



선제적 사업대응을 위한 *Software Tech Stack*

2018.07.10

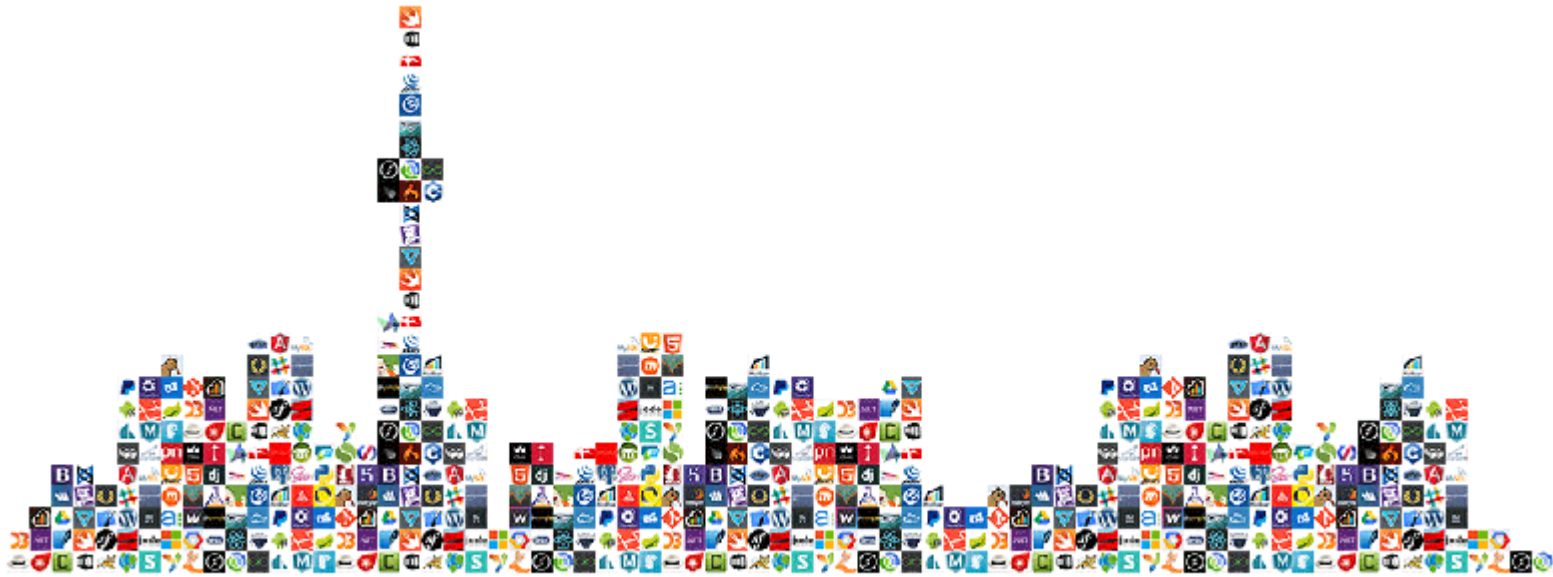
소프트웨어개발센터 박경규



목 차

1. Software Tech Stack 기술조사
2. 선제적 사업지원을 위한 Tech Stack
3. 개발생산성 비교

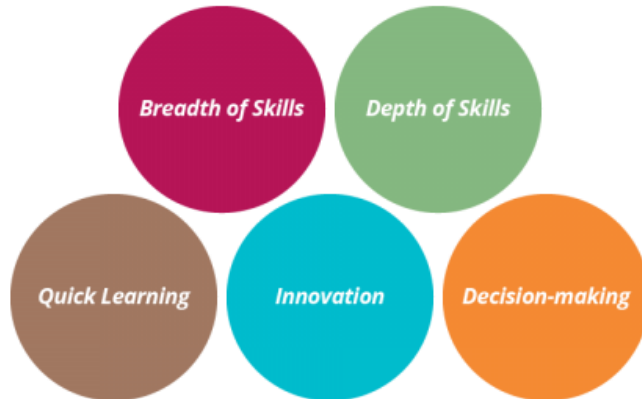
1. *Software Tech Stack* 기술조사



1.1 Tech Stack

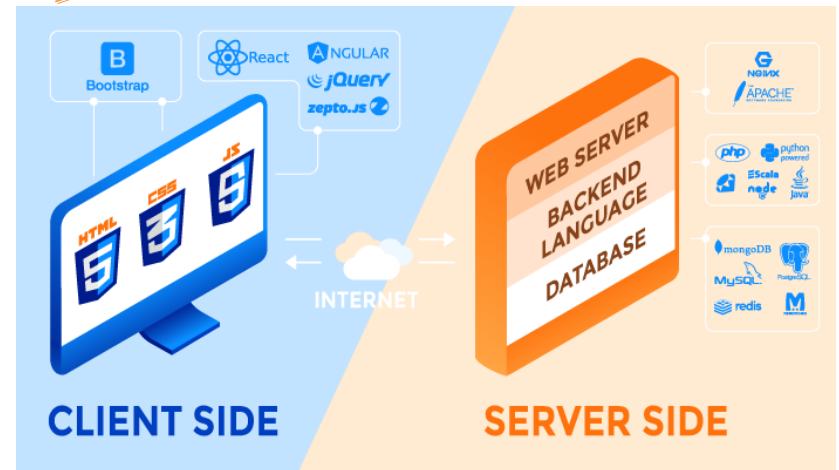
선제적 사업지원을 위한 소프트웨어의 핵심역량은 빠르게 반복할 수 있는 Software Delivery 역량구축과 Software Tech Stack 입니다.

THE FIVE KEY FACTORS IN BUILDING A WORLD-CLASS SOFTWARE DELIVERY CAPABILITY



[출처 : <https://www.thoughtworks.com/insights/blog/implications-tech-stack-complexity-executives/>]

Software Tech Stack



[출처 : <https://www.upwork.com/hiring/for-clients/how-to-choose-a-technology-stack-for-web-application-development/>]

