



코딩 기초에서 서비스 런칭까지 4박 5일

제주 코딩 베이스캠프



이호준, 이범재, 유준모

발표자

Python, Django 트랙 이호준

(現) 바울랩아이씨티기술연구원 대표이사

(現) 바울랩아이씨티컴퓨터학원 대표이사

(現) 사도출판 대표이사

Javascript, JQuery 트랙 이범재

(現) 웹개발자

인터랙티브 HTML, CSS 트랙 유준모

(現) 스튜디오밀 대표

제주 코딩 베이스캠프

코딩 기초부터 기획, 서비스 배포까지 4박 5일. 88,000원. 우리는 단지 열정 있는 청년들에게 열정만큼 성장할 수 있는 기회를. 서비스 구축 성공경험을 통해 이 취업난 시대에 자신감, 자존감, 성취감, 자립할 수 있는 능력을 배양할 수 있도록. 또, 이러한 경험이 단순히 금액이 높다는 이유로 또 한번에 좌절이 되지 않도록. 이 행사를 진행합니다.



목차

1일차 오전

프리뷰 ----- 4

1일차 저녁

홈페이지 기획 ----- 32

2일차 오전

HTML, CSS ----- 44

2일차 저녁

Javascript & JQuery ----- 54

3일차 오전

Python ----- 87

4일차 오전

Django ----- 108

10월
3주차

오전

드로이블

Jeju coding basecamp

열정 있는 청년들에게 열정만큼 성장할 수 있는 기회를.
서비스 구축, 성공경험을 통해 이 취업난 시대에 자신감,
자존감, 성취감, 자립할 수 있는 능력을 배양할 수 있도록.
또, 이러한 경험이 단순히 금액이 높다는 이유로 또 한번
에 조절이 되지 않도록.

Facebook 그룹 : 제주 코딩 베이스캠프
캠프 주기 : 1년 2회

일정표

<공지용>

제주 코딩 베이스캠프

구 분	Front-end		Back-end		해커톤(자율) 금	
	월	화	수	목		
09 ~ 12	소개하는 시간 Avengers I 프리뷰 front-end back-end html css javascript python 개요 설명				Avengers I Python 문법 Avengers I Django 활용	구성원 팀원모집
점심시간 (2시간)						
14 ~ 17	생활코딩 동영상 시청 과제	Avengers I Bootstrap	Avengers I Python 문법	Avengers I Django 활용		
저녁시간 (2시간)						
19 ~ 22	Avengers III 홈페이지 기획 (늦게 오시는 분이 있을 시에는 개별 보강)	Avengers III Javascript&jQuery	자유시간		Avengers I, II, III 결과발표	해 커 토 진 행

부트스트랩 강의는 라이브 코딩입니다.

12인 토소 개발 (A2, A3 소개는 추후에 각각 별도로 예정입니다.)



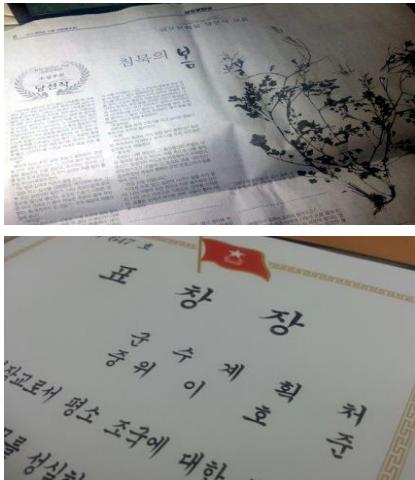
금오공과대학교 컴퓨터공학부

아주대학교 정보통신대학원 정보보안학과
정보통신장교 출신(학군)

00부대 CERT과장, 세미정비과장 역임
한국공학기술연구원 팀장

S전자 주니어소프트웨어 아카데미 메인 강사
신한금융그룹 신한데이터시스템즈(2개 회사) 정보보안 담당
제주코딩해커톤 등 다양한 사회공헌 활동 주최
현) Paul Lab ICT 컴퓨터학원 대표이사
현) Paul Lab ICT 기술연구원 대표이사
현) 사도출판 대표이사

A1 이호준 010-5044-2903



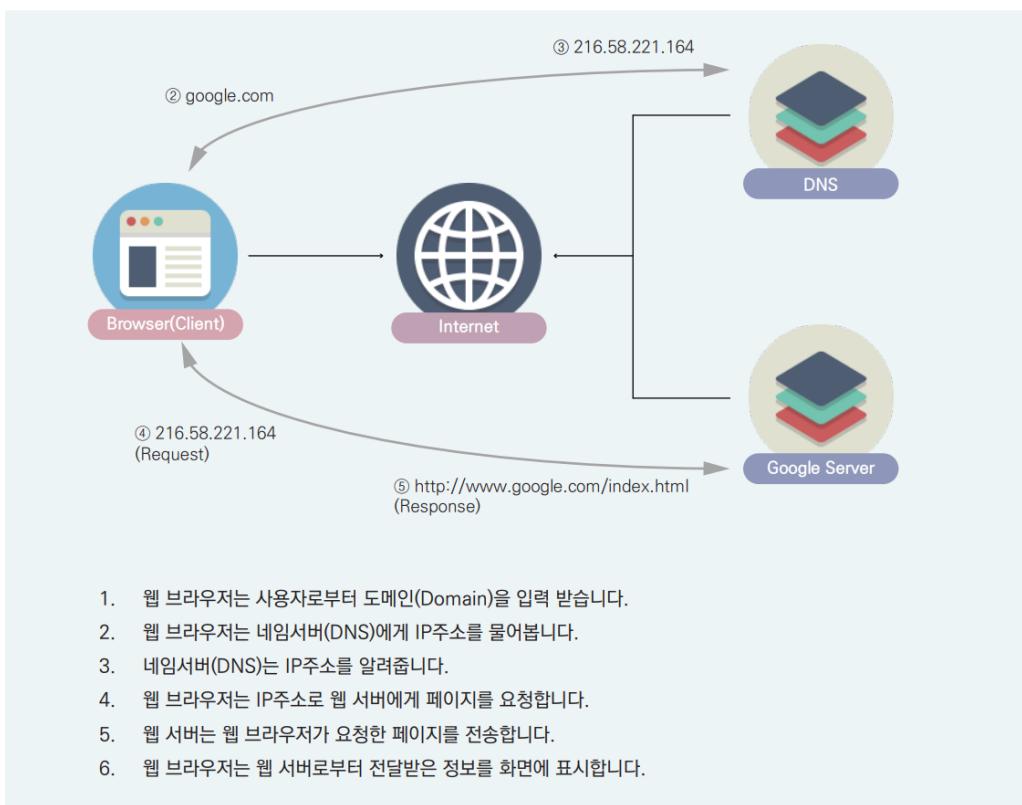


소통해요

1. 어디서 오셨나요?
2. 기대하는 것은 무엇인가요?
3. 어떤 서비스를 만들고 싶으신가요?

인터넷은 어떻게 구성되어 있나?

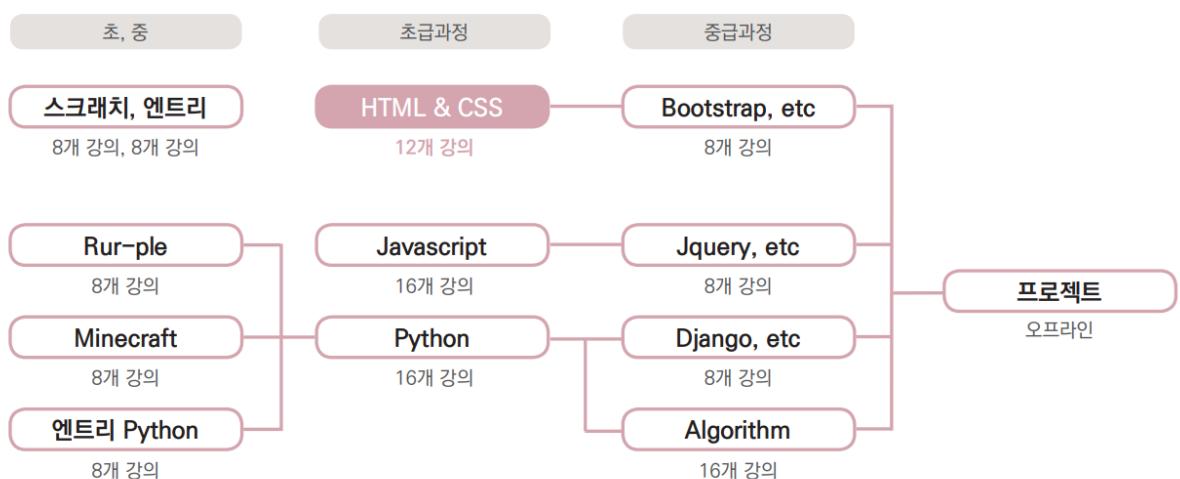
이미지 - 튜토리얼로 배우는 HTML & CSS
디자인 - 김미경 디자인



1. 웹 브라우저는 사용자로부터 도메인(Domain)을 입력 받습니다.
2. 웹 브라우저는 네임서버(DNS)에게 IP주소를 물어봅니다.
3. 네임서버(DNS)는 IP주소를 알려줍니다.
4. 웹 브라우저는 IP주소로 웹 서버에게 페이지를 요청합니다.
5. 웹 서버는 웹 브라우저가 요청한 페이지를 전송합니다.
6. 웹 브라우저는 웹 서버로부터 전달받은 정보를 화면에 표시합니다.



무엇을 어떻게 이용하여 개발할 것인가?



이미지 - 투토리얼을 배우는 HTML & CSS

디자인 - 컴퓨터 디자인

어디에서 개발할 것인가?

클라우드 서비스란?

1. SaaS(Software as a Service)

SaaS는 가장 사용자 단위로 친밀한 서비스이며 네트워크를 통해 애플리케이션 기능을 이용할 수 있는 서비스입니다.

2. PaaS(Platform as a Service)

PaaS는 윈도우와 리눅스 같은 운영체제를 제공하고 개발 가능한 플랫폼도 함께 제공되는 클라우드 서비스입니다.

3. IaaS(Infrastructure as a Service)

IaaS는 인프라를 제공하는 클라우드 서비스입니다. 기업에서 특히 많이 쓰입니다.

프로젝트는 어떻게 설계할 것인가?

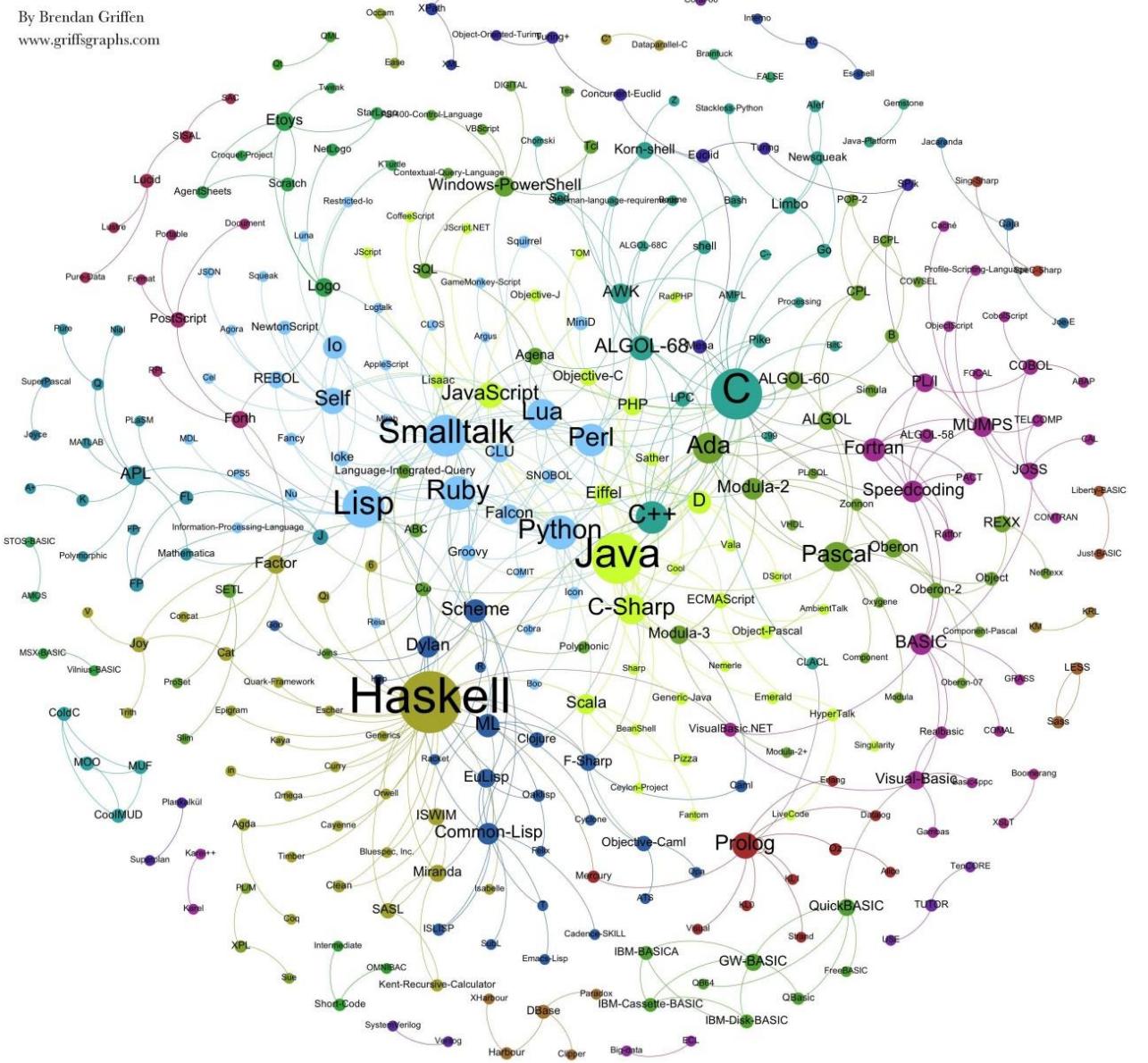
1. 초기가한 해커톤에서 완성 홈페이지 시연
 - 시연
 - usecase
 - 프로젝트 문서
2. 구름 IDE 세팅, 환경설정
3. 구름 IDE에서 HTML, CSS, Javascript, Python 훑어보기.

WHY IS IT ORGANIZED?

The Graph Of Programming Languages

By Brendan Griffen

www.griffsgraphs.com



<https://namu.wiki/w/%EB%82%98%ED%8A%8D%EC%84%A4%EC%84%9C%ED%83%80%EB%82%98>

와H 표이시건인가?

수 많은 언어 중 단연 주목받는 언어

직관적이고 배우기 쉬움

여러 Framework(Django, Flask 등)을 사용하여 개발을 쉽게 할 수 있음.

해커의 언어라고도 불리움(라이브러리가 워낙 많다보니 할 수 있는 것들이 많습니다.)

와H Django인가?

Popular Full-Stack Frameworks

A web application may use a combination of a base HTTP application server, a storage mechanism such as a database, a template engine, a request dispatcher, an authentication module and an AJAX toolkit. These can be individual components or be provided together in a high-level framework.

These are the most popular high-level frameworks. Many of them include components listed on the [WebComponents](#) page.

Name	Latest version	Latest update date	description
 Django	1.11	2017-04-04	The Web framework for perfectionists (with deadlines). Django makes it easier to build better Web apps more quickly and with less code. Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. It lets you build high-performing, elegant Web applications quickly. Django focuses on automating as much as possible and adhering to the DRY (Don't Repeat Yourself) principle. See Django
 TurboGears	2.3.10	2016-12-04	the rapid Web development webframework you've been looking for. Combines  SQLAlchemy (Model) or  Ming (MongoDB Model),  Genshi (View),  Repoze and  Tosca Widgets. Create a database-driven, ready-to-extend application in minutes. All with designer friendly templates, easy AJAX on the browser side and on the server side, with an incredibly powerful and flexible Object Relational Mapper (ORM), and with code that is as natural as writing a function. After reviewing the  Documentation, check out  the Tutorials
 web2py	2.14.6	2016-05-10	* Python 2.6 to 2.7, Python 3.x friendly (compile but not tested no support yet) * All in one package with no further dependencies. Development, deployment, debugging, testing, database administration and maintenance of applications can be done via the provided web interface, but not required. * web2py has no configuration files, requires no installation, can be run off a USB drive. * web2py uses Python for the Model, View and the Controller * Built-in ticketing system to manage errors * Internationalization engine and plurals, caching system * Flexible authentication system (LDAP, MySQL, Janrain etc) * NIX(Linux, BSD), Windows, Mac OSX, tested on EC2, Webfaction * works with MySQL, PostgreSQL, SQLite, Firebird, Oracle, MSSQL and the Google App Engine via an ORM abstraction layer. * Includes libraries to handle HTML/XML, RSS, ATOM, CSV, RTF, JSON, AJAX, XMLRPC, WIKI markup. * Production ready, capable of upload/download of very large files * Emphasis on backward compatibility.

<https://wiki.python.org/moin/WebFrameworks>

Full-Stack Framework로 Front-end와 Back-end까지 다 경험해본다.

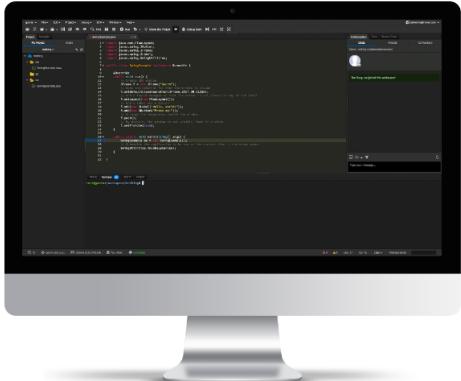
약간(?) 무거지만 뭐든지 만들 수 있다.

Flask도 많이 쓰지만 이외 대체는 Django!

Bootstrap, Django

- Framework에 대한 것
- Bootstrap, Django에 대한 것

구름IDE 환경설정 (<https://www.goorm.io/>)



구름IDE 환경설정

구름IDE 구름EDU 구름TEST

로그인 회원가입 도움말

클라우드 통합 개발 환경

구름IDE

구름IDE를 이용하면 언제 어디서나 효과적인 소프트웨어 개발이 가능합니다. 가장 진화한 웹기술 기반의 통합 개발 환경을 만나보세요. 컨테이너 기반의 다양한 SW개발 환경을 원클릭으로 구성하여 환경구축과 배포에 걸리는 시간을 절약하고, 실시간 소스 코드 동시 편집 기능을 활용하여 효율적인 소프트웨어 개발이 가능합니다. 미래의 소프트웨어 개발 환경이 지금 바로 여러분 앞에 도착해있습니다.

