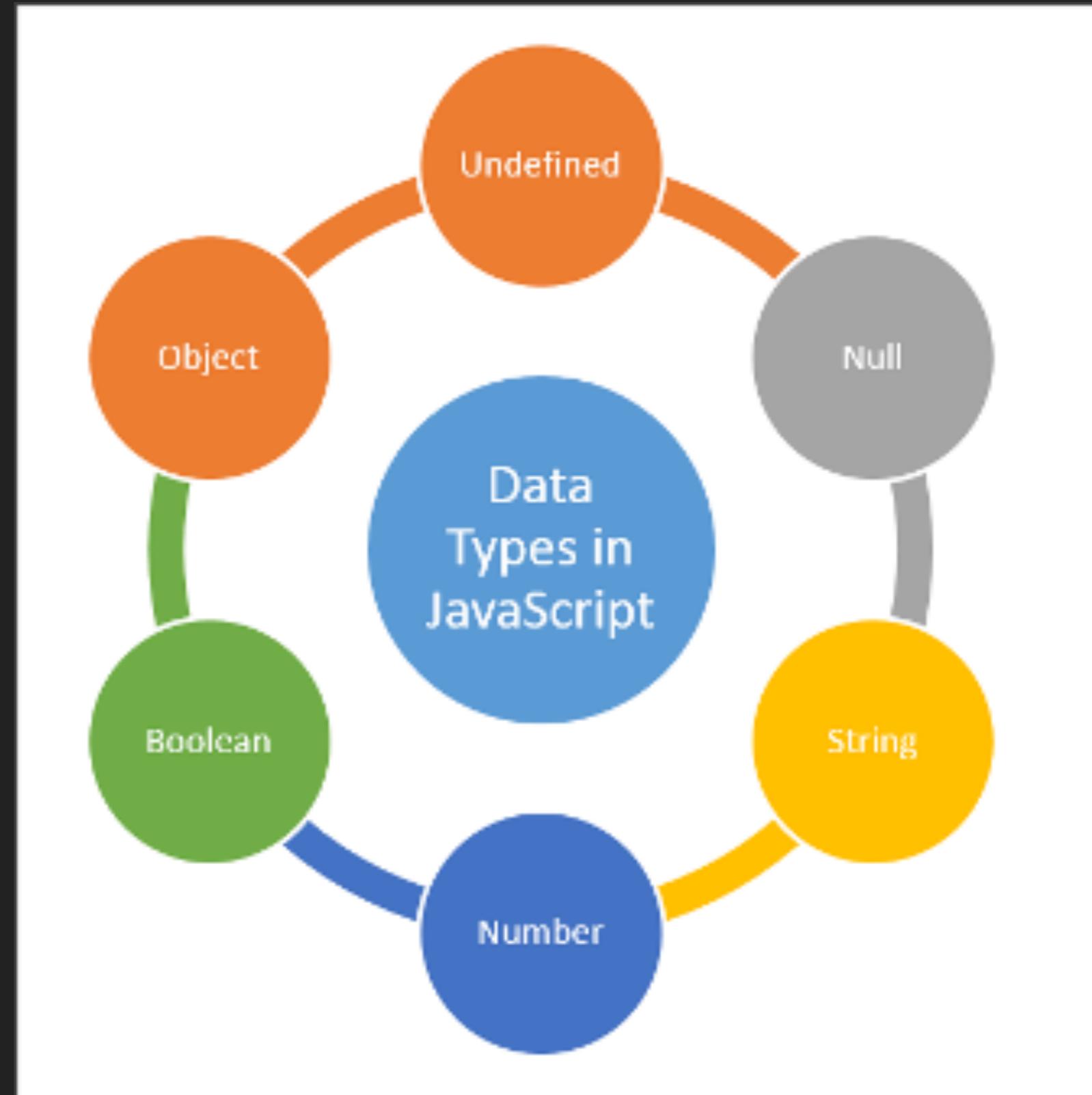
- ▶ RedMonk's language ranking for 2015 determines popularity by analyzing activity on both GitHub and StackOverflow.