FRAMEWORK	LANGUAGE
Ruby on Rails	Ruby
Django	Python
Node.js	Javascript
Laravel	PHP
.NET	C#
Spring	Java

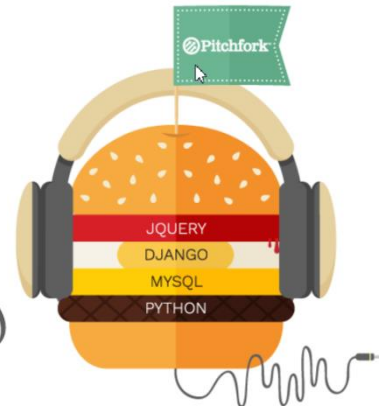
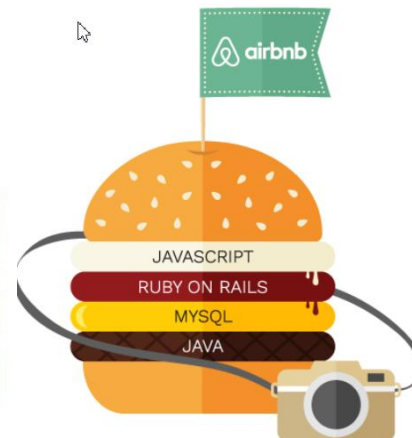
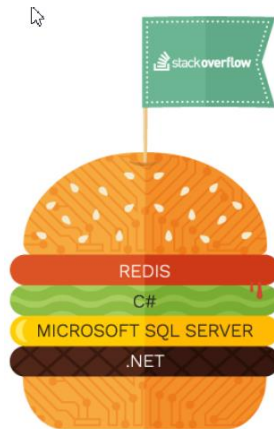
1.2 Full-Stack Development

풀스택개발자는 다양한 기술을 사용하여 제품의 프로토타입을 신속하게 만들 수 있으며, 어떤 영역에서도 투입가능하고 즉시 참여하는 마인드는 신규솔루션 출시에 매우 중요합니다.

Full-Stack Development: If software developers are “full-stack,” they are comfortable working with both back-end and front-end technology. This means that they have a specialized knowledge of all software development stages. These developers have the skills to work with various components of each type of development, including databases, HTML, JavaScript, CSS and more.