- 클라우드 기반으로 언제 어디서나 개발이 가능합니다.
- 실시간 채팅으로 효율적인 개발이 가능합니다.
- 동시 소스 코드 편집으로 업무 효율이 상승합니다.

구름IDE로 이동



구름IDE 서비스 플랜 대시보드

로그인 회원가입 도움말

세상에서 가장 가벼운 컨테이너에
타오르는 아이디어를 쓸아내세요.

국내 유일 클라우드 통합 개발 환경 구름IDE

구름IDE는 코딩/빌드/컴파일/공유가 가능한 독립가상환경, 컨테이너를 제공합니다.

언제 어디서나 컨테이너에 아이디어를 채우고 쌓아 세상에 없던 서비스를 만들어보세요!

구름 IDE 환경 설정 (<https://ide.goorm.io/>)

구름 회원가입

하나의 아이디로 구름의 다양한 서비스를 이용해 보세요.

이호준

paul-lab@naver.com

강함

구름 서비스에 대한 소식을 이메일로 받아봅니다.

구름에서 제공하는 서비스 약관에 동의합니다. [약관 보기](#)

가입하기

구름 IDE 환경 설정 (<https://ide.goorm.io/>)

구름에 가입해주셔서 감사합니다.

구름에 등록한 메일주소가 올바른지 확인하기 위한 메일입니다.
메일등록을 완료하시려면 아래 버튼을 누르세요.

[이메일 주소 확인](#)



이메일

비밀번호

로그인

이메일 저장

구름IDE환경설정(https://ide.goorm.io/)

클라우드 코딩 서비스 구름IDE



[바로가기](#)

클라우드 SW 교육 환경 구름EDU



[바로가기](#)

클라우드 코딩테스트 구름TEST



[바로가기](#)



[서비스 플랜](#)

[대시보드](#)

내 컨테이너 리스트(0개)



[새 컨테이너 생성](#)

구름 IDE 환경 설정 (<https://ide.goorm.io/>)

새 컨테이너 생성

이름	paullab														
설명	컨테이너 설명을 입력해주세요.														
소스	<input checked="" type="radio"/> 템플릿 <input type="radio"/> SCM <input type="radio"/> 암축파일														
 C/C++  JAVA  Node.js  Python  Ruby  JSP															
 PHP  .NET  Go  Arduino															
소프트웨어 스택 선택	<p>Python</p> <table border="1"><tr><td>프로젝트 유형</td><td>Python 빈 프로젝트</td></tr><tr><td>OS</td><td>✓ Ubuntu 14.04 LTS</td></tr><tr><td>Python</td><td>✓ 3.4.3</td></tr><tr><td>Python</td><td>✓ 2.7.6</td></tr><tr><td>pip</td><td>✓ 1.5.4</td></tr><tr><td>Django</td><td>✓ 1.8.17 LTS</td></tr><tr><td>Flask</td><td>✓ 0.12</td></tr></table>	프로젝트 유형	Python 빈 프로젝트	OS	✓ Ubuntu 14.04 LTS	Python	✓ 3.4.3	Python	✓ 2.7.6	pip	✓ 1.5.4	Django	✓ 1.8.17 LTS	Flask	✓ 0.12
프로젝트 유형	Python 빈 프로젝트														
OS	✓ Ubuntu 14.04 LTS														
Python	✓ 3.4.3														
Python	✓ 2.7.6														
pip	✓ 1.5.4														
Django	✓ 1.8.17 LTS														
Flask	✓ 0.12														
<button>취소</button> <button>생성하기</button>															

구름 IDE 환경 설정 (<https://ide.goorm.io/>)



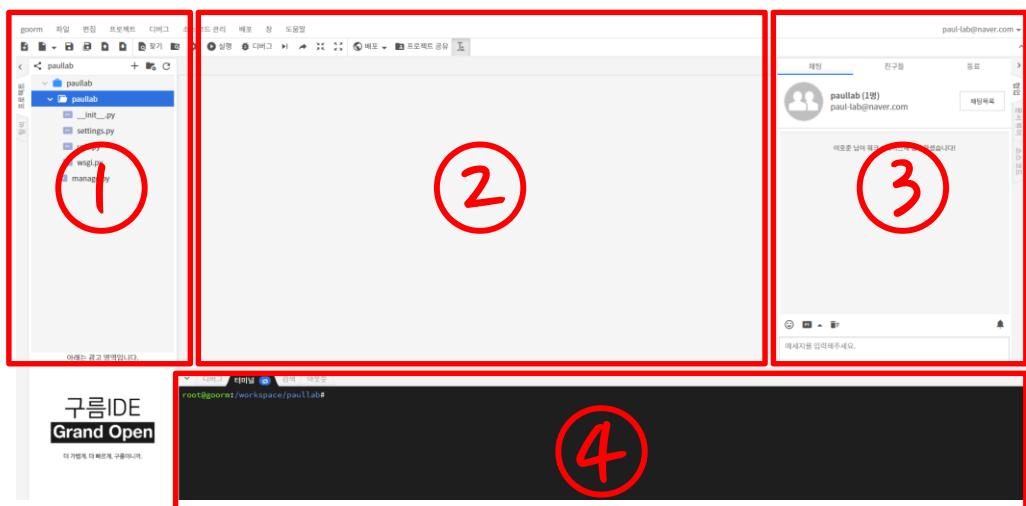
구름 IDE 환경 설정 (<https://ide.goorm.io/>)

The screenshot shows the Goorm IDE environment configuration page for a user named 'paullab'. The page includes the following details:

- Software Stack: paullab
- Storage Space: Python 5GB
- Always On: Softwarerack
- Usage Note: Use for purchasing premium products.
- Action Buttons: 실행 (Execution) and 터미널 실행 (Terminal Execution). The '실행' button is highlighted with a red box.

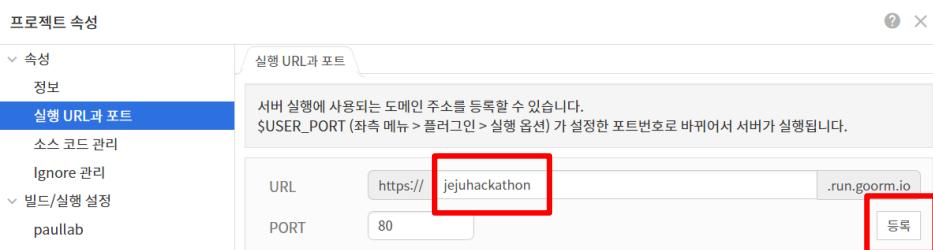
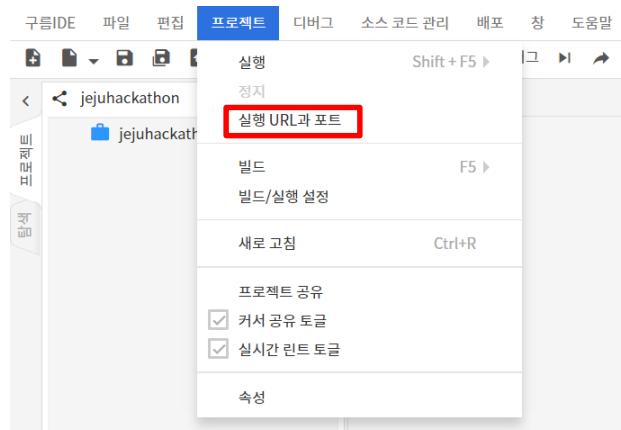
On the right side, there is a large 'goormide' logo and a progress bar indicating '컨테이너 로딩 중입니다.' (Container loading in progress).

구름IDE 환경설정 (<https://ide.goorm.io/>)



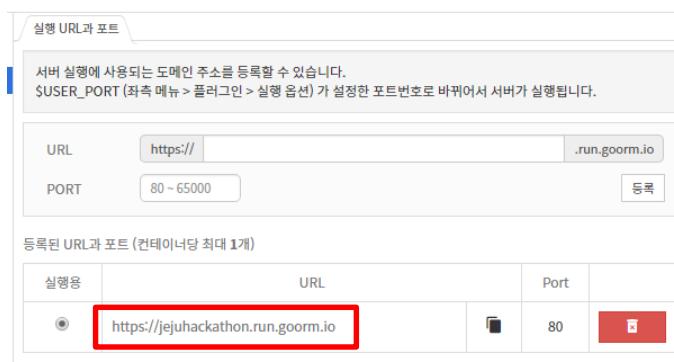
1. 풀더 구조를 볼 수 있는 공간입니다. 우측 상단에 보시면 새로고침 버튼이 있어요. 원가 작업을 했는데 보이지 않는다면 새로고침을 해보시기 바랍니다.
2. Txt, html, py파일 등 다양한 파일을 edit 할 수 있는 공간입니다.
3. 험언을 위한 공간이에요. 이번 해커톤에서는 사용하지 않습니다.
4. 콘솔창입니다. 그동안 GUI환경에만 익숙해 계시겠죠? 전 세계의 서버 대부분은 리눅스입니다. CLI 환경에 익숙해 지시기를 권해드려요. 모든 명령어는 이곳에 입력합니다. 우측클릭 후 '입력상자 띄우기' 를 하시면 붙여넣기가 가능합니다.

구름 IDE 환경 설정 (<https://ide.goorm.io/>)

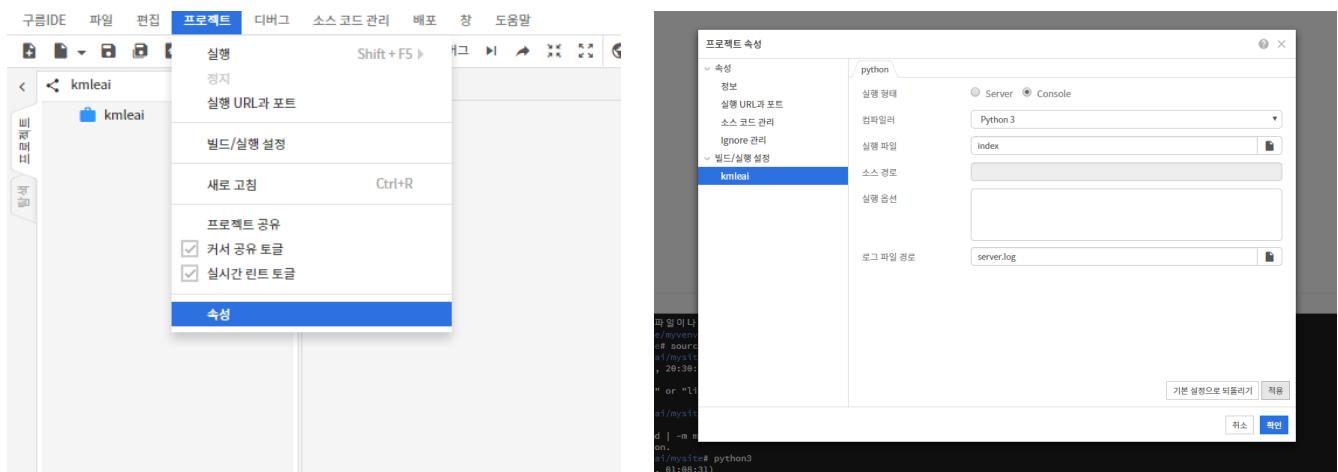


URL은 원하시는 이름으로 작성해주세요.

구글 IDE 환경 설정 (<https://ide.goorm.io/>)



• 노트북으로 사용하는 URL입니다.



Python 버전을 맞추기 위해 프로젝트 > 속성을 클릭합니다.

컴파일러를 클릭하고 Python version을 3으로 맞춰줍니다.
다. 적용버튼을 누른 후 확인버튼을 클릭하세요.

구름IDE 환경설정 (<https://ide.goorm.io/>)

The screenshot shows the Goorm Cloud IDE interface. On the left, a sidebar menu is open under the '구름IDE' tab, with '컨테이너 나가기' selected. Below it are '로그아웃' and a file named 'manitest.vml'. At the bottom of the sidebar, there is a message: '한번 실행해야 하므로 컨테이너를 나가습니다.' (You need to run it once, so please exit the container). On the right, a project card for 'kmleai' is displayed, updated on '2017. 8. 9.'. It shows '2명과 공유' (Shared with 2 people), '소프트웨어 스택 Python', '저장 공간 5GB', and '항상 켜두기 PREMIUM 상품 구매 시 사용 가능' (Always on, available when purchasing a PREMIUM product). At the bottom right of the card are '실행' (Run) and '터미널 실행' (Terminal Run) buttons.

구글 IDE 환경 설정 (<https://ide.goorm.io/>)

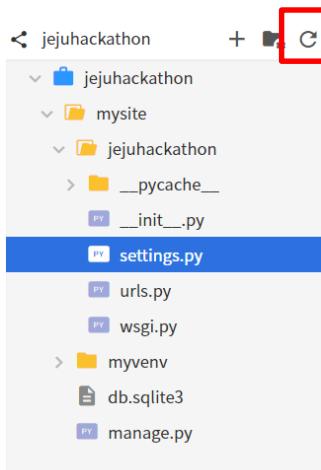
```
root@goorm:/workspace/여러분ID# python --version  
root@goorm:/workspace/여러분ID# mkdir web  
root@goorm:/workspace/여러분ID# cd web  
root@goorm:/workspace/여러분ID/web# sudo apt-get update  
root@goorm:/workspace/여러분ID/web# sudo apt-get install nginx  
root@goorm:/workspace/여러분ID/web# vi /etc/nginx/sites-available/default
```

vi 에디터를 이용하여 root /var/www/html; 를 root /workspace/구글아이디/web; 로 고침세요.
(i를 누르면 편집이 됩니다. 편집을 다 하신 후에는 ESC, :, wq!를 순서대로 입력하세요.)

```
root@goorm:/workspace/여러분ID/web# sudo service nginx start
```

멈추고 싶을 때에는 sudo service nginx stop를 입력합니다.

구글 IDE 환경 설정 (<https://ide.goorm.io/>)



작성하신 파일이 안보이실 때에는 새로고침 버튼을 클릭해주세요.

HTML, CSS, Javascript, Python 훑어보기.

- 구글 IDE에서 HTML, CSS, Javascript, Python 실행하고 훑어보기

HTML, CSS, Javascript, Python 헷갈리지!

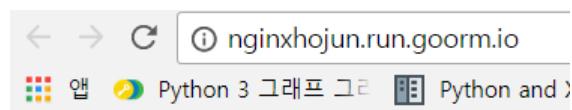
The screenshot shows a code editor interface. On the left, there is a file tree with a project named 'nginxhojun'. Inside 'nginxhojun' is a folder 'web' containing 'index.html' and 'index.py'. To the right of the file tree is a code editor window titled 'index.html - nginxhojun/mys...'. The code in the editor is:

```
1 <html>
2 <head>
3 </head>
4 <body>
5     <h1>test</h1>
6 </body>
7 </html>
8
```

HTML, CSS, Javascript, Python 헷갈리지!

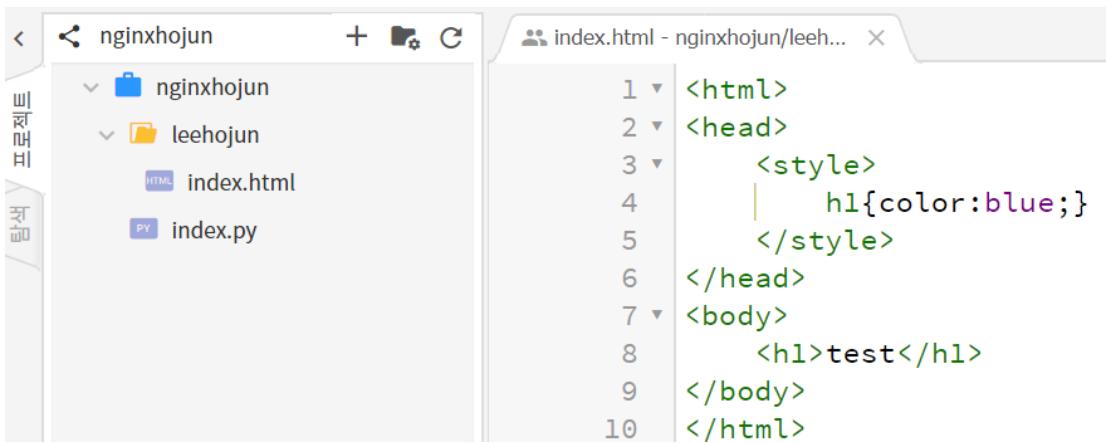
The screenshot shows a code editor interface. On the left, there is a file tree with a project named 'nginxhojun'. Inside 'nginxhojun' is a folder 'web' containing 'index.html' and 'index.py'. To the right of the file tree is a code editor window titled 'index.html - nginxhojun/mys...'. The code in the editor is:

```
1 <html>
2 <head>
3 </head>
4 <body>
5     <h1>test</h1>
6 </body>
7 </html>
8
```



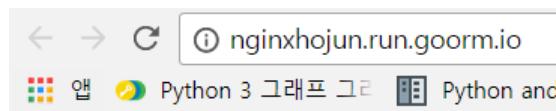
test

HTML, CSS, Javascript, Python 풀어짰지!