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();`

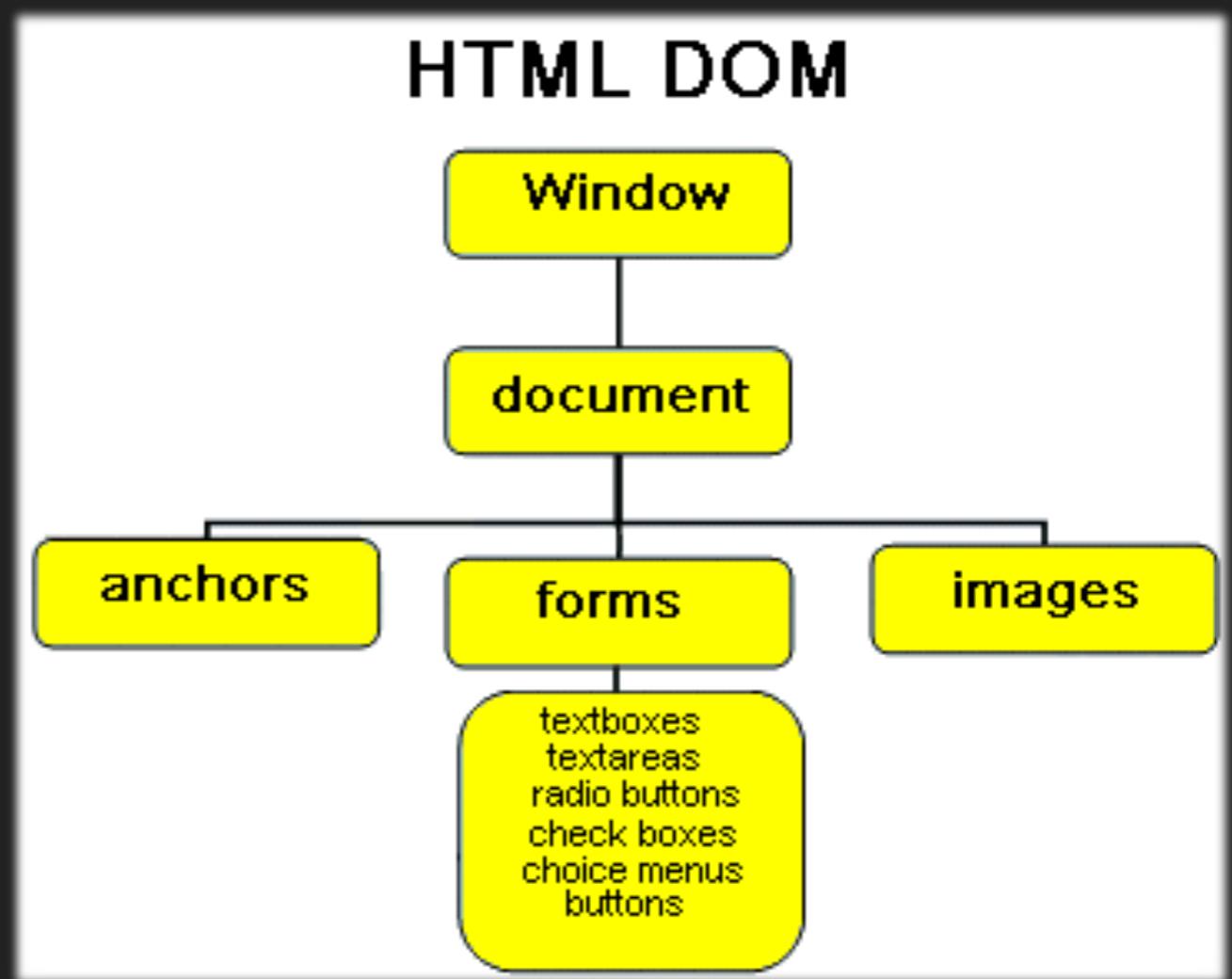
OBJECT

- ▶

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

DOM, DOCUMENT OBJECT MODEL

- ▶ html문서를 트리 구조로 변경하여 원하는 Object를 선택/조작 할 수 있도록 함.
- ▶ 부모/자식 관계를 따라 추적
- ▶ Element의 Attribute를 통한 추적
- ▶ 최상위 window 객체에 포함