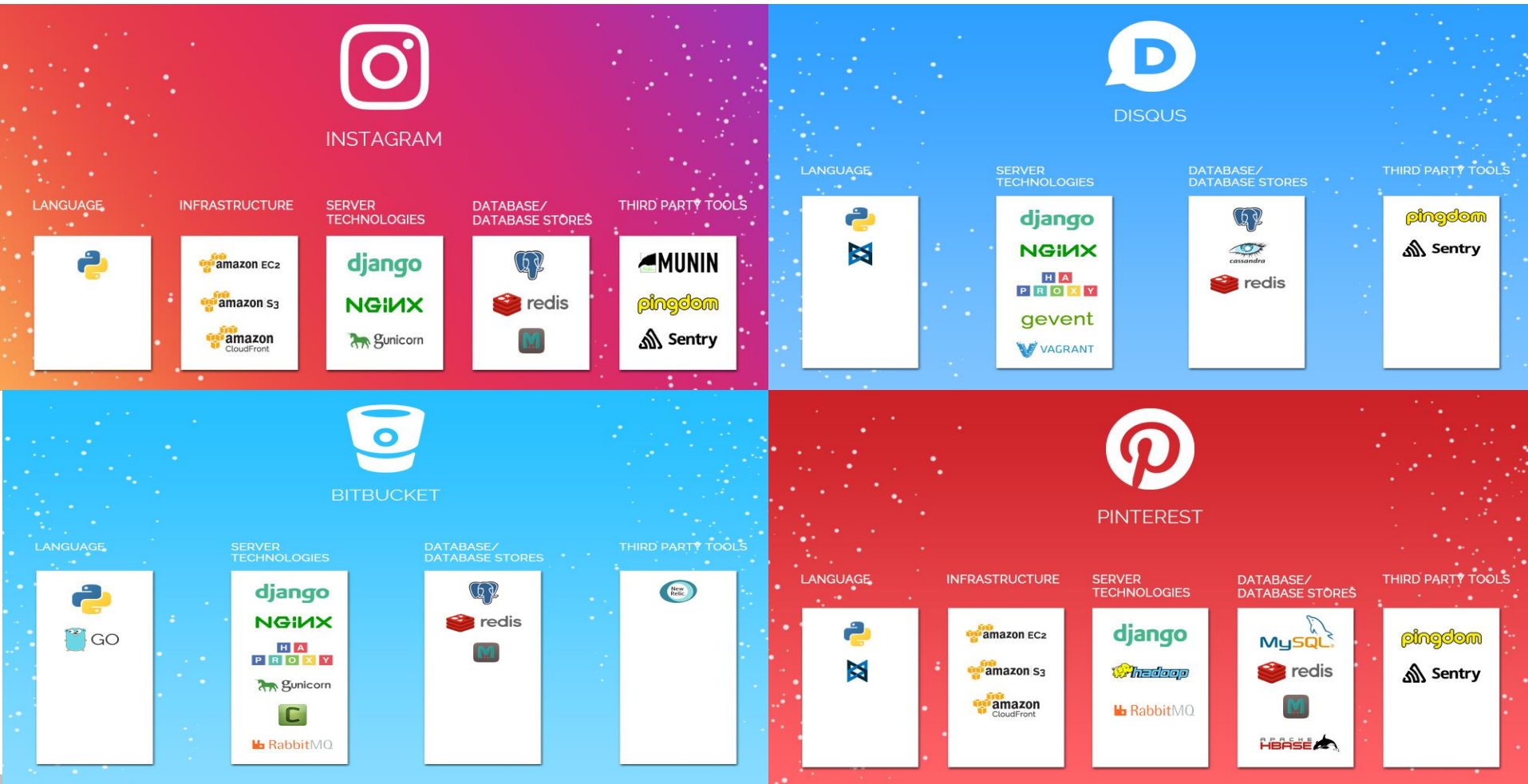


..... SIGNATURE STACKS



1.3 Software tech stack of hot startups

스타트업의 기술스택은 높은 생산성, 적절한 신뢰성, 개발자 고용 가능, 커뮤니티의 도움 특성을 가지고 있으며 최대한 빠르게 제품을 만드는 데 중점을 두고 있습니다.