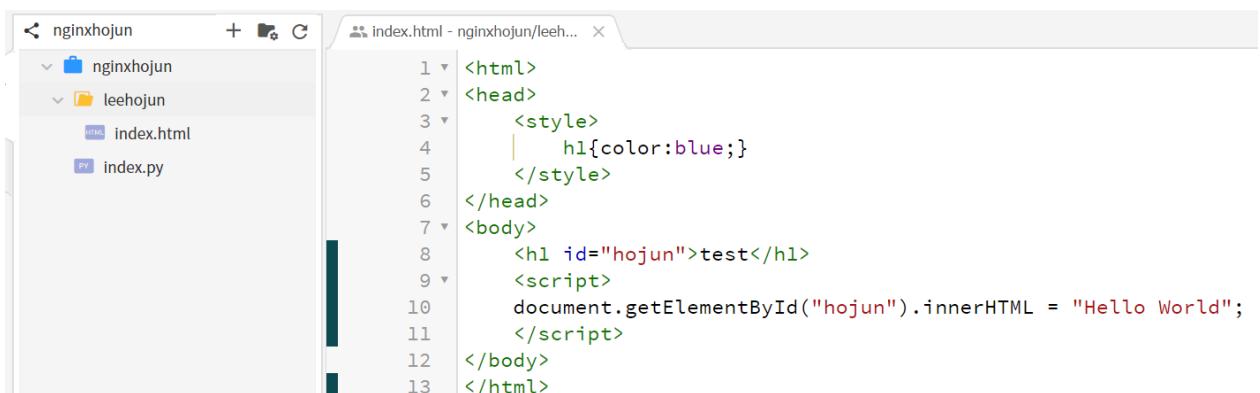
```
nginxhojun
+ nginxhojun
  - leehojun
    + index.html
    + index.py
```

```
index.html - nginxhojun/leeh...
1 <html>
2 <head>
3   <style>
4     h1{color:blue;}
5   </style>
6 </head>
7 <body>
8   <h1>test</h1>
9 </body>
10 </html>
```



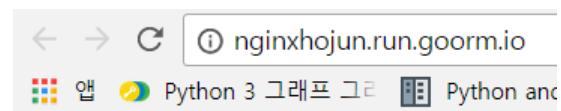
test

HTML, CSS, Javascript, Python 풀어짰지!



```
nginxhojun
+ nginxhojun
  - leehojun
    + index.html
    + index.py
```

```
index.html - nginxhojun/leeh...
1 <html>
2 <head>
3   <style>
4     h1{color:blue;}
5   </style>
6 </head>
7 <body>
8   <h1 id="hojun">test</h1>
9   <script>
10    document.getElementById("hojun").innerHTML = "Hello World";
11   </script>
12 </body>
13 </html>
```



Hello World

일상

저녁

홈페인팅 기획

강사소개

이름 : 이병재

- (주)제주넷 웹개발자
- 경력
 - 배움커뮤니티 - Ruby on Rails
 - 온라인시험솔루션회사 - classic ASP
 - 무선공중전화기 개발회사 - JSP

홈페이지 기획하기

왜 웹사이트를 만드는가?

핵심요소는 무엇인가?

누가 이 홈페이지를 방문하는가?

기능들은 무엇이 있는가?

오늘 할 것은

freemind를 이용한 기능목록 작성

oven을 이용한 와이어프레임 그리기

freemind

전체 이미지 동영상 뉴스 도서 더보기 설정 도구

검색결과 약 2,460,000개 (0.62초)

[FreeMind: Main Page](#)
freemind.sourceforge.net/ ▾ 이 페이지 번역하기
FreeMind is a premier free mind-mapping software written in Java. The recent development has hopefully turned it into high productivity tool. We are proud that ...

[Download](#)
Asked questions - Files - 1.0.0 -
FreeMind on Linux - ...

[FreeMind Version 1.1.0 Beta 2](#)
Dear FreeMind users,, the second
beta of 1.1.0 is available ...

운영체제에 맞는 것을 다운로드 해주세요.

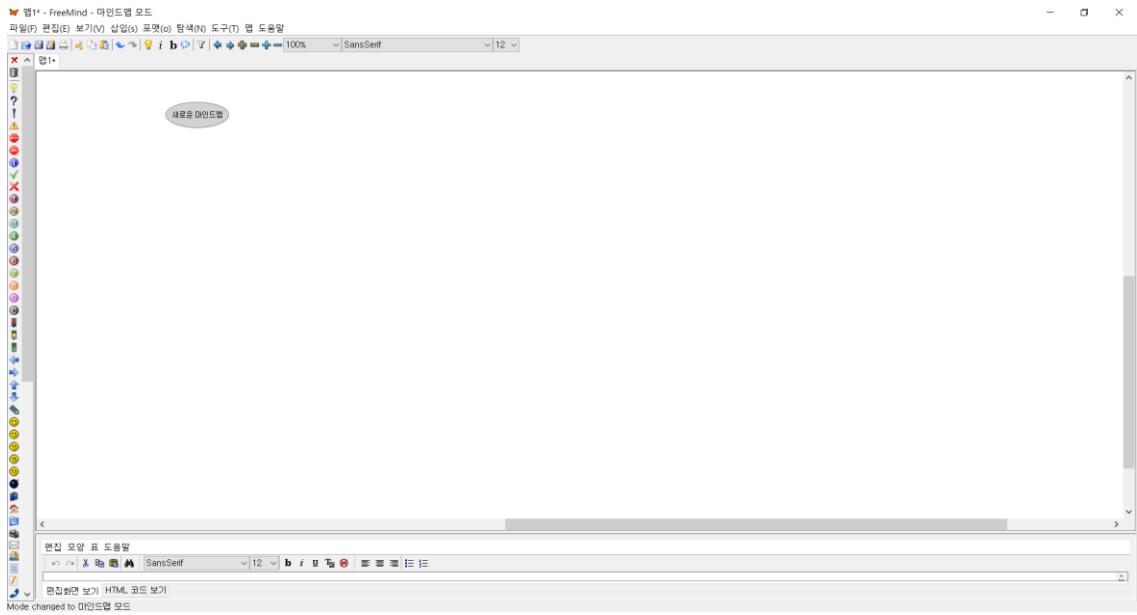
oven

전체 이미지 지도 뉴스 동영상 더보기 설정 도구

검색결과 약 449,000,000개 (0.53초)

[카카오 오븐](#)
<https://ovenapp.io/> ▾
HTML5-Powered Free Web/App Prototyping Tool by Kakao.

접속해서 가입해주세요.



여기서 필요한 기능과 페이지를 정리해보도록 하겠습니다.
인스타그램 클론 페이지를 만들어 볼거에요.
이렇게 정리해서 아래 oven에서 구현해보면 우리가 생각지도 못했던 페이지나
놓쳤던 것을 체크할 수 있어요.

인스타클론

로그인화면

회원가입

포스트 목록

내프로필

Untitled

+ 페이지 추가

INSTACLONE

email

password

login

회원가입

Heading

Heading 1 Heading 2

Heading 3 Heading 4

Text

리스트

문장

단락

글자체

• List Item

1. List Item

2. List Item

리스트

순서 리스트

Link

Go to the Link

링크 양식

<https://goo.gl/jJFawA>



완성된 페이지입니다.

INSTACLONE

email

password

login

회원가입

INSTACLONE

email

password

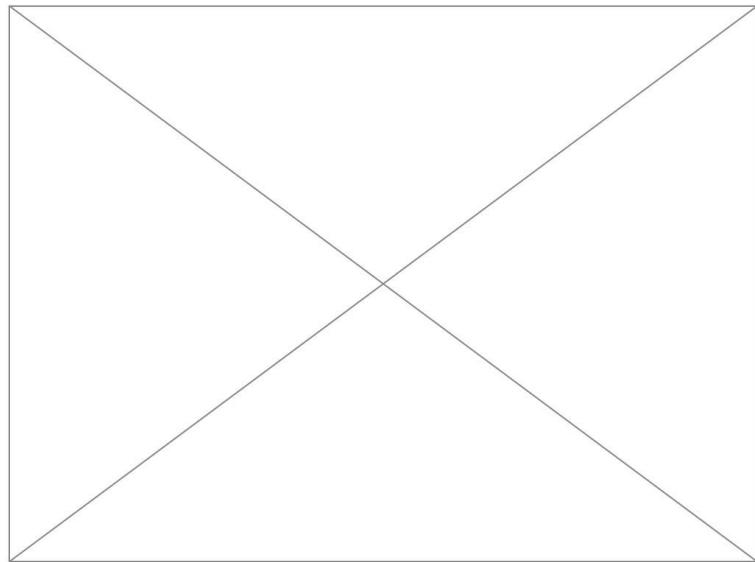
password confirm

회원가입

로그인

INSTACLONE

내 프로필



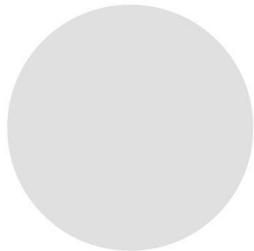
JEJUCODINGBASECAMP

인생을 풍부하게 하는 온갖 과실이 어디 있으랴? 이상! 우리의 청춘이 가장 많이 품고 있는 이상! 이것이야말로 무한한 가치를 가진 것이다 사람은 크고 작고 간에 이상이 있음으로써 용감하고 굳세게 살 수 있는 것이다

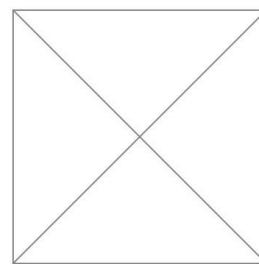
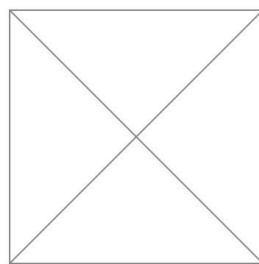
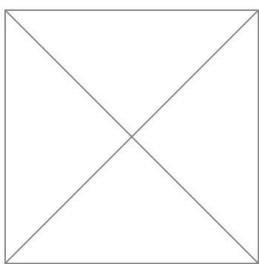
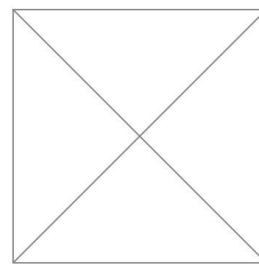
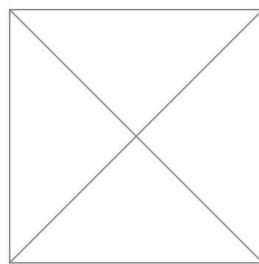
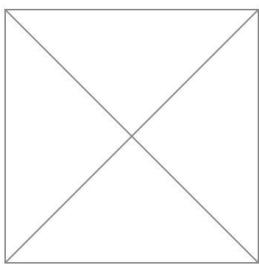
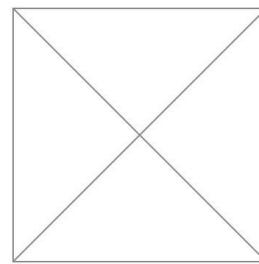
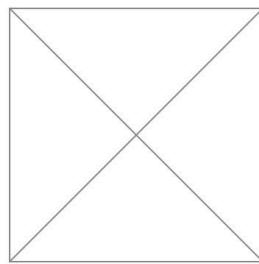
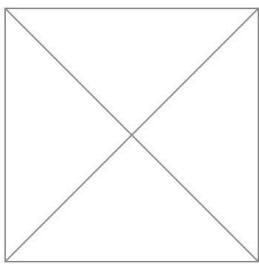
INSTACLONE

JEJUCODINGBASECAMP

내프로필 수정



무엇이 타오르고 있는가? 우리 눈이 그것을 보는 때에 우리
의 귀는 생의 찬미를 듣는다 그것은 웅대한 관현악이며 미묘
한 교향악이다 빠 끝에 스며들어 가는 열락의 소리다이것은
피어나기 전인 유소년에게서 구하지 못할 바이며 시들어 가
는



INSTACLONE



JEJUCODINGBASECAMP

이름

입력하세요.

사용자이름

입력하세요.

웹사이트 소개

입력하세요.

소개

입력하세요.

저장

취소

Google search results for "balsamiq mockups". The search bar shows the query. Below it, a navigation bar includes tabs for 전체 (selected), 이미지, 동영상, 도서, 뉴스, 더보기, and buttons for 설정 and 도구. The search results page displays a count of 315,000 results found in 0.39 seconds.

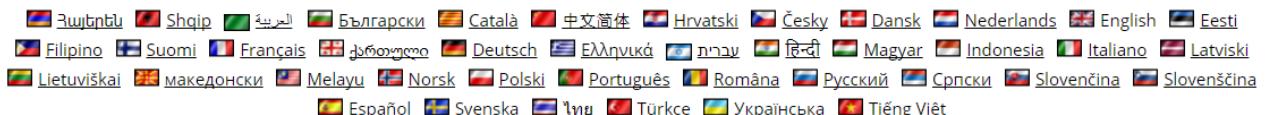
검색결과 약 315,000개 (0.39초)

Mockups - Balsamiq Mockups | Balsamiq

<https://balsamiq.com/products/mockups/> ▾ 이 페이지 번역하기

The first impression might be disorienting. There are very few interface elements on the screen. Start exploring however, and you'll find out that **Mockups** is filled ...

홈페이지를 처음 만들 때에는 의미 없는 문구가 필요할 때가 있어요.



Lorem Ipsum

"*Neque porro quisquam est qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit...*"

"There is no one who loves pain itself, who seeks after it and wants to have it, simply because it is pain..."

What is Lorem Ipsum?

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Why do we use it?

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

한글입숨

무의미한 한글 텍스트 생성기입니다.

왜 사용하나요?

문단 수

문단 길이

텍스트 소스

3

길게

청춘예찬

생성하기

 Tweet

 G+

 좋아요 57

Powered by [Thefron](#)

한글 페이지도 있습니다.

2° 3가지

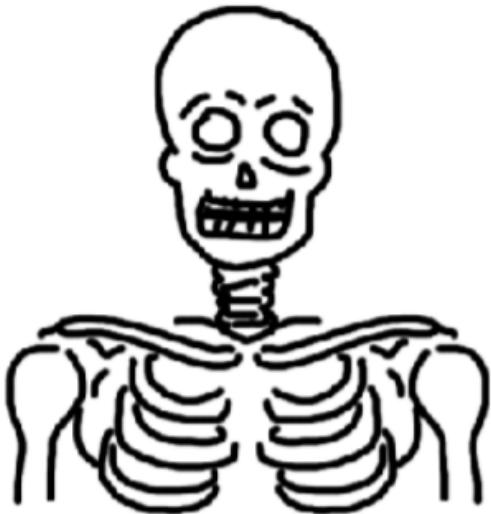
오전

HTML, CSS

html에이지1 시도

<http://studiomeal.com/>

HTML



HTML + CSS



HTML & CSS

HTML5와 XHTML1.0

	xhtml1.0	html5
DTD	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">	<!doctype html>
캐릭터셋	<meta http-equiv="content-type" content="text/html; charset=UTF-8" />	<meta charset="UTF-8">
type 표시 여부	생략 불가 <link type="text/css" rel="stylesheet" href="foo.css" /> <script type="text/javascript" src="foo.js"></script>	생략 가능 <link rel="stylesheet" href="foo.css" /> <script src="foo.js"></script>
태그 닫기 여부	단독으로 쓰이는 태그는 />로 닫아주어야 함 	단독으로 쓰이는 태그를 닫아줄 필요가 없음

주요 HTML5 아웃라인 엘리먼트

<article>

다른 곳에 단독으로 사용이 가능한 내용을 담는 엘리먼트입니다.

잡지의 기사, 포럼이나 블로그의 게시글, 사용자가 남긴 댓글, 위젯 등이 될 수 있습니다.

중첩해서 사용할 경우 하위의 article들은 상위 article에 관련된 내용임을 의미합니다.

<section>

문서에서 일반적인 섹션의 의미라고 생각하면 됩니다. 하나의 주제로 그룹화된 컨텐츠를 담습니다.

각 섹션의 주제는 일반적으로 섹션 요소의 하위 항목으로 제목을 포함시켜 식별해야합니다.

<header>

Sectioning content*의 헤더를 나타냅니다.

일반적으로 헤딩 엘리먼트를 포함하지만 반드시 그럴 필요는 없습니다.

헤딩 없이 목차, 검색폼같은 내용을 header로 감싸도 무방합니다.

<footer>

Sectioning content의 풋터를 나타냅니다.

순서상 반드시 Sectioning content의 끝에 위치할 필요는 없습니다.

* Sectioning content: 헤딩과 풋터의 범위를 정의합니다. article, aside, nav, section이 해당됩니다.

<nav>

링크의 묶음입니다. 모든 링크 묶음에 nav를 사용할 필요는 없습니다. 일반적으로 메인 네비게이션 역할의 링크 묶음에 사용합니다.

<aside>

페이지의 주요 내용의 흐름과 관련없는 다른 내용을 나타냅니다. 사이드바 형태의 디자인으로 자주 표현됩니다. 각주에는 aside를 사용하지 않습니다.

<h1> ... <h6>

Sectioning content의 제목을 나타냅니다. 서브타이틀로는 사용하면 안됩니다.

(틀린 예)

<header>

 <h1>집에서 맛있는 피자 만들기</h1>

 <h2>주위에서 쉽게 구입할 수 있는 재료를 중심으로</h2>

</header>

(좋은 예)

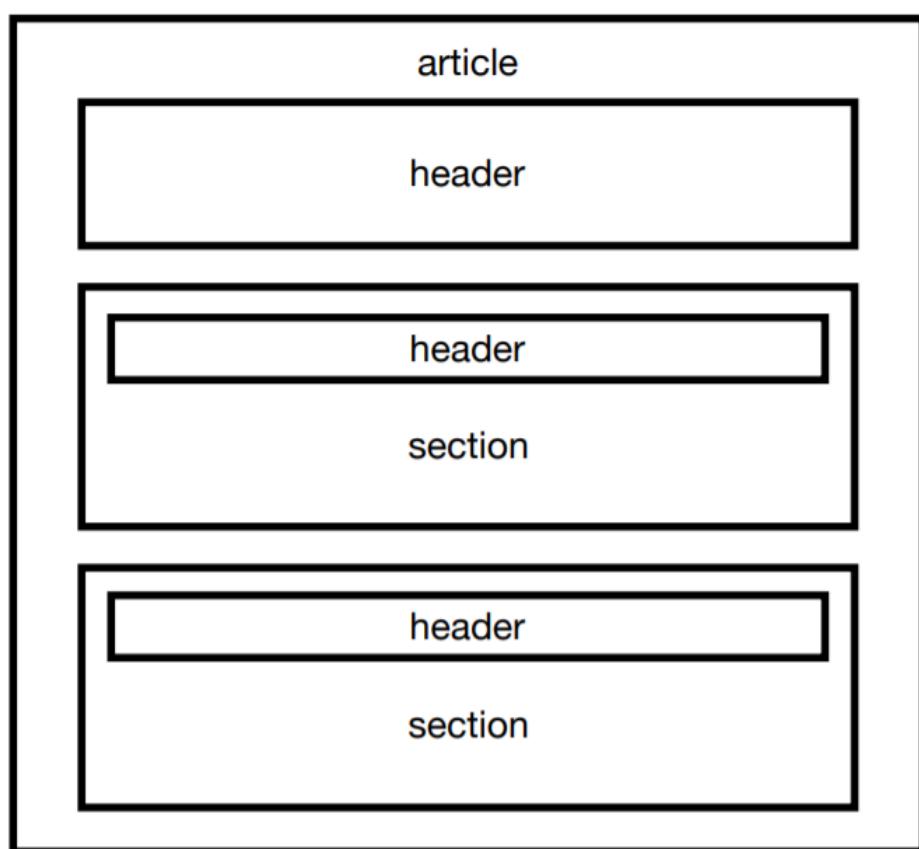
<header>

 <h1>집에서 맛있는 피자 만들기</h1>

 <p>주위에서 쉽게 구입할 수 있는 재료를 중심으로</p>

</header>

레이아웃 구성 예시



```
<article>
  <header>
    내용
  </header>
  <section>
    내용
  </section>
  <section>
    내용
  </section>
  <section>
    내용
  </section>
</article>
```

CSS를 적용하는 방법들

외부 CSS 파일 로드하기

```
<head>
  <link rel="stylesheet" href="css/foo.css">
</head>
```

HTML 안에 CSS 작성하기

```
<head>
  <style>
    body { font-size: 14px; }
  </style>
</head>
```

인라인 스타일

```
<body style="font-size: 14px;">
```

CSS 파일 안에 CSS 포함하기

```
@import "foo.css";
```

Reset CSS

에릭 마이어의 reset CSS

<http://meyerweb.com/eric/tools/css/reset/>

Normalize.css

<https://necolas.github.io/normalize.css/>

각 브라우저들에 자체적으로 적용되어있는 기본 스타일(여백 등)을 초기화 시켜서 원하는 디자인을 적용하기 쉽게 만들어 줍니다.