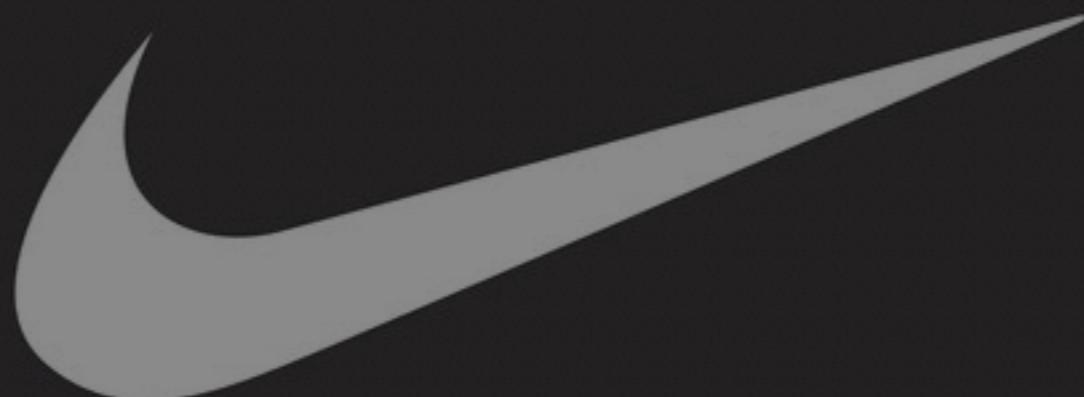
DOM CONTROL

- ▶ DOM에서 특정 element를 선택한다.
document.getElementsByName('태그명')
- ▶ document.getElementById ('id속성값')
- ▶ document.getElementsByClassName('class속성값')
- ▶ document.querySelector('찾고자 하는 Dom 객체에대한 CSS 선택자')
- ▶ document.querySelectorAll('찾고자 하는 Dom 객체에대한 CSS 선택자')

DOM CONTROL

- ▶ element를 활용한다.
- ▶ element.value
- ▶ element.textContent
- ▶ element.appendChild()
- ▶ element.insertBefore()
- ▶ element.replaceChild()
- ▶ element.removeChild()
- ▶ element.cloneNode()
- ▶ element.setAttribute('name', 'val')
- ▶ element.removeAttribute('name')

JUST GOOGLE IT.



DOM EVENT

- ▶ Event Driven Model
- ▶ type별 event handler에 function을 등록
- ▶ load, click
- ▶ mousedown, mousemove, mouseover, mouseup
- ▶ keydown, keypress, keyup
- ▶ change, reset, submit

DOM EVENT

- ▶ <input type="button" value="다이얼로그 표시" onclick="btn_onclick()">
- ▶ window.onload = function() { ... };
- ▶ window.addEventListener('load', function() { ... });

JQUERY

- ▶ 2006, John Resig
- ▶ DOM -> jQuery Object로 변환하여 조작
- ▶ Cross Browsing
- ▶ Effect / Animation
- ▶ AJAX
- ▶ Plug-in



SELECTORS

Basics

- *
- .class
- element
- #id
- selector1, selectorN, ...