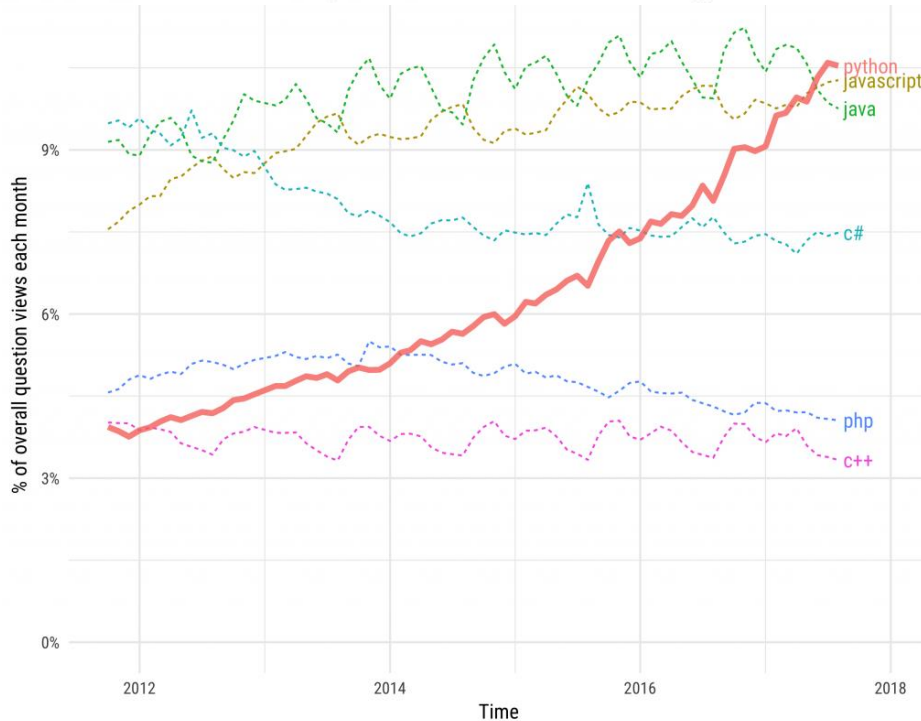


1.4 Programming Language-Python

Python은 가독성이 높고 강력한 개발언어로 개발자 저변이 계속 확장되고 있습니다.

Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries



[출처 : <https://stackoverflow.blog/2017/09/06/incredible-growth-python/>]

Python Has Amazing Libraries

- Scientific computing : NumPy, SciPy
- Web development : Django, Flask
- AI : Sci-kit Learn, Tensorflow

Corporate Sponsors

- C# : Microsoft
- Java : Oracle
- Python : Google

Less Code

Java

```
1 Class HelloWorld{
2     Public static void main(String[] args){
3         System.out.println("hello, World!");
4     }
5 }
```

Python

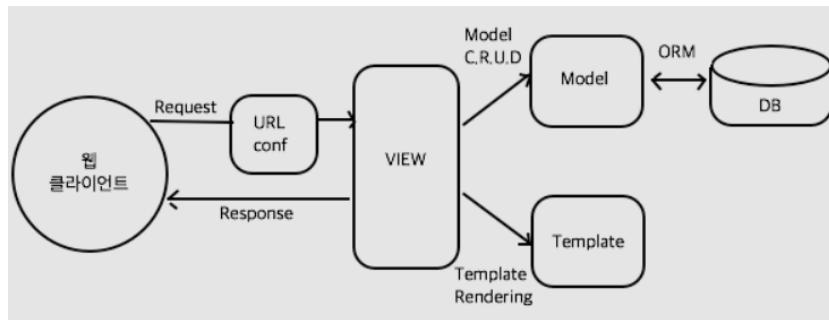
```
1 print("hello, World!")
```

1.5 Web development framework-Django

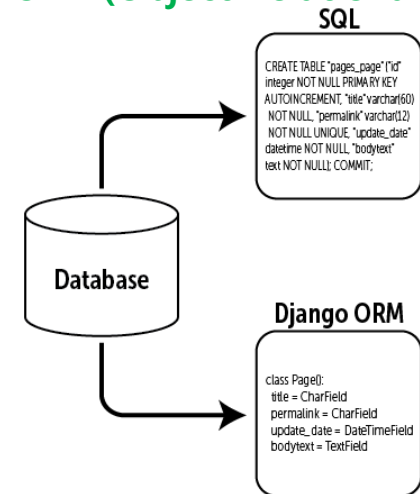
Django는 적은 양의 코드로 더 나은 App들을 만드는 웹개발 프레임워크입니다.