위 두 가지를 일반적으로 많이 사용하며, 프로젝트에 맞게 직접 만들어 쓰기도 합니다.

CSS의 기본 형태

선택자

```
body {  
    background-color: red;  
}  
          속성           값
```

body의 background-color를 red로 설정

선택자

```
header a {  
    display: inline-block;  
}  
          속성           값
```

header에 포함되어 있는 a의 display를 inline-block으로 설정

CSS 선택자(Selector)

아래 내용 외에도 많은 선택자들이 있지만, 공부를 처음 시작할 때 우선 알면 좋은 것들 위주로 정리했습니다.

*

문서 내의 모든 엘리먼트를 선택.

body, header, h1, section 등

해당 이름을 가진 html 엘리먼트를 선택.

#foo

id 선택자.

<div id="foo">의 형태로, id는 중복이 불가능한 유일한 값.

.foo

class 선택자.

<div class="foo">의 형태로, 중복이 가능. class로 자주 쓰는 스타일을 정의해 두고 재사용.

input[type="text"]

input 엘리먼트 중 type 속성의 값이 text인 엘리먼트를 선택.

.foo > p

class="foo"인 엘리먼트의 직계 자식 엘리먼트 중 p를 선택.

<section class="foo">

 <header>

 <h1>집에서 맛있는 피자 만들기</h1>

 <p>주위에서 쉽게 구할 수 있는 재료를 중심으로</p>

 </header>

 <p>AAAAA</p> ← .foo > p

</section>

.foo:first-child

.foo:last-child

.foo:nth-child(n)

class="foo"인 엘리먼트 중 첫번째, 마지막, n번째 자식인 엘리먼트를 선택.

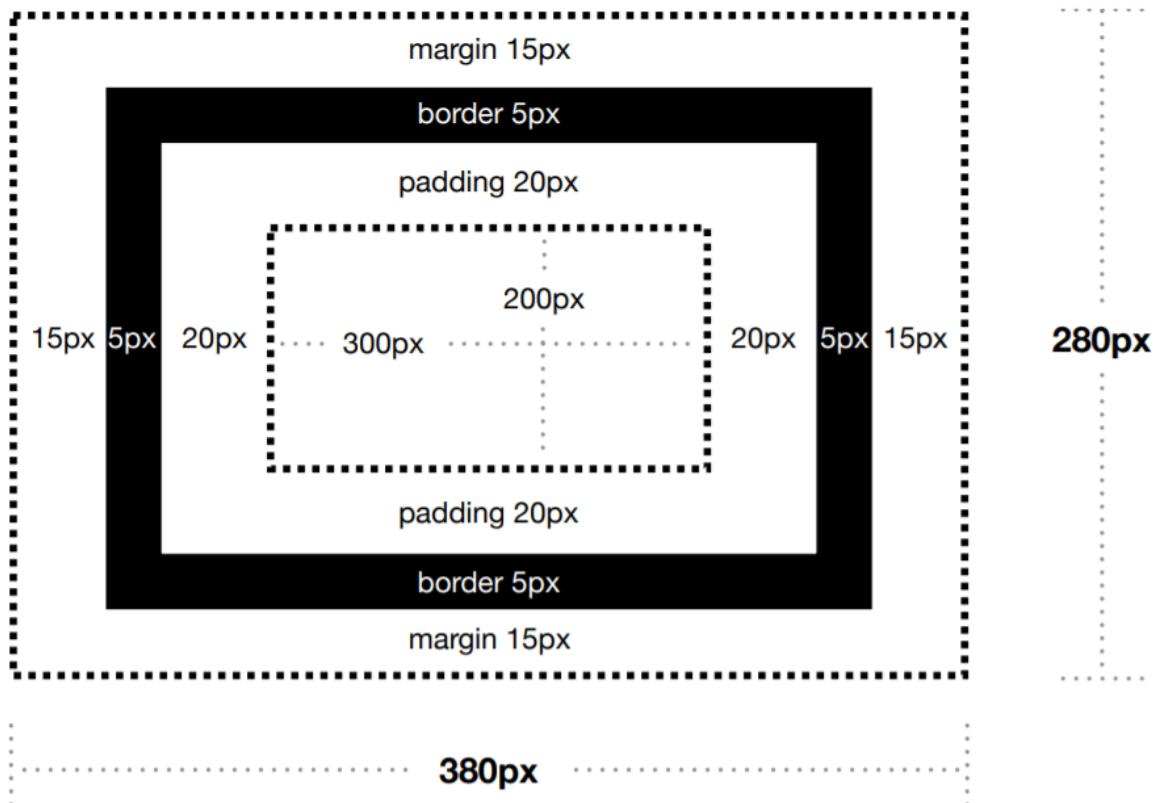
 <li class="foo">1 ← .foo:first-child

 <li class="foo">2 ← .foo:nth-child(3)

 <li class="foo">3 ← .foo:last-child

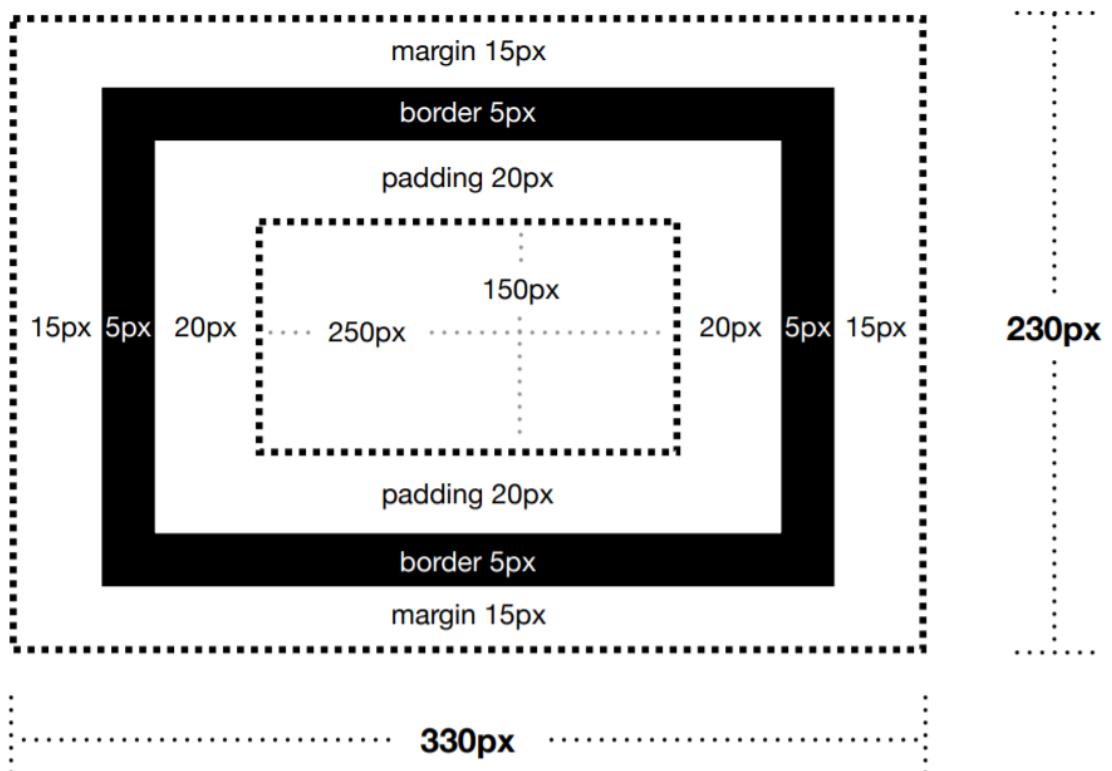
 <li class="foo">4

 <li class="foo">5



box-sizing: content-box (기본값)

width: **300px**; height: **200px**; padding: **20px**; border: **5px solid black**; margin: **15px**;



box-sizing: border-box (border까지 width에 포함)

width: 300px; height: 200px; padding: 20px; border: 5px solid black; margin: 15px;

2°3가지

제작

Javascript & JQuery

강사소개

이름 : 이병재

- (주)제주넷 웹개발자
- 경력
 - 배움커뮤니티 - Ruby on Rails
 - 온라인시험솔루션회사 - classic ASP
 - 무선공중전화기 개발회사 - JSP

이번 시간에 배울 것은

JavaScript

그리고

jQuery

자바스크립트란?

- 프로그래밍 언어
- 웹브라우저가 해석해서 실행할 수 있는 유일한 프로그래밍 언어
- 요즘에는 게임프로그래밍, 서버프로그래밍 등 다양한 분야에서 쓰여지고 있음.

HTML

정보를 표현한다

CSS

정보를 꾸며준다

JAVASCRIPT

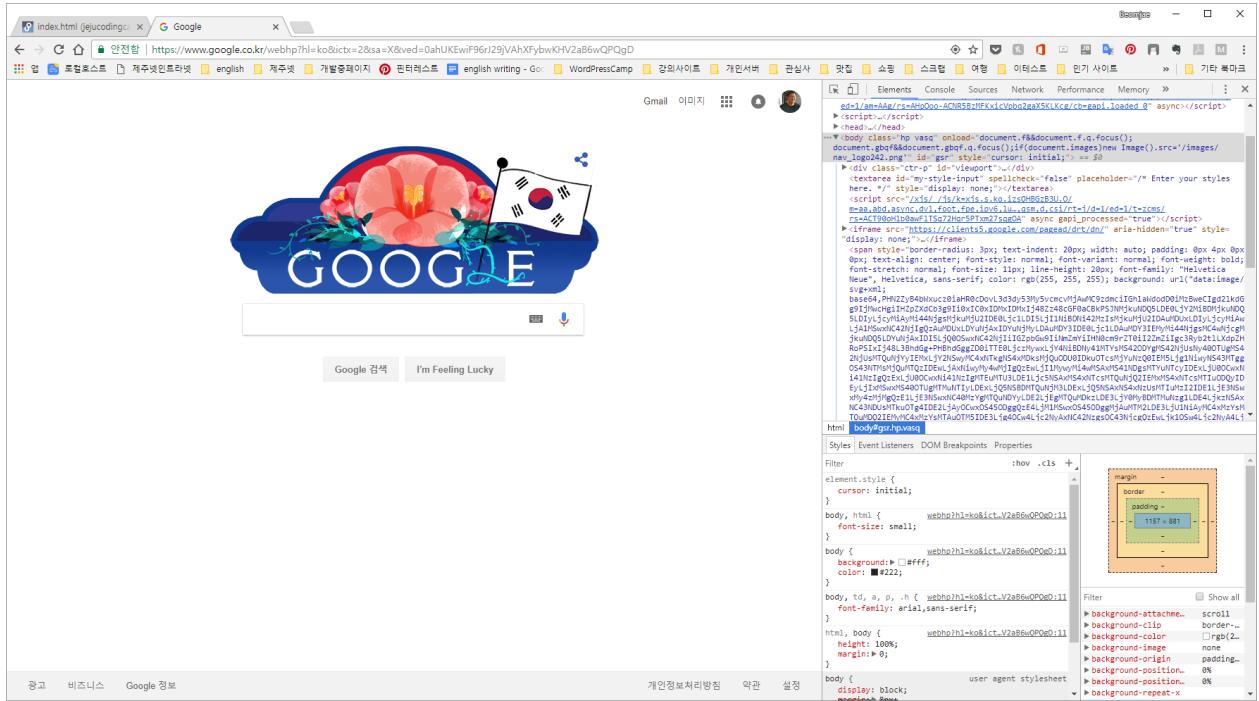
HTML을 프로그래밍으로 제어한다

실습 방법

- goorm IDE를 사용해 소스코드를 작성하고 실행합니다.
- 브라우저는 chrome을 사용합니다

크롬 개발자 도구

단축키 : F12 또는 Ctrl + Shift + i



HTML에서 javascript로 드립기

INLINE

HTML내에 javascript를 포함하고 로드합니다.

SCRIPT

외부 파일로 저장을 하고 로드합니다.

변수 variable

변하는 값

값을 저장할 수 있는 문자 또는 이름

변수 사용 방법

변수를 선언합니다.

사용할 변수의 이름과 종류를 프로그램에 알립니다.

변수를 할당합니다.

변수에 저장될 값을 제공합니다.

변수를 사용합니다.

변수에 저장된 값을 검색하여 프로그램에서 사용합니다.

변수를 선언합니다.

사용할 변수의 이름과 종류를 프로그램에 알립니다.

var a;

var my_name;

변수를 할당합니다.

변수에 저장될 값을 제공합니다.

a = 1;

my_name = "lee";

변수를 사용합니다.

변수에 저장된 값은 절제하여 프로그램에서 사용합니다.

`console.log(a);`

`console.log(my_name);`

비교연산자

`==` (같은)

`!=` (다른)

`>` (~보다 큼)

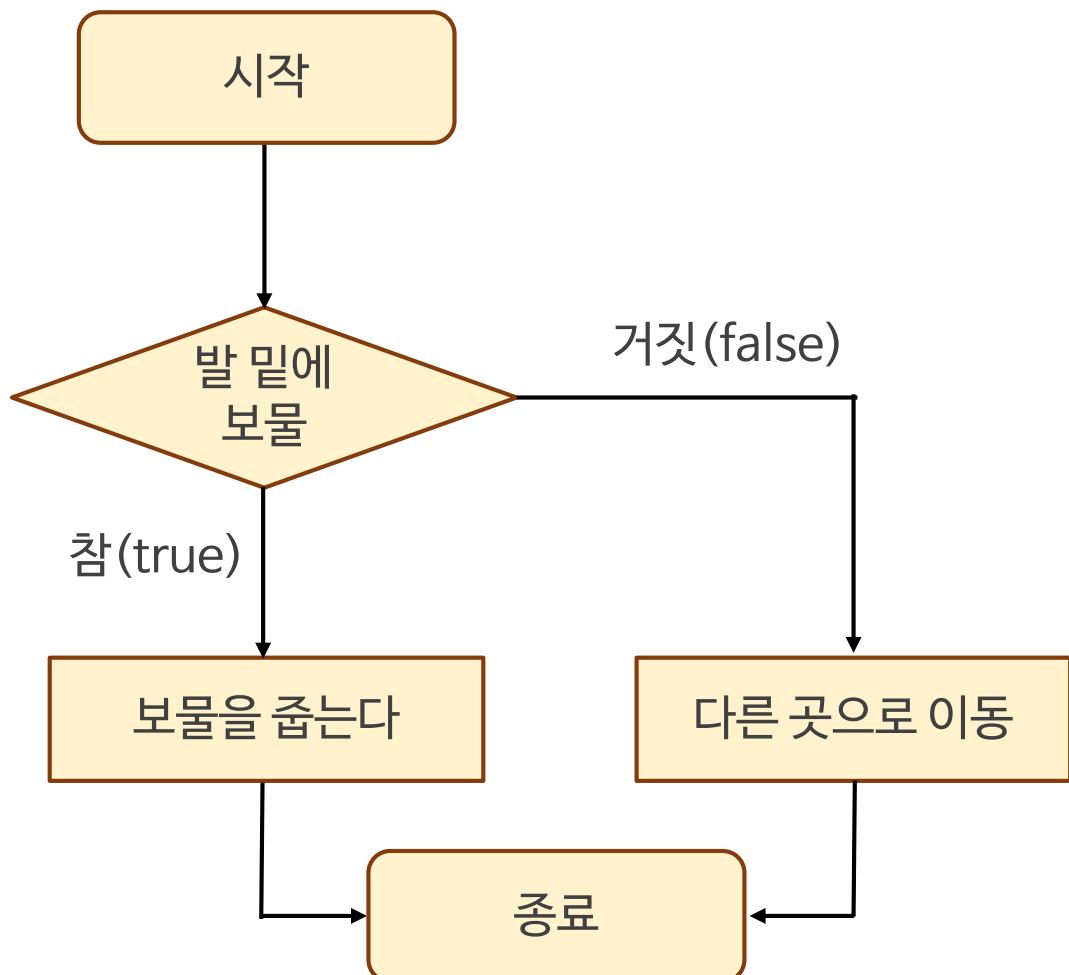
`>=` (~보다 크거나 같은)

`<` (~보다 작은)

`<=` (~보다 작거나 같은)

조건문 condition

조건이 참(true)인지 거짓(false)인지에 따라 계산이나 상황을 수행하는 프로그래밍 언어의 특징



if

```
var test = 5;  
if(test < 10) {  
    // codes  
}
```

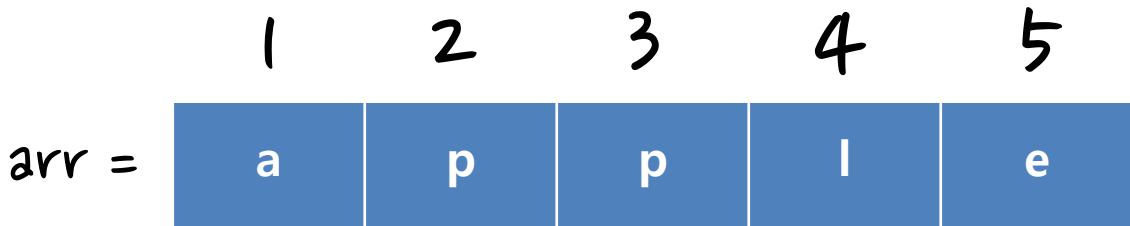
if

```
var test = 5;  
if(test < 10) {  
    // codes  
} else {  
    // codes  
}
```