Hierarchy

- parent > child
- ancestor descendant
- prev + next
- prev ~ siblings

Basic Filters

- :animated
- :eq()
- :even
- :first
- :gt()
- :header
- :lang()
- :last
- :lt()
- :not()
- :odd
- :root
- :target

Content Filters

- :contains()
- :empty
- :has()
- :parent

Visibility Filters

- :hidden
- :visible

Attribute

- [name ="value"]
- [name*= "value"]
- [name~= "value"]
- [name\$= "value"]
- [name= "value"]
- [name!="value"]
- [name^= "value"]
- [name]
- [name="value"][name2="value2"]

Child Filters

- :first-child
- :first-of-type
- :last-child
- :last-of-type
- :nth-child()
- :nth-last-child()
- :nth-last-of-type()
- :nth-of-type()
- :only-child
- :only-of-type()

Forms

- :button
- :checkbox
- :checked
- :disabled
- :enabled
- :focus
- :file
- :image
- :input
- :password
- :radio
- :reset
- :selected
- :submit
- :text

ATTRIBUTES / CSS	MANIPULATION	TRAVERSING
Attributes .attr() .prop() .removeAttr() .removeProp() .val()	Copying .clone() DOM Insertion, Around .wrap() .wrapAll() .wrapInner()	Filtering .eq() .filter() .first() .has() .is() .last() .map() .not() .slice()
CSS .addClass() .css() jQuery.cssHooks .hasClass() .removeClass() .toggleClass()	DOM Insertion, Inside .append() .appendTo() .html() .prepend() .prependTo() .text()	Miscellaneous Traversing .add() .andSelf() .contents() .each() .end()
Dimensions .height() .innerHeight() .innerWidth() .outerHeight() .outerWidth() .width()	DOM Insertion, Outside .after() .before() .insertAfter() .insertBefore()	Tree Traversal .addBack() .children() .closest() .find() .next() .nextAll() .nextUntil() .parent() .parents() .parentsUntil() .prev() .prevAll() .prevUntil() .siblings()
Offset .offset() .offsetParent() .position() .scrollLeft() .scrollTop()	DOM Removal .detach() .empty() .remove() .unwrap()	
Data jQuery.data() .data() jQuery.hasData() jQuery.removeData() .removeData()	DOM Replacement .replaceAll() .replaceWith()	

EVENTS

Browser Events

.error()
.resize()
.scroll()

Document Loading

.holdReady()
.load()
.ready()
.unload()

Event Handler Attachment

.bind()
.delegate()
.die()
.live()
.off()
.on()
.one()
.trigger()
.triggerHandler()
.unbind()
.undelegate()

Form Events

.blur()
.change()
.focus()
.select()
.submit()

Keyboard Events

.keydown()
.keypress()
.keyup()