MTV(Model-Template-View) 아키텍처



ORM(Object-Relational Mapper)



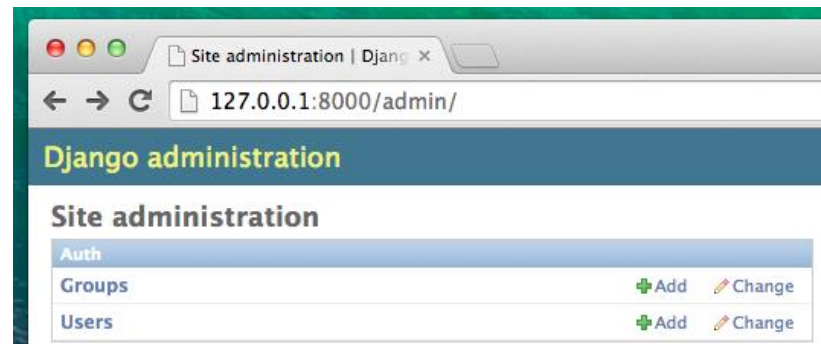
Quickly launch applications

- django-admin startproject **mysite**
- python manage.py startapp **polls**
- python manage.py runserver

```
mysite/
  manage.py
  mysite/
    __init__.py
    settings.py
    urls.py
    wsgi.py
```







```
polls/
  __init__.py
  admin.py
  apps.py
  migrations/
    __init__.py
  models.py
  tests.py
  views.py
```

Automatic admin interface



1.6 Third-Party Libraries

Python 라이브러리는 다른 개발언어가 따라올 수 없는 부분으로
AI 연구개발에서는 최전방에 위치하고 있습니다.









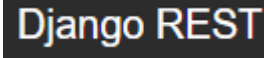




Library	설명
	웹 API를 구현하기 위한 강력하고 유연한 툴킷
Django Channels	웹소켓, IoT 프로토콜, 실시간 프로토콜을 다룰 수 있도록 확장하는 라이브러리
	행렬과 대규모 다차원 배열을 쉽게 처리 할 수 있게 해주는 라이브러리
	과학, 분석, 엔지니어링을 위한 과학(계산)적 컴퓨팅 영역의 작업을 위한 라이브러리
	머신러닝 교육 및 실무를 위한 Python패키지
	기계 학습과 딥러닝을 위해 구글에서 만든 오픈소스 라이브러리
	하이레벨 딥러닝 API로 대부분의 딥러닝 라이브러리 지원(TensorFlow, Theano, CNTK)

2. 선제적 사업지원을 위한 *Tech Stack*



2.1 New Biz. Solution Tech stack

신규사업 솔루션 개발과 융복합서비스 제공 니즈에 기민한 대응이 가능한 개발 프레임워크

Front-end UI	 Vue.js Progressive JavaScript Framework	 plotly Charting Library
Machine Learning Deep Learning	 scikit-learn Machine Learning	 TensorFlow Deep Learning  Keras A deep learning library
Data Analysis	 NumPy Base N-dimensional Array Package	 SciPy.org Scientific Computing Library  pandas $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$ High Performance Data Analysis Tool
Data Processing	 Django REST REST Framework	 Celery Distributed Task Queue  redis Message Broker
Web Framework	 django	
Language	 python ™	

2.2 핵심역량 강화 방안

기술변화가 가속화 되면서 신기술을 평가하고 채택하고 적응할 수 있는
개발조직으로 육성하는 것이 지속성장의 핵심방안입니다.