배열 array

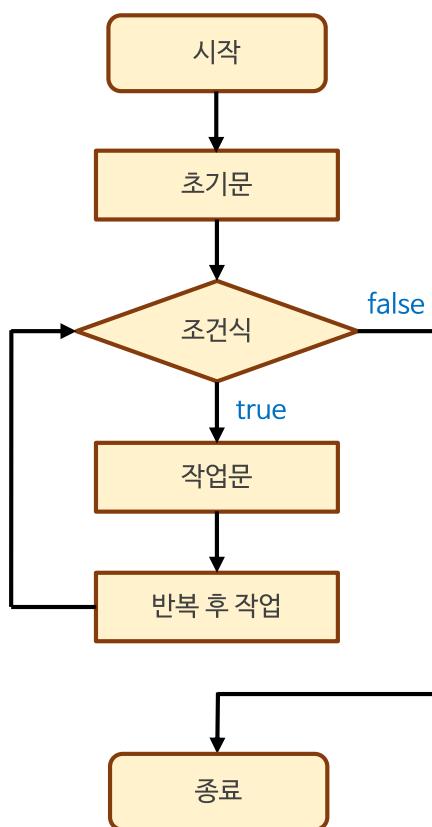
번호(인덱스)와 번호에 대응하는 데이터들로 이루어진 자료 구조

```
var arr = ['a', 'p', 'p', 'l', 'e'];
```



반복문

특정한 부분의 코드가 반복적으로 수행될 수 있도록 한다



```
var cars = ["BMW", "volvo", "Saab", "Ford", "Fiat",  
"Audi"];
```

```
var text = "";
```

cars의 내용을 모두 text 변수에 넣으려면?

before

```
text += cars[0] + "<br>";  
text += cars[1] + "<br>";  
text += cars[2] + "<br>";  
text += cars[3] + "<br>";  
text += cars[4] + "<br>";  
text += cars[5] + "<br>";
```

after

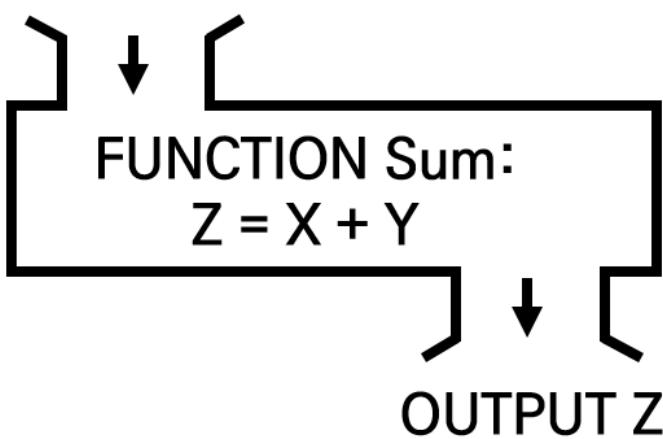
```
var i;  
for (i = 0; i < cars.length; i++) {  
    text += cars[i] + "<br>"  
}
```

할수

자바스크립트 코드의 끝을 단위

INPUT X = 3

INPUT Y = 7



선언방법

```
function add(x, y) {  
    return x + y;  
}
```

사용방법

```
add(1, 2);
```

객체 object

데이터(주체)와 그 데이터에 관련되는 동작(절차, 방법, 기능)을 모두 포함하고 있는 개념

프로퍼티(property)

객체의 속성을 나타내는 접근 가능한 이름과 활용 가능한 값으로 가지는 특별한 형태

메소드(method)

객체가 가지고 있는 동작

객체 선언 방법

```
var car = {  
    type: "Fiat",  
    model: "500",  
    color: "white"  
};
```

객체 선언 방법

```
var person = {  
    firstName: "John",  
    lastName: "Doe",  
    age: 50,  
    fullname : function() {  
        return firstName + lastName;  
    }  
};
```

자바스크립트 제작자

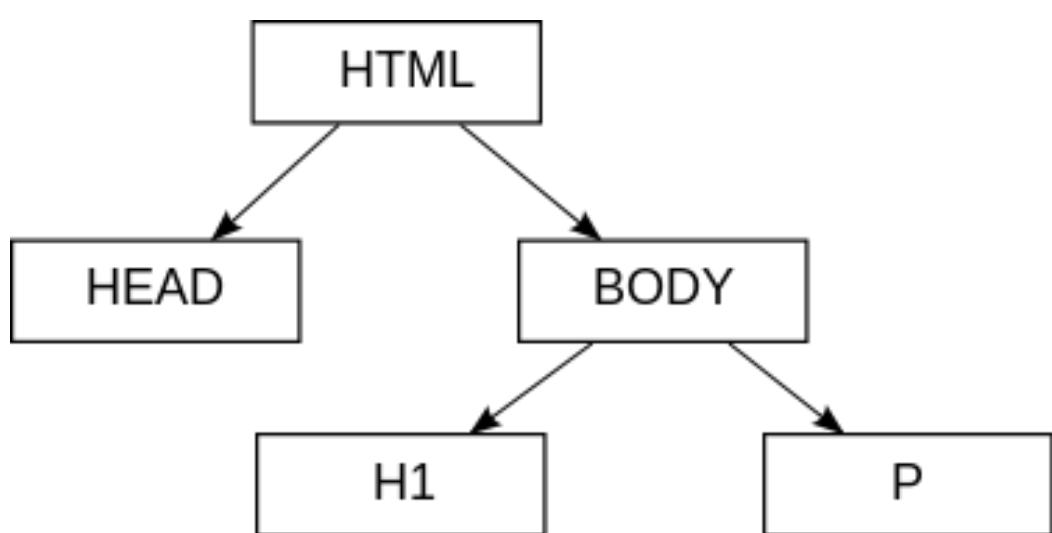
car["model"];

car["color"];

person["age"];

person.fullName();

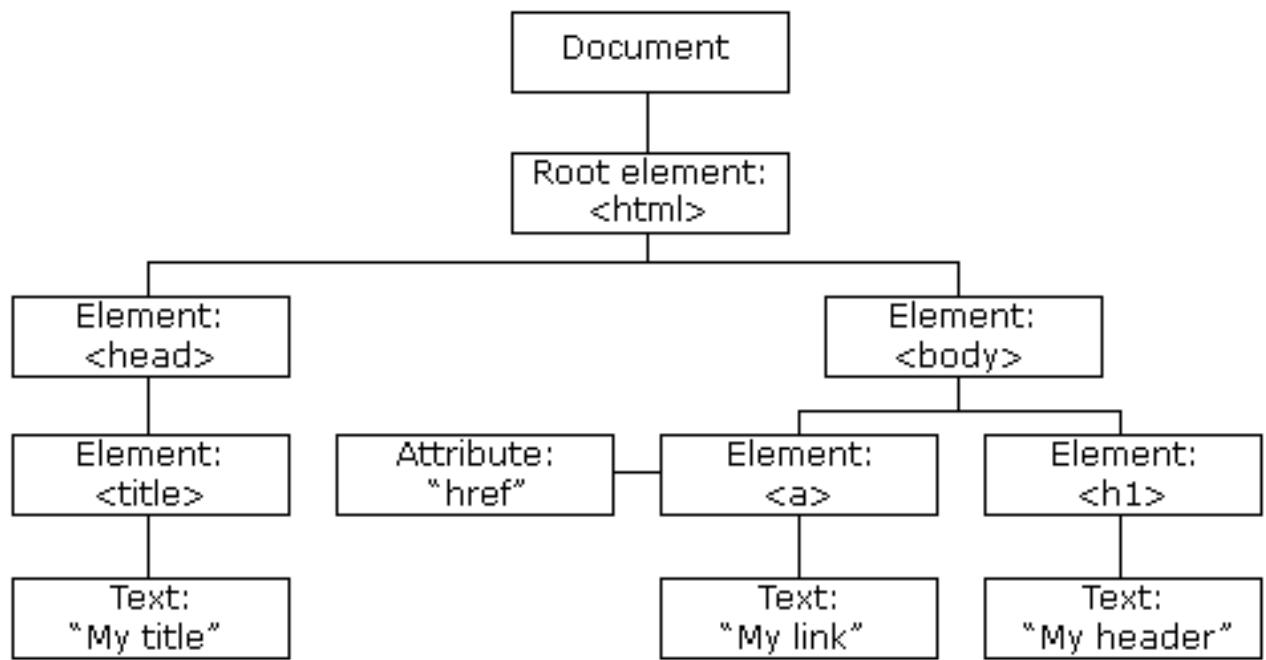
DOM(Document Object Model)



This is an header

This is some text

DOM(Document Object Model)



HTML FORM

*Mandatory to fill

Firstname:

*

Lastname:

Birthday:

Day	Month	Year
-----	-------	------

*

Username:

*

E-mail:

Website:

Password:

*

Re-password:

*



I agree to the terms & conditions.

Sign up

First name:

Last name:

<form>

First name:

<input type="text" name="firstname">

Last name:

<input type="text" name="lastname">

</form>

구구단 만들기

수는 시간

라이브러리란

자주 사용하는 코드들을 재사용할 수 있는 형태로 제공해서 프로그래밍 효율을 높여주는 코드들



jQuery

jQuery is a fast, small, and feature-rich JavaScript library.

엘리먼트를 선택하는 강력한 방법과

선택된 엘리먼트들을 효율적으로 제어할 수 있는 다양한 수단을 제공하는

자바스크립트 라이브러리

jQuery

<http://jquery.com/>

jQuery 레인저이트 모듈사이트

<http://www.jqueryrain.com/>

로딩바너

CDN 사용

구글apis - jquery cdn

jQuery CDN - Latest Stable Versions

Powered by  StackPath

jQuery Core

Showing the latest stable release in each major branch. [See all versions of jQuery Core.](#)

jQuery 3.x

- jQuery Core 3.2.1 - [uncompressed](#), [minified](#), [slim](#), [slim minified](#)

jQuery 2.x

- jQuery Core 2.2.4 - [uncompressed](#), [minified](#)

<head>

```
<script src="https://code.jquery.com/jquery-2.2.4.min.js"></script>
</head>
```

사용순서

1. 선택하고

2. 실행한다

1. 선택하고

Selectors

`$(“p”)`

`$(“#test”)`

`$(“.test”)`

2. 실행한다

Selectors

`$(“p”).hide();`

`$(“#test”).show();`

`$(“.test”).click(function(){`

`alert(“클릭!”);`

`});`

다양한 선택 방법

다양한 실행 방법

Selectors

```
$(“p”).hide();  
$(“#test”).show();  
$(“.test”).click(function(){  
    alert(“클릭!”);  
});
```

다양한 선택자 방법

- 기본
 - element
 - .class
 - #id
 - all(*)
- 계층
 - child(>)

제한 조건

- :eq(index)
- :even
- :odd
- :first
- :last
- :gt(index)
- :lt(index)
- :not(filter)

조건문 조건

- :contains(text)
- :empty
- :has(selector)

속성 필터

- :attributecontains - input[name*='man']
- attributeEquals - input[name='newsletter']

자식 필터

- :first-child, :last-child
- :nth-child(2)
- :nth-child(even), :nth-child(odd)
- :nth-child(3n)

다양한 실행 방법

- 속성 필터 연결
- DOM 탐색
- DOM 조작
- CSS
- 이벤트
- 흐름
- AJAX
- 흐름

속성값 제어

속성값 초기화

`$("#name").val();`

속성값 쓰기

`$("#name").val();`

DOM Element

```
<div class="wrapper">  
  <h1 class="title">TITLE</h1>  
  <ul class="items">  
    <li class="item">item 1</li>  
    <li class="item">item 2</li>  
  </ul>  
</div>
```

DOM 선택

부모 선택

```
$(“span”).parent();
```

자식 선택

```
$(“div”).children();
```

DOM 선택

```
$(“div”).first();
```

```
$(“div”).last();
```

DOM 선택

```
$(“#btn1”).closest(“.item”);
```

```
<div class=“item”>
```

```
  <div class=“item”>
```

```
    <div class=“item”>
```

```
      <button id=“btn1”>click</button>
```

```
    </div>
```

```
  </div>
```

```
</div>
```

DOM 테스트

```
$("#test1").text("Hello world");
$("#test2").html("<b>Hello world</b>");
$("#test3").val("<b>Hello world</b>");
```

DOM 테스트

```
 $("p").append("Some appended text.");
 $("p").prepend("Some prepended text.");
 $("#div1").remove();
```

css

```
 $("div").addClass("important");
 $("h1, h2, p").removeClass("blue");
 $("h1, h2, p").toggleClass("blue");
```

css

```
 $("p").css("background-color");
 $("p").css("background-color", "yellow");
```

Effect(효과)

```
$( "#p" ).hide();
```

```
$( "#p" ).show();
```

```
$( "#p" ).toggle();
```

Effect(효과)

```
$( "#div1" ).fadeIn();
```

```
$( "#div1" ).fadeOut();
```

```
$( "#div1" ).fadeToggle();
```

```
$( "#div3" ).fadeIn(3000);
```

이벤트 (event)

어떤 사건

- 사용자가 버튼을 클릭했을 때
- 마우스를 스크롤 했을 때
- 필드의 내용을 바꾸었을 때

클릭 이벤트

```
$( "#p" ).click(function(){
  $(this).hide();
});
```

마우스 이동 이벤트

```
$( "#p1" ).mouseenter(function(){
  alert("You entered p1!");
});
$( "#p1" ).mouseleave(function(){
  alert("Bye! You now leave p1!");
});
```

<https://api.jquery.com/category/events/>



Your donations help fund the continued development and growth of **jQuery**.

[SUPPORT THE PROJECT](#)

[Download](#) [API Documentation](#) [Blog](#) [Plugins](#) [Browser Support](#)

Search



- Ajax
 - Global Ajax Event Handlers
 - Helper Functions
 - Low-Level Interface
 - Shorthand Methods
- Attributes
- Callbacks Object
- Core
- CSS
- Data
- Deferred Object
- Deprecated
 - Deprecated 1.3
 - Deprecated 1.7
 - Deprecated 1.8
 - Deprecated 1.9
 - Deprecated 1.10
 - Deprecated 3.0

Category: Events

These methods are used to register behaviors to take effect when the user interacts with the browser, and to further manipulate those registered behaviors.

.bind()

Also in: [Deprecated > Deprecated 3.0](#) | [Events > Event Handler Attachment](#)

Attach a handler to an event for the elements.

.blur()

Also in: [Events > Form Events](#) | [Forms](#)

Bind an event handler to the "blur" JavaScript event, or trigger that event on an element.

.change()

Also in: [Events > Form Events](#) | [Forms](#)

Bind an event handler to the "change" JavaScript event, or trigger that event on an element.

.click()

Also in: [Events > Mouse Events](#)

Bind an event handler to the "click" JavaScript event, or trigger that event on an element.

.contextmenu()

Also in: [Events > Mouse Events](#)

Bind an event handler to the "contextmenu" JavaScript event, or trigger that event on an element.

ON

```
$("p").on("click", function(){  
    $(this).hide();  
});
```

이벤트 걸세우기

구글에서 원하는 태그명과 jquery를 함께 검색한다.

(1)

jquery radio change

jQuery Plugins

- jQuery UI
- jQuery validation
- jQuery lightbox
- Bootstrap
- more... more... more...

30%할인

오늘전

Python

주석과 입출력

```
1 # 행 단위 주석입니다.  
2  
3 """  
4  
5 " 로 세번으로 묶거나  
6 ' 로 세번으로 묶으면  
7 열 단위로 주석처리됩니다.  
8  
9 """
```

```
1 a = input() # 입력을 받아 a 변수에 저장한다.
```

```
1 a = input("a를 입력하세요 : ")  
2 print(a + a)
```

```
1 age = 10  
2 name = 'leehojun'  
3  
4 print('1. 제 이름은', name, '입니다. 제 나이는', age, '입니다. ')  
5 print('2. 제 이름은 {}입니다. 제 나이는 {}입니다.'.format(name, age))  
6 print('3. 제 이름은 %s입니다. 제 나이는 %d입니다.' % (name, age))
```

자료형

```
1 #Type과 변수
2 #다른 타입끼리는 연산이 안됨
3 #(숫자는 손실이 없도록 자연스럽게 형변환이 됨)
4 a = 10
5 b = 10.1
6 c = -1
7 d = True
8 e = 'good'
9 f = '10'
10 g = 'lee'
11 h = 'hojun'
12 i = 'jun100!!'
13 j = 10 + 2j
14 k = 0b110
15 l = 0o56
16 m = 0xAC

18 print('type(10) : ', type(a))
19 print('type(10.1) : ', type(b))
20 print('type(-1) : ', type(c))
21 print('type(True) : ', type(d))
22 print('type(\\'good\\') : ', type(e))
23 print('\\\'10\\' + '\\\'10\\' : ', f + f)
24 print('\\\'10\\' * 3, f*3)
25 print('\\\'lee\\' + '\\\'hojun\\'', g + h)
26 print('type('\\\'hojun\\') : ', type(h))
27 print('type('\\\'hojun100!!\\') : ', type(i))
28 print('type(10 + 2j) : ', type(j))
29 print('type(0b110) : ', type(k))
30 print(k)
31 print('type(0o56) : ', type(l))
32 print(l)
33 print('type(0xAC) : ', type(m))
34 print(m)
```

형변환과 산술연산, 비교연산

```
1 print(type(int(3.5)))
2 print(int(3.5))
3 print(type(float(3)))
4 print(float(3))
5 print(type(str(3)))
6 print(str(3))
```

```
1 print("bool('test') : ", bool('test'))
2 print("bool(1) : ", bool(1))
3 print("bool(0) : ", bool(0))
4 print("bool(-1) : ", bool(-1))
5 print("bool(' ') : ", bool(' '))
6 print("bool('') : ", bool(''))
7 print("bool(None) : ", bool(None))
```

```
1 a = 9
2 b = 2
3
4 print('a + b = ', a + b)
5 print('a - b = ', a - b)
6 print('a * b = ', a * b)
7 print('a ** b = ', a ** b)
8 print('a / b = ', a / b)
9 print('a // b = ', a // b)
10 print('a % b = ', a % b)
```

```

1 a = 10
2 b = 5
3
4 print('a > b : ', a > b)           # a의 값이 b보다 큰가?
5 print('a < b : ', a < b)           # a의 값이 b보다 작은가?
6 print('a >= b : ', a >= b)          # a의 값이 b보다 크거나 같은가?
7 print('a <= b : ', a <= b)          # a의 값이 b보다 작거나 같은가?
8 print('a == b : ', a == b)           # a와 b의 값이 같은가?
9 print('a != b : ', a != b)           # a와 b의 값이 다른가?