Mouse Events

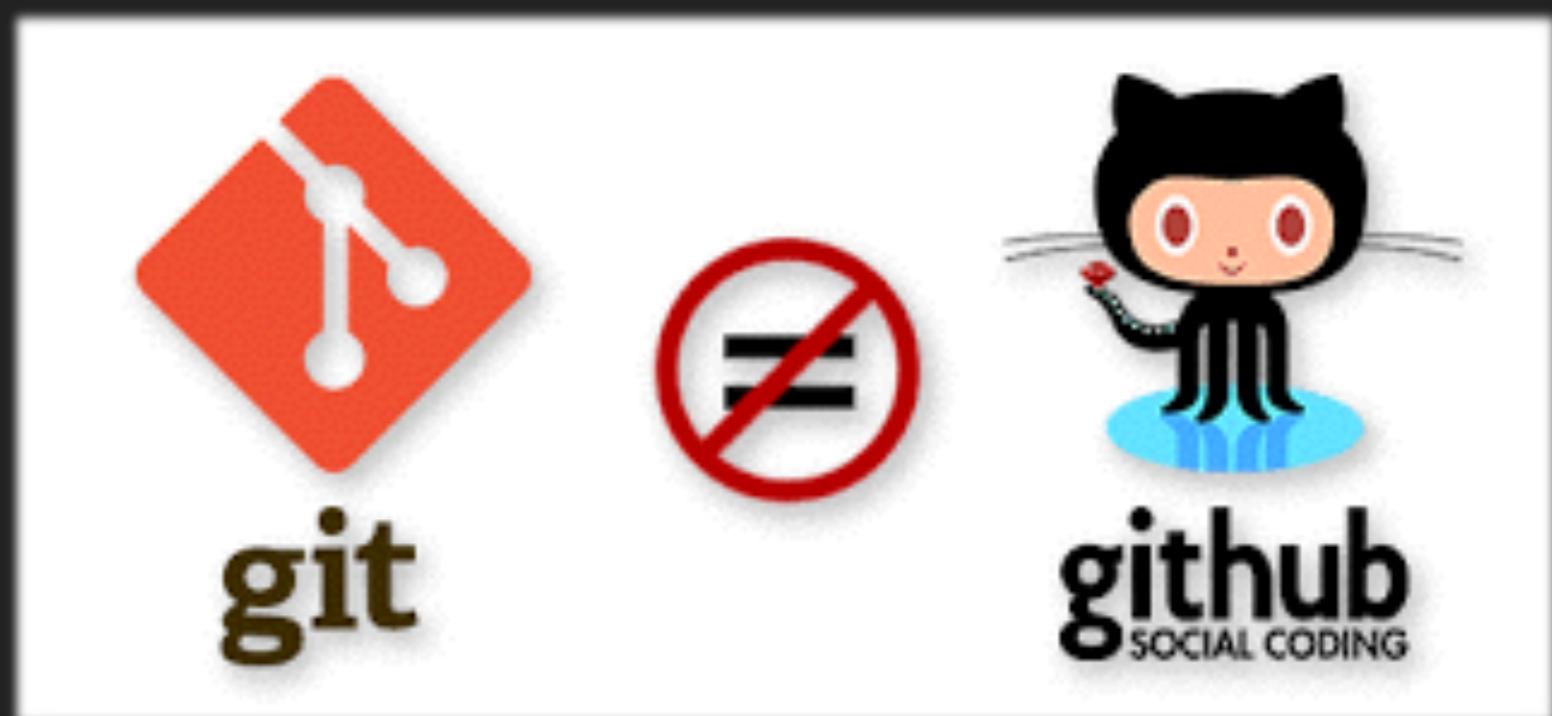
.click()
.dblclick()
.focusin()
.focusout()
.hover()
.mousedown()
.mouseenter()
.mouseleave()
.mousemove()
.mouseout()
.mouseover()
.mouseup()
.toggle()