필요성

서비스에 요구되는 기술의 폭과 깊이가 넓어지고 있으며 최신 서비스에는 최신 기술이 필요
서비스 차별화와 성능개선이 갈수록 소프트웨어에 의존
증가하는 신사업에 신속한 대응과 개발생산성 향상을 위한 핵심역량 강화가 필요

핵심역량

빠르게 반복할 수 있는 Software Delivery 역량
어떤 영역에서도 투입 가능하고 즉시 참여할 수 있는 마인드셋
신기술을 평가하고 채택하고 적응할 수 있는 조직 역량

핵심역량 강화방안

선제적 사업지원이 가능하도록 풀스택 개발자 확보
개발 코딩량을 줄이게 하는 개발 프레임워크 활용
신기술을 평가하고 사업에 적용하는 선도 개발자 양성

3. 개발생산성 비교



3.1 개발 소요인력

일반(분야별 투입)

항 목		소요인원
PM	일정관리	1
프로젝트관리	개발기획/설계	1
	고객협의	1
백엔드 서비스 개발	서비스#1 개발	1
	서비스#2 개발	1
	서비스#3 개발	1
프론트엔드 개발	Design	1
	HTML/CSS	
	Javascript	1
게이트웨이 개발	TCP/IP	1
	MODBUS	
	BACnet	
합 계		9

풀스택(모든 분야 투입)

항 목		소요인원 (Sport mode)	소요인원 (Comport mode)
프로젝트관리	일정관리	0.2	0.4
	개발기획/설계		
	고객협의		
백엔드 서비스 개발	서비스#1 개발	0.8	1.6
	서비스#2 개발		
	서비스#3 개발		
프론트엔드 개발	Design	0.8	1.6
	HTML/CSS		
	Javascript		
게이트웨이 개발	TCP/IP	0.2	0.4
	MODBUS		
	BACnet		
합 계		2	4

3.2 생산성 비교 - Programming Language

Java

- verbose

```
public class HelloWorld
{
    public static void main (String[] args)
    {
        System.out.println("Hello!");
    }
}
```

- not compact

```
public Vector<Integer> aList = new Vector<Integer>;
public int    aNumber    = 5;
public int    anotherNumber;

aList.addElement(aNumber);
anotherNumber = aList.getElement(0);
```

```
public class Employee
{
    private String myEmployeeName;
    private int    myTaxDeductions = 1;
    private String myMaritalStatus = "single";

    //----- constructor #1 -----
    public Employee(String EmployeeName)
    {
        this(employeeName, 1);
    }

    //----- constructor #2 -----
    public Employee(String EmployeeName, int taxDeductions)
    {
        this(employeeName, taxDeductions, "single");
    }

    //----- constructor #3 -----
    public Employee(String EmployeeName,
                    int taxDeductions,
                    String maritalStatus)
    {
        this.employeeName = employeeName;
        this.taxDeductions = taxDeductions;
        this.maritalStatus = maritalStatus;
    }
}
```

Python

- concise

```
print("Hello, world!")
```

- compact

```
aList = []
aNumber = 5

aList.append(aNumber)
anotherNumber = aList[0]
```

```
class Employee():
    def __init__(self, employeeName, taxDeductions=1, maritalStatus="single"):
        self.employeeName = employeeName
        self.taxDeductions = taxDeductions
        self.maritalStatus
```

[출처 : <https://pythonconquerstheuniverse.wordpress.com/2009/10/03/python-java-a-side-by-side-comparison/>]

3.4 생산성 비교 - Framework

Spring(Java)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE sqlMap PUBLIC "-//ibatis.apache.org/DTD SQL Map 2.0//EN" "http://ibatis.apache.org/dtd/sql-map-2.dtd">

<sqlMap namespace="bems.stat.power">

  <typeAlias alias="hashMap" type="java.util.HashMap"/>
  <typeAlias alias="powerVO" type="com.kt.smcp.bems.power.vo.PowerVO"/>

  <select id="select-DAQHIST" parameterClass="hashMap" resultClass="powerVO">
    select ONE.DAQ_TIME DAQ_TIME,ONE.DEVICE_ID DEVICE_ID,ONE.TAG_ID TAG_ID,ONE.DAQ_VALUE
    DAQ_VALUE,DEVICE.DEVICE_NAME DEVICE_NAME,TAG.TAG_NAME TAG_NAME
    from TBR_PLF_DAHIST ONE,
    (SELECT DEVICE_ID, DEVICE_NAME
     FROM TBR_CNF_DEVICE) DEVICE,
    (select TAG_ID, TAG_NAME
     from TBR_CNF_TAG) TAG
    where ONE.DAQ_TYPE_CD = '10'
    and ONE.DAQ_TIME like #DATE# + '%'
    and ONE.DEVICE_ID = DEVICE.DEVICE_ID
    and ONE.TAG_ID = TAG.TAG_ID
    order by DAQ_VALUE desc
  </select>