```

```

1 print('True and True : ', True and True)
2 print('True and False : ', True and False)
3 print('False and False : ', False and False)
4 print('True or True : ', True or True)
5 print('True or False : ', True or False)
6 print('False or False : ', False or False)
7 print('True & True : ', True & True)
8 print('True & False : ', True & False)
9 print('False & False : ', False & False)
10 print('True | True : ', True | True)
11 print('True | False : ', True | False)
12 print('False | False : ', False | False)

```

```

1 a = 10
2
3 a += 2
4 print('a += 2 : ', a)                # a = a + 2 와 같음
5 a -= 2
6 print('a -= 2 : ', a)                # a = a - 2 와 같음
7 a *= 2
8 print('a *= 2 : ', a)                # a = a * 2 와 같음
9 a **= 2
10 print('a **= 2 : ', a)               # a = a ** 2 와 같음
11 a // 2
12 print('a // 2 : ', a)                # a = a // 2 와 같음
13 a %= 2
14 print('a %= 2 : ', a)                # a = a % 2 와 같음

```

함수

```
1 #들여쓰기
2 def printnum():
3     print('1')
4     print('2')
5     print('3')
6 printnum()
```

```
1 def printnum():
2     print('1')
3     print('2')
4     print('3')
5 printnum()
```

```
1 def function(x, y): # Input
2     z = x + y          # Funciton
3     return z           # Output
4 print(function(5, 7))
```

함수

```
1 #2개의 함수를 만들어서  
2 #1개의 함수에서 2번함수를 호출하도록 작성하세요.  
3 #1번 함수 : 1부터 10까지 출력하고  
4 #2번 함수 : good job!  
5 def one():  
6     for i in range(10):  
7         print(i, '번째 숫자')  
8     two()  
9 def two():  
10    print('good job')  
11 one()
```

```
1 #팩토리얼 함수를 제작해 보겠습니다.  
2 #5! = 5 * 4 * 3 * 2 * 1  
3 def factorial(x):  
4     value = 1  
5     for i in range(1, x+1):  
6         value = value * i  
7     return value  
8 print(factorial(5))
```

記入

```
1 def one():
2     print('hello')
3     return
4 def two():
5     print('hello')
6     print('hello')
7     return None
8 def three():
9     print('hello')
10    print('hello')
11    print('hello')
12    return None
13 nameone = [one, two, three]
14 nameone[0]()
```

if, elif, else

```
1 score = 90
2
3 if score >= 90:
4     print("Mom : I'm so happy!!")
5 elif score >= 80:
6     print("Mom : Not Bad~")
7 else :
8     print("Mom : I'm so angry!!")
```

1. 단일문

```
1 for a in range(10):
2     print(a)
3
4 for a in range(1,10):
5     print(a)
```

```
1 list = ['one', 'two', 'three']
2
3 # for a in ['one', 'two', 'three'] 와 같음
4
5 for a in list:
6     print(a)
```

```
1 sum = 0
2 for a in range(1,101):
3     sum += a
4 print(sum)
```

자료구조

```
1 A = [1,1,2,2,3,3,4,4,5,7,7,7,7,7,7]
2 B = (1,2,3,4,5,7,7,7,7,7,7,7,7,7)
3 C = set(A)
4 D = {'one' : 1, 'two' : 2, 'three' : 3}
5 print (A)
6 print (B)
7 print (C)
8 print (D)
```

```
1 A = [1,1,2,2,3,3,4,4,5,7,7,7,7,7,7]
2 B = (1,2,3,4,5,7,7,7,7,7,7,7,7,7)
3 C = set(A)
4 D = {'one' : 1, 'two' : 2, 'three' : 3}
5 print (A[2])
6 print (B[2])
7 print (D['one'])
```

```
1 A = [1,1,2,2,3,3,4,4,5,7,7,7,7,7,7]
2 B = (1,2,3,4,5,7,7,7,7,7,7,7,7,7,7)
3 C = set(A)
4 D = {'one' : 1, 'two' : 2, 'three' : 3}
5 maxA = max(A)
6 maxB = max(B)
7 maxC = max(C)
8 maxD = max(D)
9 print(maxA)
10 print(maxB)
11 print(maxC)
12 print(maxD)
```

```
1 A = [1,1,2,2,3,3,4,4,5,7,7,7,7,7,7]
2 B = (1,2,3,4,5,7,7,7,7,7,7,7,7,7,7)
3 C = set(A)
4 D = {'one' : 1, 'two' : 2, 'three' : 3}
5 minA = min(A)
6 minB = min(B)
7 minC = min(C)
8 minD = min(D)
9 print(minA)
10 print(minB)
11 print(minC)
12 print(minD)
```

내장함수

```
1 A = [1,1,2,2,3,3,4,4,5,7,7,7,7,7,7]
2 B = (1,2,3,4,5,7,7,7,7,7,7,7,7,7)
3 C = set(A)
4 D = {'one' : 1, 'two' : 2, 'three' : 3}
5 A.reverse()
6 #reverseB = B.reverse()
7 #reverseC = C.reverse()
8 #reverseD = D.reverse()
9 print(A)
10 #print(reverseB)
11 #print(reverseC)
12 #print(reverseD)
```

swap

```
1 #swap  
2 x = 10  
3 y = 20  
4 temp = x  
5 x = y  
6 y = temp  
7 print (x)  
8 print (y)
```

```
1 #swap  
2 x = 10  
3 y = 20  
4 x, y = y, x  
5 print (x)  
6 print (y)
```

슬라이싱

```
1 name = '안녕하세요. 저는 11살 이호준이라고 합니다.'
2 print('name[3] :', name[3])
3 print('name[3:] :', name[3:])
4 print('name[3:9] :', name[3:9])
```

class

```
1 class Car(object):
2     MaxSpeed = 300
3     MaxPeople = 5
4     def move(self, x):
5         print(x, '의 스피드로 움직이고 있습니다.')
6     def stop(self):
7         print('멈췄습니다.')
8 k5 = Car()
9 k3 = Car()
10 k5.move(10)
11 k5.stop()
12 k3.move(5)
13 k3.stop()
14 print(k5.MaxSpeed)
15 print(k3.MaxSpeed)
```

```
1 class Car(object):
2     kinds = []
3     def addkinds(self, name):
4         self.kinds.append(name)
5     MaxSpeed = 300
6     MaxPeople = 5
7     def move(self, x):
8         print(x, '의 스피드로 움직이고 있습니다.')
9     def stop(self):
10        print('멈쳤습니다.')
11 k5 = Car()
12 k3 = Car()
13 k5.addkinds('k5')
14 k3.addkinds('k3')
15 print(k5.kinds)
```

class

```
1 class Car(object):
2     def __init__(self, name):
3         self.kinds = []
4         self.kinds.append(name)
5         MaxSpeed = 300
6         MaxPeople = 5
7         def move(self, x):
8             print(x, '의 스피드로 움직이고 있습니다.')
9         def stop(self):
10            print('멈췄습니다.')
11 k5 = Car('k5')
12 k3 = Car('k3')
13 print(k5.kinds)
```

3.1.1

```
1 add = lambda x, y : x + y
2 mul = lambda x, y : x * y
3 print(add(2,5))
4 print(mul(2,5))
```

```
1 add = lambda x, y : x + y
2 sub = lambda x, y : x - y
3 mul = lambda x, y : x * y
4 div = lambda x, y : x / y
5
6 calculator = [add, sub, mul, div]
7 print(calculator[0](2,5))
```

```
1 calculator = [lambda x, y : x + y, lambda x, y : x - y, lambda x, y : x * y, lambda x, y : x / y]
2 listx = [100, 200, 300, 400, 500]
3 listy = [1000, 2000, 3000, 4000, 5000]
4 sumxy = list(map(calculator[0], listx, listy))
5 print(sumxy)
```

break, continue

```
1 age = int(input("나이를 입력해주세요!"))
2
3 while 1:
4     if age > 19:
5         print("성년이되었습니다")
6         break
7     else:
8         age+=1
9     if age == 15:
10        print(age, "살에는 서울에서 나이를 하나 더 먹었습니다.")
11        continue
12        print("제주도에서 나이를 하나 더 먹었습니다.", age)
13 print('세상에 나온 것을 환영합니다.')
```

args, kwargs

```
1 def print_args(*args):
2     print(args)
3     for x in args:
4         print(x)
5
6 print_args(100, True, 'good')
```

```
1 def print_args_kwargs(*args, **kwargs):
2     print('args : ', args)
3     for x in args:
4         print(x)
5     print('kwargs : ', kwargs)
6     for x in kwargs:
7         print(x)
8         print(kwargs[x])
9
10 inputlist = [100, True, 'good']
11 inputdic = {'apple': 100, 'banana': 200, 'melon': 500}
12 print_args_kwargs(*inputlist)
13 print('-----')
14 print_args_kwargs(**inputdic)
15 print('-----')
16 print_args_kwargs(*inputlist, **inputdic)
```

데코레이터

```
1 def decorator(printname_func): #3
2     def wrapper(w_name): #5
3         print("이렇게 Object 앞에도 실행할 수 있습니다.")
4         printname_func(w_name) #6
5         print("이렇게 Object 뒤에도 실행할 수 있습니다.")
6     return wrapper #4
7
8 @decorator #2
9 def printname(name): #7
10    print(name)
11
12 printname('leehojun') #1
```

한글코딩(hangulcoding.org)

```
1 def 한글코딩(글자):  
2     print(글자)  
3 한글코딩('안녕, 세상아.')
```

4°일자

오전

Django

(해커들은 자료를 우려먹고 있습니다.)

동영상(1편) : <https://youtu.be/wqADDhjmQUY>

유튜브 검색 : 제주코딩해커톤

Sourcefile :

jejucodinghackathon.paullabedu.synology.me/download.zip

Django 트랙 상편 - Step1 ~ Step4

Django 트랙 하편 - Step5 ~ Step9

* 이 책은 동영상강좌와 Sourcefile이 진행될 수 있는 책입니다.

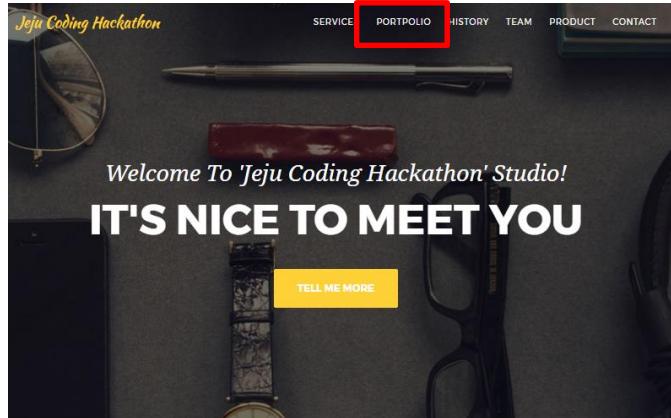
Step 1

구글 IDE 환경설정

홈페이지 구성

(완성페이지 <https://jejuhackathon.run.goorm.io/>)

(완성페이지에서는 해커톤 진행 중에만 실행됩니다.)



SERVICES

jeju Coding Hackathon은 무료로 진행되는 코딩 해커톤입니다.



SW인력강화

"Jeju Coding Hackathon"은 활성화된 제품으로서 그동안 만나보지 못한 고급 SW 기술 경험을 제공합니다.



SW인력강화

"Jeju Coding Hackathon"은 제주도의 지역 특성을 극복하고 이 분야에 다양한 인력을 사귈 수 있도록 지원합니다.



지속가능SW행사

"Jeju Coding Hackathon"은 Facebook 그룹 제주포밍해커톤으로 매년 월파티를 모집하며 지속적인 모임을 제공합니다.

Main 홈페이지

Name	Contents	Date
Productname_orange_1	content_orange_1	May 2, 2017, 3:05 a.m.
Productname_orange_2	Content_orange_2	May 2, 2017, 3:05 a.m.
Productname_orange_3	Content_orange_3	May 2, 2017, 3:05 a.m.
Productname_hanlabong_1	Contents_hanlabong_1	May 2, 2017, 2:02 p.m.
Productname_hanlabong_2	Contents_hanlabong_2	May 2, 2017, 2:02 p.m.
Productname_hanlabong_3	Contents_hanlabong_3	May 2, 2017, 2:03 p.m.

Product 페이지



PRODUCTNAME_ORANGE_1

천혜향01



PRODUCTNAME_HANLABONG_1

한라봉01



PRODUCTNAME_HANLABONG_2

한라봉02



PRODUCTNAME_ORANGE_1

천혜향01



PRODUCTNAME_ORANGE_2

천혜향02



Product 세부 페이지

Orange 상세 페이지

Orange 상세 페이지

제주해커톤 웹페이지 구성

(완성페이지 <https://jejuhackathon.run.goorm.io/>)

(완성페이지는 해커톤 진행 중에만 실행됩니다.)

Sourcefile : jecodinghackathon.paulabedu.synology.me/download.zip

===== 기초과정 =====

1. <http://사용자이름.run.goorm.io/>

- 메인페이지, 서비스, 포트폴리오, 연혁, 팀 소개, 지도, 제품소개, contact
- Template : index.html

2. <http://사용자이름.run.goorm.io/product>

- 메인페이지에서 제품소개를 눌렀을 때 이동, 제품 List page
- Template : product.html

3. <http://사용자이름.run.goorm.io/제품번호>

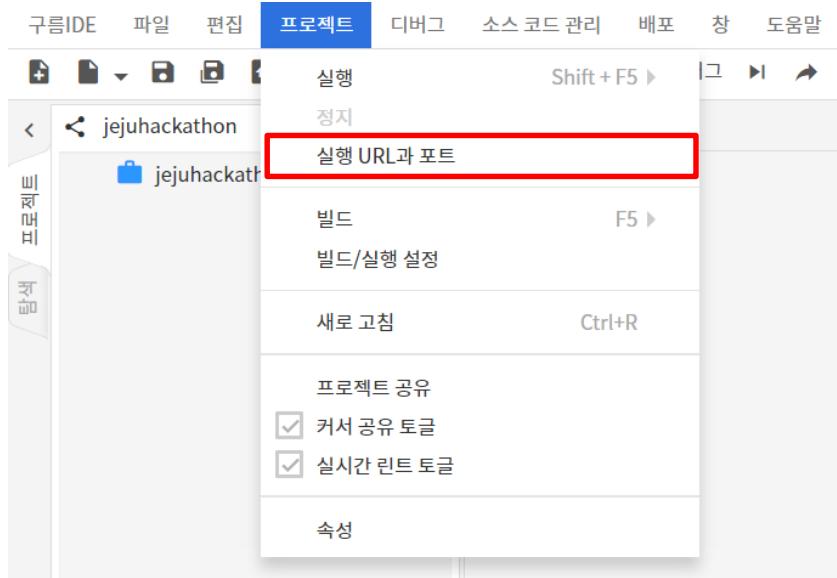
- product 페이지에서 제품을 눌렀을 때 이동
- Template : productdetails.html

===== 1/2기과정 =====

4. <http://사용자이름.run.goorm.io/태그이름>

- 2단계에서 태그 버튼을 눌렀을 때 태그 페이지로 이동하도록 구성
- 3단계에서 태그를 클릭하였을 때 태그 페이지로 이동하도록 구성
- Template : tag.html

구름IDE 환경 설정 (<https://ide.goorm.io/>)



프로젝트 속성

속성

정보

실행 URL과 포트

소스 코드 관리

Ignore 관리

빌드/실행 설정

paullab

실행 URL과 포트

서버 실행에 사용되는 도메인 주소를 등록할 수 있습니다.
\$USER_PORT (좌측 메뉴 > 플러그인 > 실행 옵션) 가 설정한 포트번호로 바뀌어서 서버가 실행됩니다.

URL: https:// **jejuhackathon** .run.goorm.io

PORT: 80

등록

URL은 원하시는 이름으로 작성해주세요.

이 부분은 오류가 생기실 경우 `python manage.py runserver 0:80` 명령어를 치기 전에 하셔도 됩니다.

Nginx와 다른 port로 하셔야 합니다.

구름 IDE 환경 설정(Django만 사용하실 경우)



아래 명령은 ④ 번 콘솔창에서 입력하세요.

```
root@goorm:/workspace/여러분ID# python --version
```

```
root@goorm:/workspace/여러분ID# mkdir mysite
```

```
root@goorm:/workspace/여러분ID# cd mysite
```

```
root@goorm:/workspace/여러분ID/mysite# apt-get install python-virtualenv
```

```
root@goorm:/workspace/여러분ID/mysite# virtualenv --python=python3.4 myvenv
```

```
root@goorm:/workspace/여러분ID/mysite# source myvenv/bin/activate
```

```
(myvenv)root@goorm:/workspace/여러분ID/mysite# pip install django whitenoise
```

```
(myvenv)root@goorm:/workspace/여러분ID/mysite# django-admin startproject 컨테이너이름 .
```

```
(myvenv)root@goorm:/workspace/여러분ID/mysite# python manage.py migrate
```

Startproject jejuhackathon 뒤에 '.'이 있으니 주의해 주세요.

구글 IDE 환경 설정 (TensorFlow를 함께 사용하실 경우)

아래 명령은 ④ 번 콘솔창에서 입력하세요.

(TensorFlow를 사용하실 경우의 명령어입니다.)

Django만 사용하실 분은 위에 페이지를 참고해주세요.)

다른 환경에서 설치하실 경우 <https://www.tensorflow.org/install/> 홈페이지를 참고하여 배포 확인을 꼭 하세요. (우분투나 파이썬 버전에 따라 TensorFlow가 작동하지 않을 수 있습니다.)

```
root@goorm:/workspace/여기ID# export TF_BINARY_URL=https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-0.12.1-cp34-cp34m-linux_x86_64.whl
```

```
root@goorm:/workspace/여기ID# sudo pip3 install --upgrade $TF_BINARY_URL
```

```
root@goorm:/workspace/여기ID# apt-get install python-virtualenv
```

```
root@goorm:/workspace/여기ID# virtualenv --system-site-packages ~tensorflow
```

```
root@goorm:/workspace/여기ID# source ~/tensorflow/bin/activate
```

```
(tensorflow)root@goorm:/workspace/여기ID# source ~/tensorflow/bin/activate
```

```
(tensorflow) root@goorm:/workspace/여기ID# mkdir mysite
```

```
(tensorflow) root@goorm:/workspace/여기ID# cd mysite
```

```
(tensorflow) root@goorm:/workspace/여기ID/mysite# pip3 install django whitenoise
```

```
(tensorflow) root@goorm:/workspace/여기ID/mysite# django-admin startproject kmleai .
```

```
(tensorflow) root@goorm:/workspace/여기ID/mysite# python manage.py migrate
```

django-admin startproject kmleai 뒤에 '.'이 있으니 주의해 주세요.