Event Object

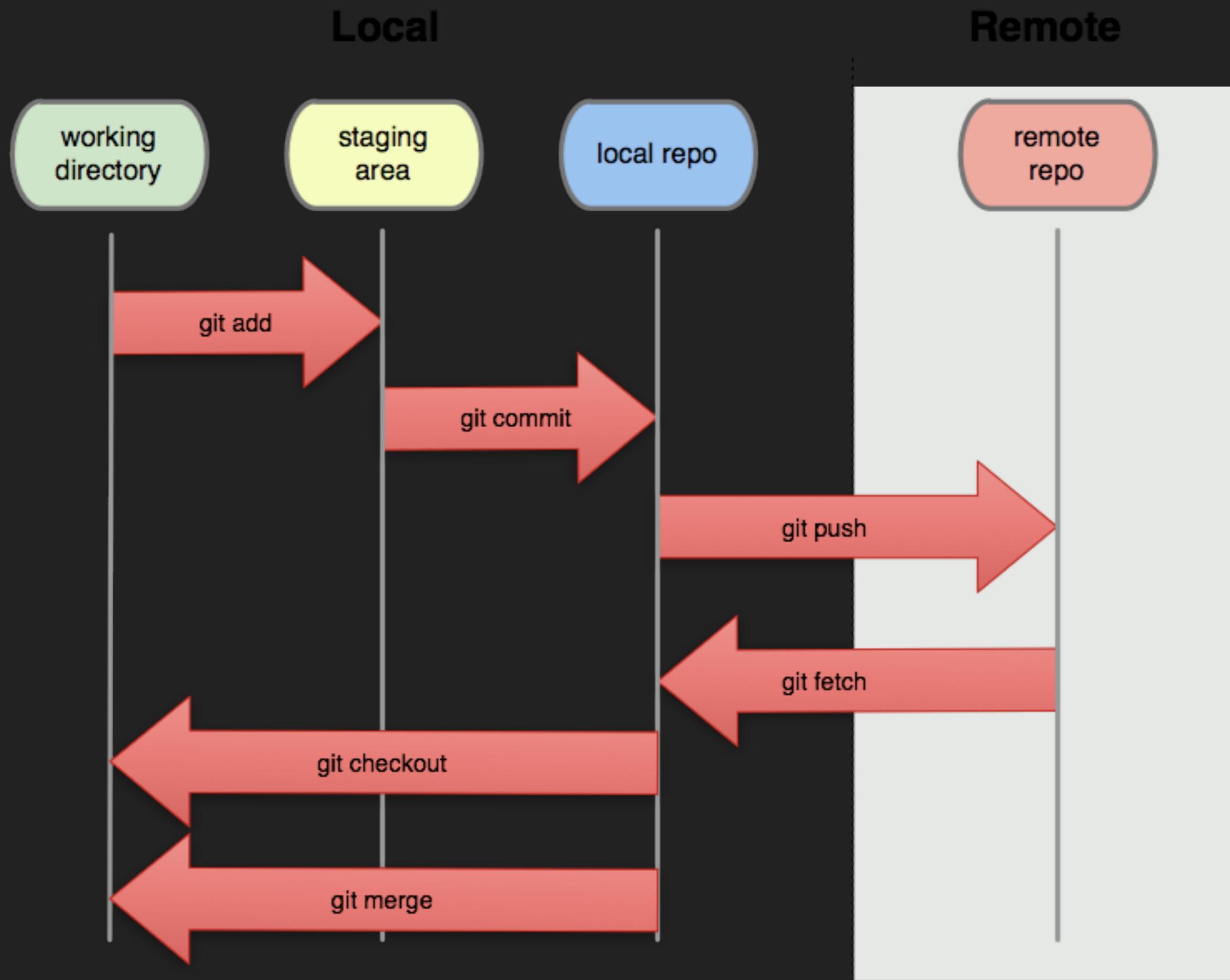
event.currentTarget
event.data
event.isDefaultPrevented()
event.isImmediatePropagationStopped()
event.isPropagationStopped()
event.namespace
event.pageX
event.pageY
event.preventDefault()
event.relatedTarget
event.result
event.stopImmediatePropagation()
event.stopPropagation()
event.target
event.timeStamp
event.type
event.which

GIT

- ▶ 형상관리도구, Configuration Management Tool
 - ▶ SourceCode 변경 history를 넘나드는 타임머신
 - ▶ 분산 / 병렬 개발의 필수 요소
-
- ▶ Repository / Clone
 - ▶ Branch / Checkout
 - ▶ Commit / Push
 - ▶ Tag
 - ▶ Fetch / Pull
 - ▶ Merge



GIT



GIT DOWNLOAD & CLONE

- ▶ git 설치
 - ▶ <https://git-scm.com/download/win>
 - ▶ <http://git-scm.com/download/mac>
- ▶ repository clone
 - ▶ git clone <http://<git-repository-url>>

GIT USAGE (WEBSTORM TIP)