</sqlMap>
```

MS-SQL

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE sqlMap PUBLIC "-//ibatis.apache.org/DTD SQL Map 2.0//EN" "http://ibatis.apache.org/dtd/sql-map-2.dtd">

<sqlMap namespace="bems.stat.power">

  <typeAlias alias="hashMap" type="java.util.HashMap"/>
  <typeAlias alias="powerVO" type="com.kt.smcp.bems.power.vo.PowerVO"/>

  <select id="select-DAQHIST" parameterClass="hashMap" resultClass="powerVO">
    /* bems.stat.power.select-DAQHIST */
    select ONE.DAQ_TIME DAQ_TIME,ONE.DEVICE_ID DEVICE_ID,ONE.TAG_ID TAG_ID,ONE.DAQ_VALUE
    DAQ_VALUE,DEVICE.DEVICE_NAME DEVICE_NAME,TAG.TAG_NAME TAG_NAME
    from TBR_PLF_DAHIST ONE,
    (SELECT DEVICE_ID, DEVICE_NAME
     FROM TBR_CNF_DEVICE) DEVICE,
    (select TAG_ID, TAG_NAME
     from TBR_CNF_TAG) TAG
    where ONE.DAQ_TYPE_CD = '10'
    and ONE.DAQ_TIME like concat(#DATE#, '%')
    and ONE.DEVICE_ID = DEVICE.DEVICE_ID
    and ONE.TAG_ID = TAG.TAG_ID
    order by DAQ_VALUE desc
  </select>

</sqlMap>
```

MySQL

- SQL문 사용으로 코딩량 많음
- SQL문 사용으로 데이터베이스 의존성이 높고
데이터베이스 변경시 재개발량 많음
- 데이터 관리기능을 별도로 제작해야 함

Django(Python)

```
qs = Energy.objects.values(
    'daq_time', 'p_usage', 'p_peak', 'p_usage_cost', 'p_unit_price').filter(
    site_id=site_id, gubun='15min', daq_time__range=(thisday, thisday + timedelta(days=1))
    ).order_by('daq_time')
df = pd.DataFrame(List(qs))
```

- SQL문을 사용하지 않아 코딩량 대폭 감소
- 데이터베이스 의존성이 없고 데이터베이스 변경이 자유로움
- 프레임워크가 데이터 관리기능을 기본 제공

3.3 생산성 비교 - JavaScript

Angular.js

```
export class ChangePasswordController {
  constructor($log, Auth, Notification) {
    'ngInject';

    this.$log = $log;
    this.Auth = Auth;
    this.Notification = Notification;
  }

  $onInit() {
    this.password = '';
  }

  changePassword() {
    this.Auth.changePassword(this.password).then(
      this.Notification.info('Password has been
    )).catch(error => {
      this.$log.error(error);
      this.Notification.error('There was an error. Please try again.');
```

```
import { Component } from '@angular/core';
import { Logger } from 'services/logger';
import { Auth } from 'services/auth';
import { Notification } from 'services/notification';
```

```
@Component({
  selector: 'change-password',
  templateUrl: './ChangePassword.component.html',
  styleUrls: ['./ChangePassword.component.scss'],
})
```

```
export class ChangePasswordComponent {
  password: string = '';
```

```
  constructor(
    private logger: Logger,
    private auth: Auth,
    private notification: Notification,
  ) {}
```

```
  changePassword() {
    this.auth.changePassword(this.password).subscribe(() => {
      this.notification.info('Password has been changes successfully');
    }).catch(error => {
      this.logger.error(error);
      this.notification.error('There was an error. Please try again');
    });
  }
}
```

Vue.js

```
import Vue from 'vue';
import Logger from 'utils/logger';
import Auth from 'actions/auth';
import Notification from 'utils/notification';

Vue.component('change-password', {
  template: '<div>{{ /* template */ }}</div>',
  data() {
    return {
      password: '',
    };
  },
  methods: {
    changePassword() {
      Auth.changePassword(this.state.password).then(() => {
        Notification.info('Password has been changed successfully.');
```

[출처 : <https://github.com/feimosi/frameworks-code-comparison/>]

감사합니다