명령어 해석

```
root@goorm:/workspace/여기분ID# python --version
```

Python의 버전을 확인합니다. 2.x버전으로 되어있지요? 우리는 3.4 버전으로 개발을 할 것입니다.

```
root@goorm:/workspace/여기분ID# mkdir mysite
```

mkdir은 디렉토리를 만들어주는 명령어입니다. mysite라는 폴더를 만들 것입니다.

```
root@goorm:/workspace/여기분ID# cd mysite
```

cd명령어는 change directory 명령어입니다. 우리가 앞서 만든 mysite라는 폴더로 이동합니다.

```
root@goorm:/workspace/여기분ID/mysite# apt-get install python-virtualenv
```

가상환경 설정을 위한 명령어입니다. 가상환경이란 현재 2.x 버전으로 실행되고 있는 환경을 마치 3.x 버전으로 쓰기凫 만들어주는 가상의 환경이에요.

```
root@goorm:/workspace/여기분ID/mysite# virtualenv --python=python3.4 myvenv
```

3.4버전으로 사용을 하겠다는 것입니다.

```
root@goorm:/workspace/여기분ID/mysite# source myvenv/bin/activate
```

이 명령어를 실행하고 앞서 실행한 Python --version을 실행하면 3.4버전으로 나옵니다.

이 명령어는 구름IDE 컨테이너를 다시 실행할때마다 카주어야 합니다.

```
(myvenv)root@goorm:/workspace/여기분ID/mysite# pip install django whitenoise
```

파이썬까지 설치하기 위한 명령어입니다. Django만 설치해주셔도 됩니다.

```
(myvenv)root@goorm:/workspace/여기분ID/mysite# django-admin startproject 컨테이너이름.
```

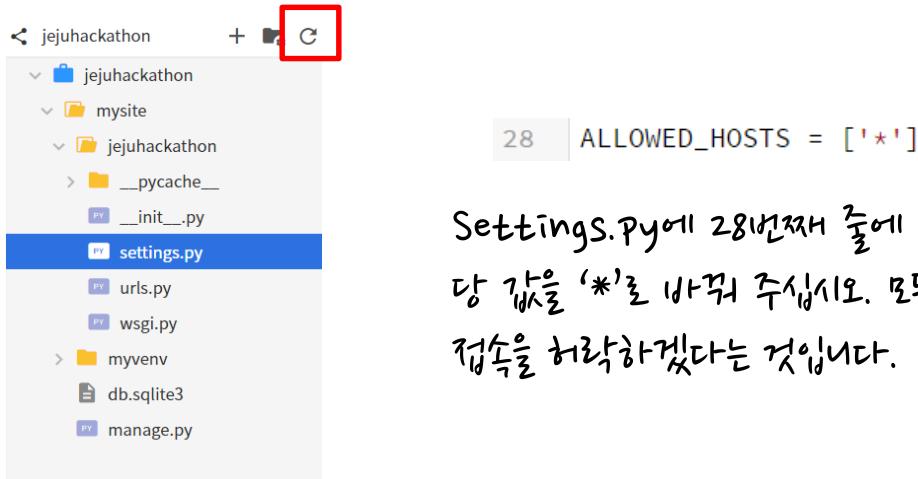
현재 있는 폴더에 프로젝트를 시작하겠다는 명령어에요.

```
(myvenv)root@goorm:/workspace/여기분ID/mysite# python manage.py migrate
```

Migrate 명령어는 하면에서 설명해 드리도록 하겠습니다. 쉽게 말해 DB에 값을 넣는 작업입니다.

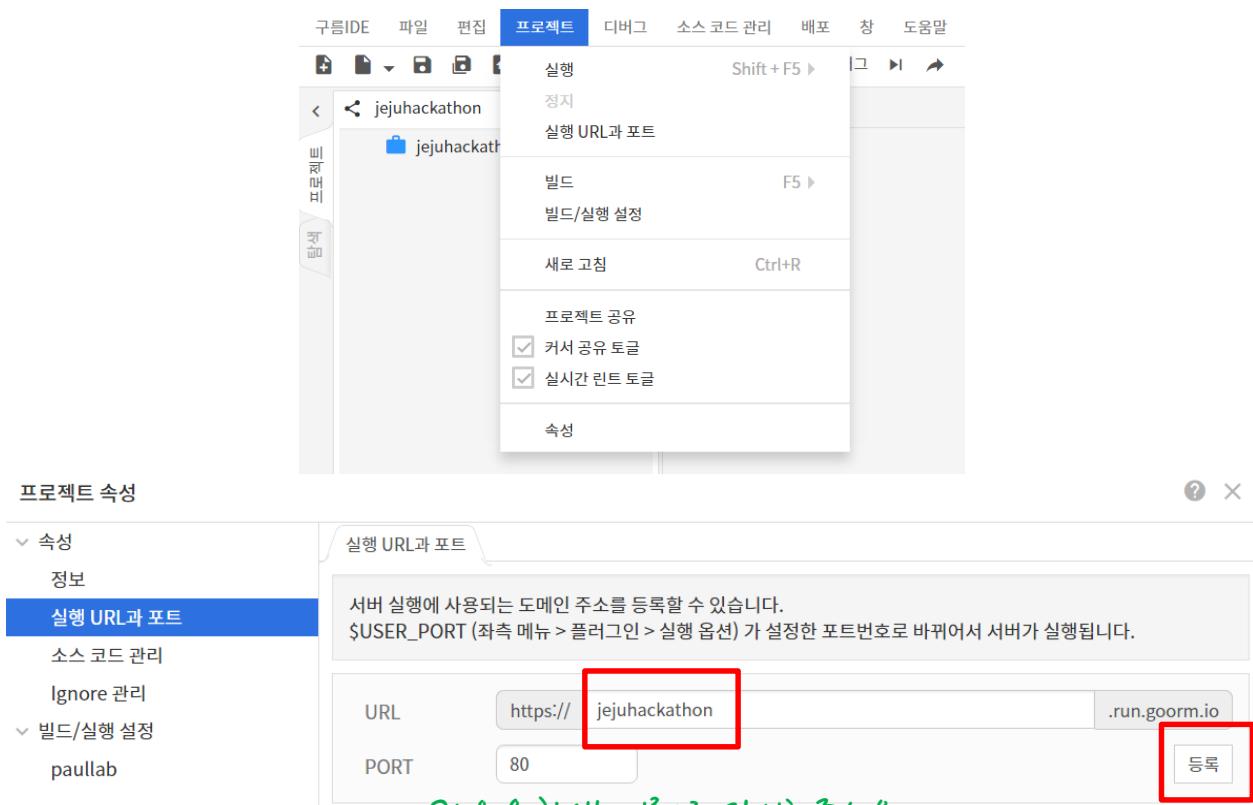
Startproject jejuhackathon 뒤에 '.'이 있으니 주의해 주세요.

구름IDE 환경 설정 (<https://ide.goorm.io/>)



```
(myvenv)root@goorm:/workspace/여러분ID/mysite# python manage.py runserver 0.0.0.0:80
```

위 명령어를 치시기 전 실행 URL과 포트 부분에서 오류가 나셨다면 아래와 같이 다시 시도해 주세요.



URL은 원하시는 이름으로 작성해주세요.

구름 IDE 환경 설정 (<https://ide.goorm.io/>)

The screenshot shows the '실행 URL과 포트' (Execution URL and Port) section of the Goorm Cloud IDE configuration. It includes fields for 'URL' (https://run.goorm.io) and 'PORT' (80 ~ 65000), a '등록' (Register) button, and a table for registered URLs and ports. The table has columns for '실행용' (Execution), 'URL', 'Port', and '삭제' (Delete). A row is selected with the URL <https://jejuhackathon.run.goorm.io>, which is highlighted with a red rectangle.

실행용	URL	Port	삭제
<input checked="" type="radio"/>	https://jejuhackathon.run.goorm.io	80	

프로젝트 > 실행 URL과 포트에서 등록하신 URL을 클릭해주세요.

The screenshot shows a success message box with the title 'It worked!' and the text 'Congratulations on your first Django-powered page.' Below the message, there are two lines of smaller text: 'Of course, you haven't actually done any work yet. Next, start your first app by running `python manage.py startapp [app_label]`.' and 'You're seeing this message because you have `DEBUG = True` in your Django settings file and you haven't configured any URLs. Get to work!'

작동이 확인되면 터미널에서 `ctrl + c`를 눌러 작동을 중지해 주세요.

다음과 같이 작동한다면 성공한 것입니다!

Step 2

기본 환경설정

환경설정 - settings.py

```
root@goorm:/workspace/여제ID/mysite# Python manage.py startapp product
```

The screenshot shows a file explorer on the left with the following structure:

- jejuhackathon (Folder)
 - __pycache__ (Folder)
 - __init__.py (File)
 - settings.py (File, highlighted in blue)
 - urls.py (File)
 - wsgi.py (File)

On the right is a code editor displaying the contents of settings.py:

```
33 INSTALLED_APPS = (
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'product'
41 )
```

The line 'product' is highlighted with a red rectangle.

Django는 여러 개의 앱을 동시에 만들 수 있습니다. 우리는 product라는 빈 앱을 만들었으니 앱 목록에 추가해야 합니다. 이 작업을 하지 않는다면 앱이 구동되지 않습니다.

환경설정 - urls.py

mysite > jejuhackathon > urls.py

```
16 from django.conf.urls import url
17 from django.contrib import admin
18 from product.views import index
19
20 urlpatterns = [
21     url(r'^admin/', admin.site.urls),
22     url(r'^$', index),
23 ]
```

작성하시면 18번째 줄이 배경색으로 변할 것입니다. 이것은 우리가 index라는 함수를 만들어주지 않아 생기는 현상입니다. 뒤에서 index를 만들어주면 없어집니다.

초기설정 - views.py

mysite > product > views.py

```
1 from django.shortcuts import render
2
3 # Create your views here.
4 def index(request):
5     return render(request, 'product/index.html')
```

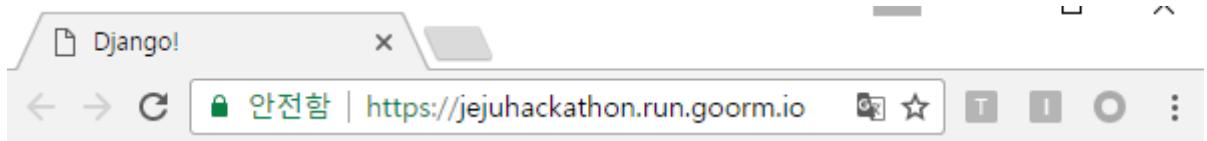
초기설정 - index.html

mysite > product > templates > product > index.html

```
<html>
<head>
    <title>Django!</title>
</head>
<body>
    <h1>Test!</h1>
</body>
<html>
```

환경설정

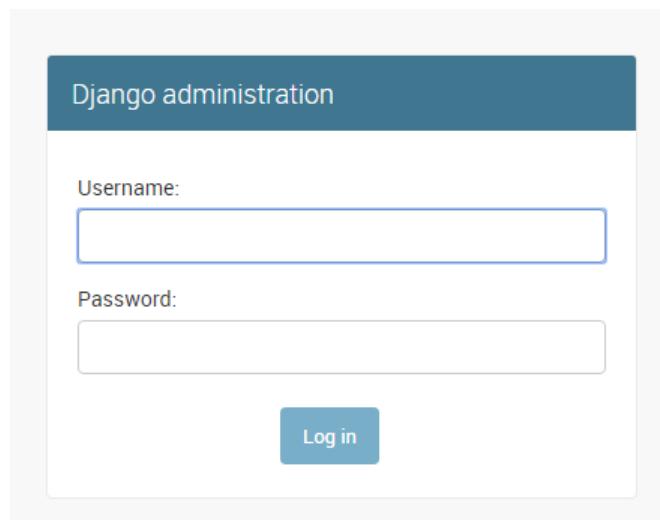
```
(myvenv)root@goorm:/workspace/여기ID/mysite# python manage.py runserver 0.0.0.0:80
```



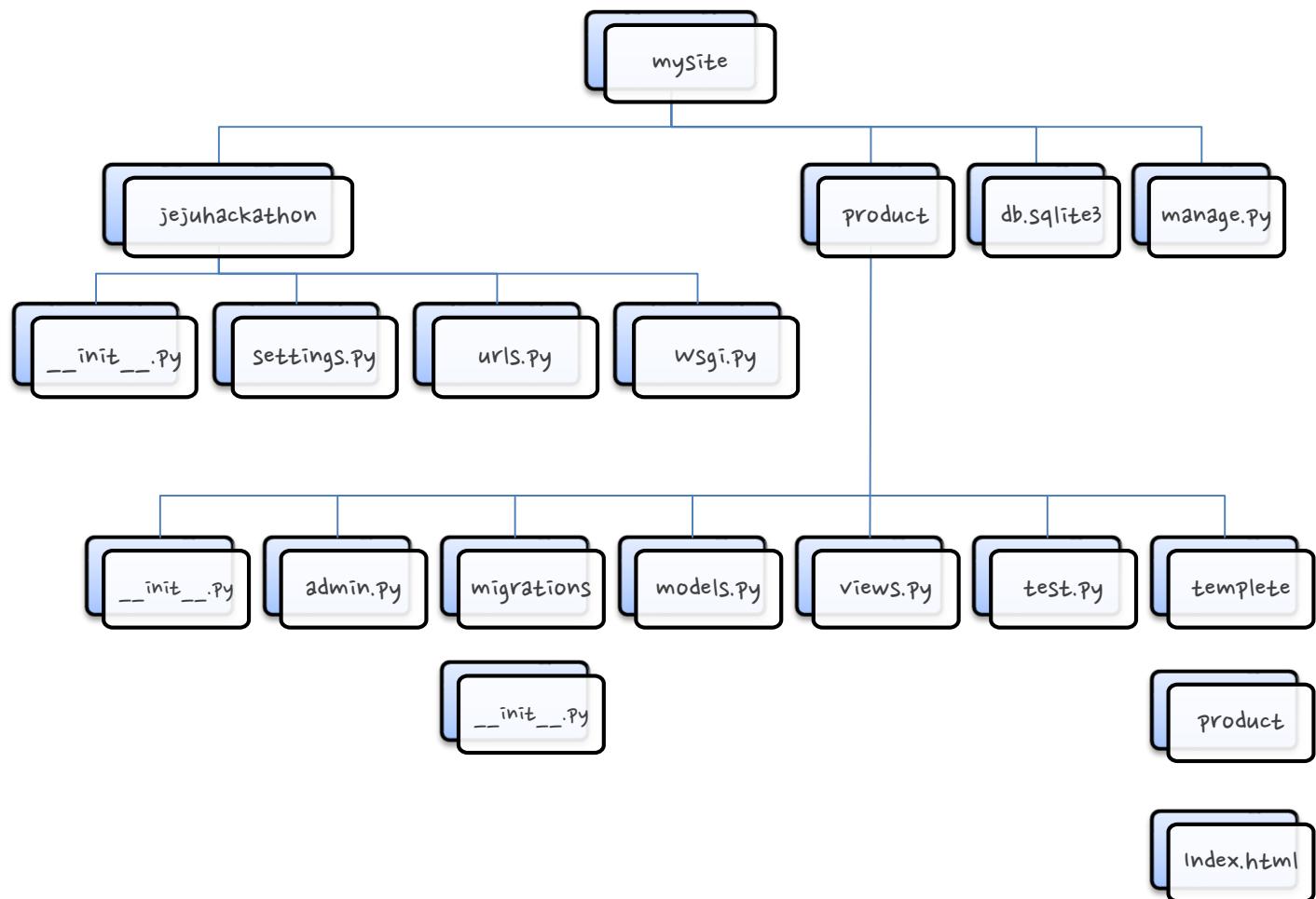
Test!

환경설정 - admin

<https://jejuhackathon.run.goorm.io/admin>

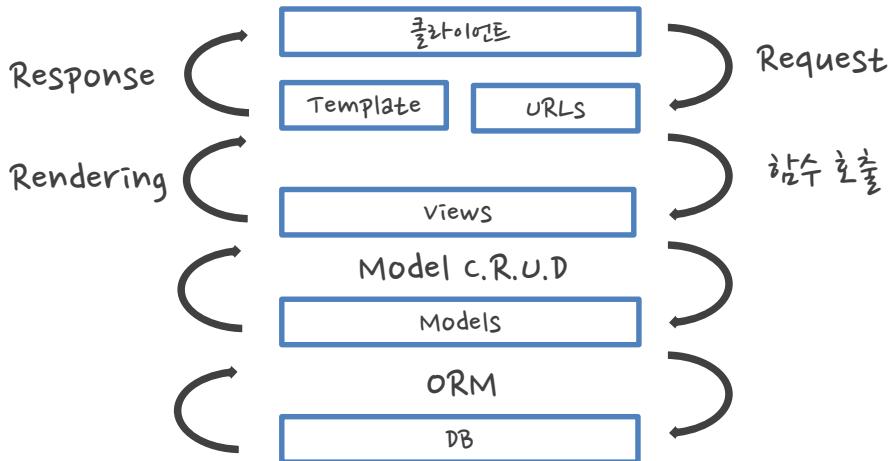


현재 폴더 구성



Django 구성

MTV



Model - DB와 연결된 Python class

Template - 사용자에게 response를 client view

View - Django에서 처리한 데이터를 Template에게 전달

각 문자는 다음과 같이 표준 SQL문으로 대응 가능하다.

이름	조작	SQL
Create	생성	INSERT
Read(또는 Retrieve)	읽기(또는 인출)	SELECT
Update	갱신	UPDATE
Delete(또는 Destroy)	삭제(또는 파괴)	DELETE

CRUD는 대부분의 컴퓨터 소프트웨어가 가지는 기본적인 데이터 처리 기능인 create(생성), Read(읽기), update(갱신), Delete(삭제)를 뜻어서 일컫는 말

<https://ko.wikipedia.org/wiki/CRUD>

Step 3

부트스트랩을 이용한 Django 배포

Step3_Django에 부트스트랩 적용

The recommended CDN for Bootstrap, Font Awesome and Bootswatch.