- ▶ git init / remote
- ▶ git clone
- ▶ git checkout / branch: (우측하단 Git Menu)
- ▶ git rm / add (네비게이션 우클릭 > Git > Add)
- ▶ git commit / push (우측상단 VCS 위 화살표)
- ▶ git fetch / pull (CTRL+SHIFT+A > fetch or pull)
- ▶ git status / tag (ALT+9 > Log)

GIT 프로젝트 만들기

- ▶ git bash 실행 > cd 프로젝트 (해당 디렉토리로 이동)
- ▶ git init (초기화)
- ▶ git add *
- ▶ git status (붉은 색으로 표시되는 파일은 일일이 다시 add)
- ▶ git remote add origin <server_url>
- ▶ git push origin master

NEW PROJECT PROCESS

PREPARE

- ▶ Meteor.js 설치: <http://docs.meteor.com/>
- ▶ Git 설치: <https://git-scm.com/download/>
- ▶ WebStorm 설치: <https://www.jetbrains.com/webstorm/download/>

MAKE PROJECT

- ▶ Terminal 실행 > cd <원하는 디렉토리>
- ▶ meteor create <프로젝트명>
- ▶ WebStorm 실행 > File > Open > 생성된 프로젝트 디렉토리 오픈
- ▶ Open Terminal > meteor add iron:router
- ▶ 기본 <프로젝트명>.html / js / css 삭제
- ▶ client / public / server / lib 디렉토리 생성

CLIENT

- ▶ layout.html: 기본 틀 생성
 - ▶ <template name='layout'>
 - ▶ {{> yield}} : 동적 화면 추가
- ▶ layout.js: Iron Router 기본 틀 지정
 - ▶ Router.configure({
 layoutTemplate: 'layout'
});
- ▶ <filename>.html: 첫번째 화면 추가
 - ▶ <template name='filename'>
- ▶ <filename>.js: 첫번째 화면 Router 등록
 - ▶ Router.route('/', 'filename');

SERVER

- ▶ Server 기능 필요시 구현
- ▶ DB 제어
- ▶ REST API
- ▶ Publication
- ▶ 본 과정에서 다루지 않음

- ▶ 브라우저에 정적(Static) 파일 전송
 - ▶ html
 - ▶ images (jpg, png, gif...)
 - ▶ video (mpeg, mov...)
 - ▶ audio (mp3, ogg...)
 - ▶ 기타 모든 파일 그대로 브라우저에 전달 후 다운로드

- ▶ collections.js: MongoDB 콜렉션 생성
- ▶ <변수명> = new Mongo.Collection('DB명')
 - ▶ 해당 DB명으로 실제 DB 생성
 - ▶ 해당 변수명으로 Client/Server에서 DB 접근 가능
 - ▶ 보통 동일하게 설정

- ▶ `.find({조건}, {옵션}).fetch():` 조건에 맞는 여러 데이터를 array 형태로 반환
- ▶ `.find({'이름': '홍길동'}):` key가 '이름'이고 value가 '홍길동'인 전체 데이터를 반환하는 조건
- ▶ `.find({}, {sort: {createdAt: -1}}):` 전체 데이터를 createdAt에 대해서 역순으로 정렬하여 반환하는 옵션
- ▶ 조건과 옵션의 조합으로 다양하게 활용
- ▶ `.findOne({조건}, {옵션}):` 해당하는 가장 첫번째 데이터 반환

- ▶ `.remove({조건})`: 조건에 해당하는 전체 데이터 삭제
 - ▶ `.remove({})`: 전체 데이터를 삭제하므로 주의
- ▶ `.update({조건}, {Object})`: 해당 조건에 맞는 하나의 데이터를 Object로 변경
 - ▶ `.update({조건}, {Object}, {multi: true})` 해당 조건에 맞는 '모든' 데이터를 Object로 변경
 - ▶ update 된 Object의 갯수를 반환(return)
- ▶ `.insert({Object})`: 해당 Object를 해당 Collection에 insert
 - ▶ insert하며 생성된 `_id` 값을 반환(return)