Star 682 Follow Follow Tweet

Quick Start

Complete CSS
https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css

Complete JavaScript
https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js

Why BootstrapCDN Believes in Subresource Integrity (SRI)

Mad Love

Alejandro Such @alejandro_such BootstrapCDN Delivers Twitter's Bootstrap Globally for Free zite.to/11u2LNz via @zite 6:46 PM - 9 Feb 2013 1 2

MaxCDN Developer 27 Dec 12 @MaxCDNDeveloper @B3nicio ? email me jdorfman at netdna dot com

<https://www.bootstrapcdncdn.com/>

Google에서 bootstrap CDN을 검색하시면 다음과 같은 사이트가 나옵니다. CSS 파일을 직접 서버에 올리지 않고 링크로 쓸 수 있도록 만들 수 있어요. 여기서는 CSS만 활용해 보도록 하겠습니다.

mysite > product > templates > product > index.html

```
1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>듀토리얼</title>
6     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
7   </head>
8   <body>
9     <div class="container">
10       <div class="row">
11
12         <div class="col-sm-6">
13           <h1>안녕하세요.<h1>
14         </div>
15
16         <div class="col-sm-6">
17           <h1>안녕하세요.<h1>
18         </div>
19
20       </div>
21     </div>
22   </body>
23 </html>
```

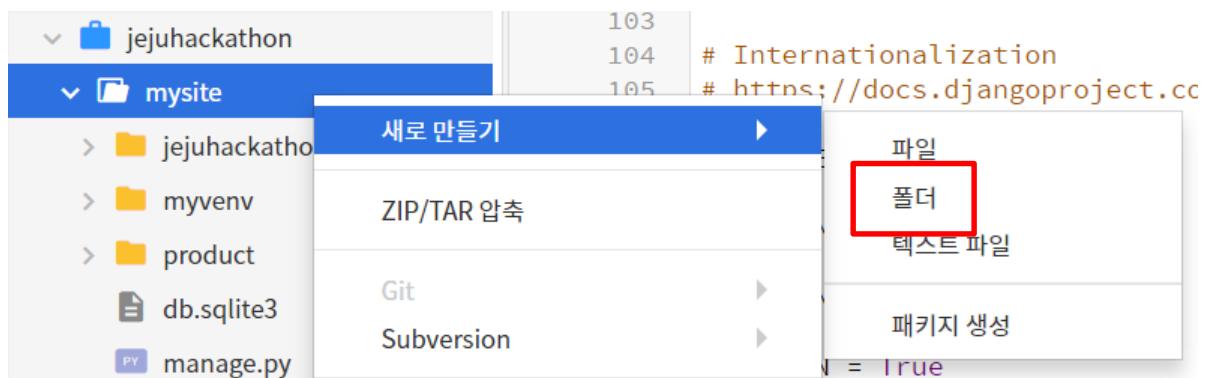
Step_3_1.html

Step3_Django에 부트스트랩 적용

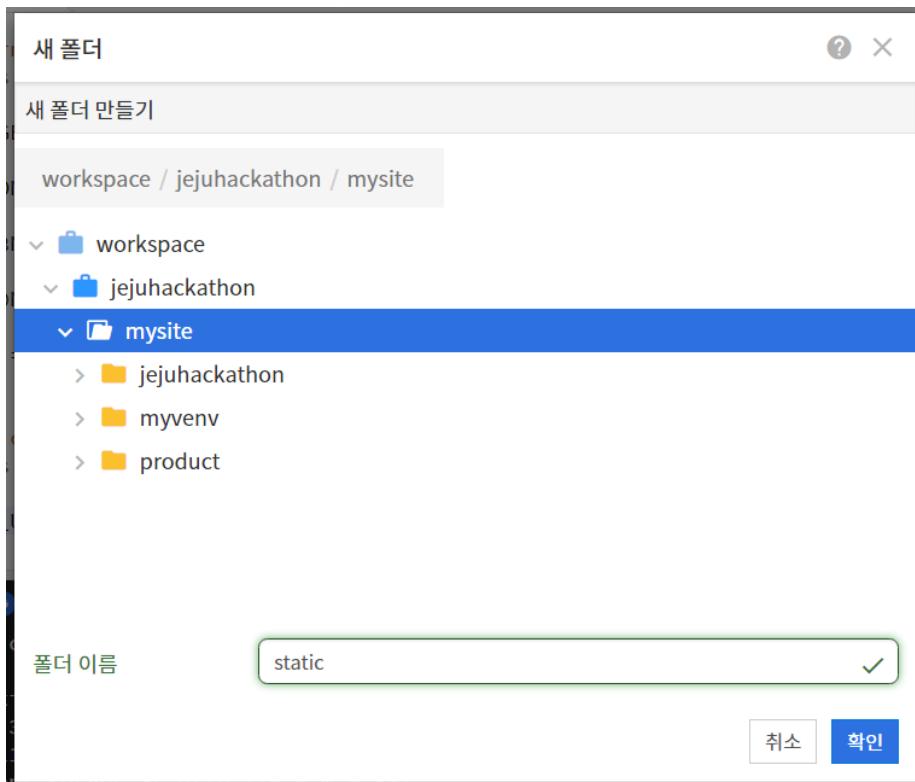


Bootstrap은 12컬럼으로 홈페이지 전체 크기를 나눕니다. 반응형 홈페이지(크기 조절에 따라 홈페이지 모습이 바뀐다는)이며 여러가지 부가 기능들을 제공하고 있어요 자세한 사항은 <http://bootstrapk.com/>를 참고하세요. 우리는 위에서 6컬럼씩 할당해 주었기 때문에 반씩 나누어 출력됩니다. 만약 'hello!!'가 밑으로 출력이 되었다면 부트스트랩이 작동하지 않고 있다는 것인 다시 한번 파일을 확인해주세요.

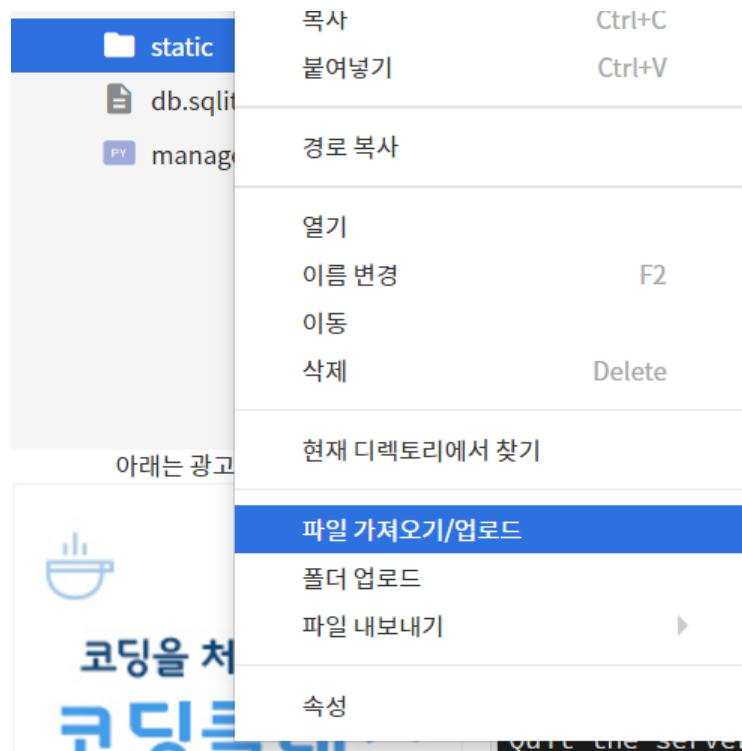
Step3_이미지 파일 불러오기



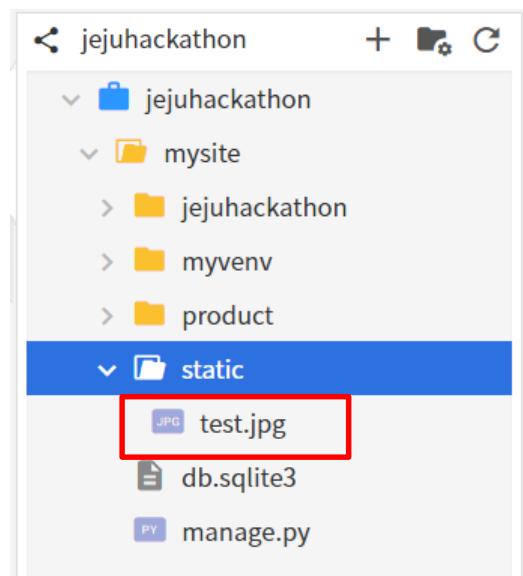
Step3_이미지 파일 올려오기



Step3_이미지 파일 올려오기



Step3_ 이미지 파일 불러오기



test.jpg는 공유해드린 폴더 내에 있습니다.

Step3_이미지 파일 불러오기

```
118 # Static files (CSS, JavaScript, Images)
119 # https://docs.djangoproject.com/en/1.11/howto/static-files/
120
121 STATIC_URL = '/static/'
122 STATICFILES_DIRS = (
123     os.path.join(BASE_DIR, 'static'),
124 )
```

Static file를 연결하기 위해 다음과 같은 경로명 설정을 해줘야 합니다.

Step3_timezone 설정하기

```
109 TIME_ZONE = 'Asia/Seoul'
110
111 USE_I18N = True
112
113 USE_L10N = True
114
115 USE_TZ = True
116
117
118 # Static files (CSS, JavaScript, Images)
119 # https://docs.djangoproject.com/en/1.11/howto/static-files/
120
121 STATIC_URL = '/static/'
122 STATICFILES_DIRS = (
123     os.path.join(BASE_DIR, 'static'),
124 )
```

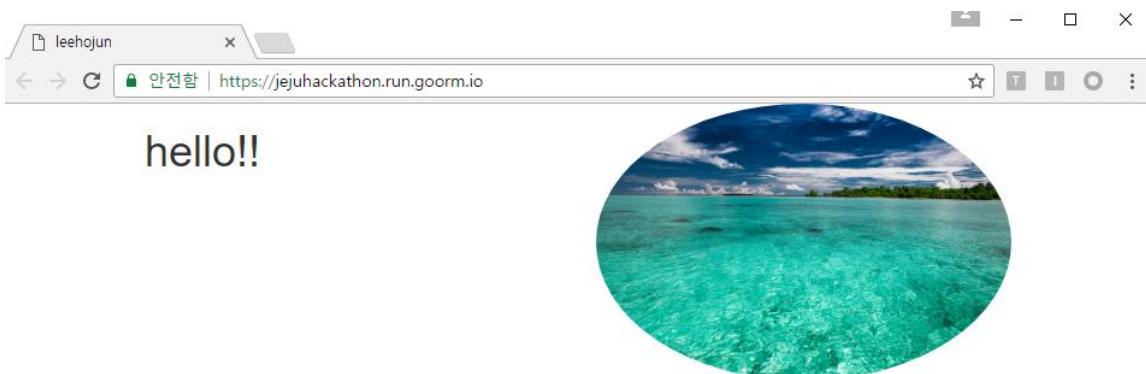
Step3_ 이미지 파일 불러오기

mySite > product > templates > product > index.html

```
1 <!doctype html>
2 <html>
3   <head>
4     <title>leehojun</title>
5     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
6   </head>
7   <body>
8     <div class="container">
9       <div class="row">
10
11         <div class="col-sm-6">
12           <h1>hello!!</h1>
13         </div>
14
15         <div class="col-sm-6">
16           {% load staticfiles %}
17           
18         </div>
19       </div>
20     </body>
21 </html>
```

Step_3_2.html

{% 문법 %}의 구조는 **template tag**라고 불립니다. 자세한 내용은 Django 공식문서 (<https://docs.djangoproject.com/en/1.11/ref/templates/builtins/>)를 참고하세요. 입출력 파일 안에도 들어가 있습니다. 또한 앞으로는 html 파일은 직접 타이핑하시지 않고 붙여넣는 식으로 웹 서비스를 개발하도록 하겠습니다.



실행화면

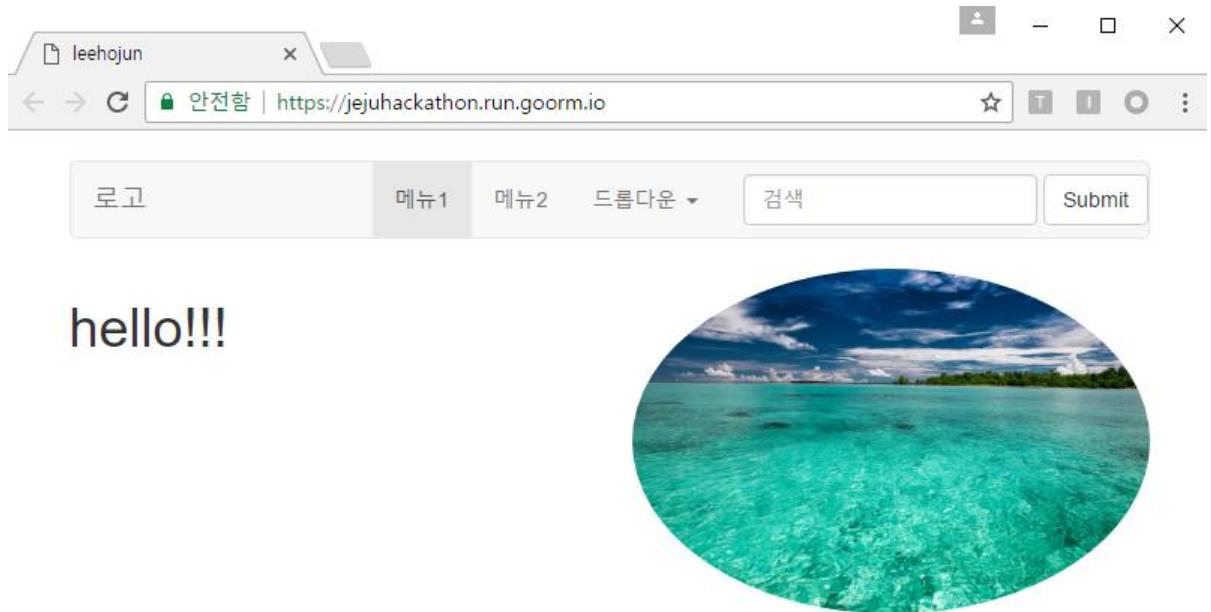
Step3_메뉴 만들기

mySite > product > templates > product > index.html

```
1 <!doctype html>
2 <html>
3   <head>
4     <title>leehojun</title>
5     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
6     <style>
7       body { padding-top: 20px; }
8     </style>
9   </head>
10  <body>
11    <div class="container">
12
13    <nav class="navbar navbar-default" role="navigation">
14      <!-- Brand and toggle get grouped for better mobile display -->
15      <div class="navbar-header">
16        <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-ex1-collapse">
17          <span class="sr-only">Toggle navigation</span>
18          <span class="icon-bar"></span>
19          <span class="icon-bar"></span>
20          <span class="icon-bar"></span>
21        </button>
22        <a class="navbar-brand" href="#">로고 </a>
23      </div>
24
25      <!-- Collect the nav links, forms, and other content for toggling -->
26      <div class="collapse navbar-collapse navbar-right navbar-ex1-collapse">
27        <ul class="nav navbar-nav">
28          <li class="active"><a href="#">메뉴1 </a></li>
29          <li><a href="#">메뉴2</a></li>
30          <li class="dropdown">
31            <a href="#" class="dropdown-toggle" data-toggle="dropdown">드롭다운 <b class="caret"></b></a>
32            <ul class="dropdown-menu">
33              <li><a href="#">서브메뉴 1</a></li>
34              <li><a href="#">서브메뉴 2</a></li>
35              <li><a href="#">서브메뉴 3</a></li>
36            </ul>
37          </li>
38        </ul>
39        <form class="navbar-form navbar-right" role="search">
40          <div class="form-group">
41            <input type="text" class="form-control" placeholder="검색">
42          </div>
43          <button type="submit" class="btn btn-default">Submit</button>
44        </form>
45
46      </div><!-- /.navbar-collapse -->
47    </nav>
48  </div> <!-- container 끝 -->
49  <div class="container">
50    <div class="row">
51
52      <div class="col-sm-6">
53        <h1>Hello!!</h1>
54      </div>
55
56      <div class="col-sm-6">
57        {% load staticfiles %}
58        
59      </div>
60    </div>
61  </body>
62 </html>
```

Step_3_3.html

Step3_ 메뉴 만들기



hello!!!

mySite > product > templates > product > index.html

```
1 <!DOCTYPE html>
2 <html lang="ko">
3
4 <head>
5
6   <meta charset="utf-8">
7   <meta http-equiv="X-UA-Compatible" content="IE=edge">
8   <meta name="viewport" content="width=device-width, initial-scale=1">
9   <meta name="description" content="">
10  <meta name="author" content="">
11
12  <title>Jeju On Coding</title>
13
14  <!-- Bootstrap Core CSS -->
15  <link href="http://1.paullabedu.synology.me/css/bootstrap.min.css" rel="stylesheet">
16
17  <!-- Custom CSS -->
18  <link href="http://1.paullabedu.synology.me/css/agency.css" rel="stylesheet">
19
20  <!-- Custom Fonts -->
21  <link href="http://1.paullabedu.synology.me/font-awesome/css/font-awesome.min.css" rel="stylesheet"
      type="text/css">
```

Step_3_4.html

실행화면

Jeju Coding Hackathon

SERVICES PORTFOLIO HISTORY TEAM PRODUCT CONTACT

Welcome To 'Jeju Coding Hackathon' Studio!

IT'S NICE TO MEET YOU

TELL ME MORE



2020-2030

회사의 설립

하고 싶은 사업과 포부를 써 넣으세요. 최대한 구체적으로 써 주세요.



MARCH 2031

회사가 날로 성장합니다.



DECEMBER 2032

회사가 날로 성장합니다.

어떻게 성장해가는지 서울에 주세요. 나는 어떤 모습일지 그려보세요.



JULY 2033

회사가 날로 성장합니다.

어떻게 성장해가는지 서울에 주세요. 나는 어떤 모습일지 그려보세요.



여러의자로
여러분의
DREAM!

SERVICES

jeju Coding Hackathon은 무료로 진행되는 코딩 해커톤입니다.



SW 역량강화

"Jeju Coding Hackathon"은 편상설 제주도에서 그동안 만나보지 못한 고급 SW 개발 경험을 제공합니다.



SW 인맥강화

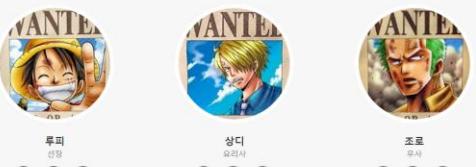
"Jeju Coding Hackathon"은 제주도의 지역 특성을 극복하고 이 분야에 다양한 인맥을 사귈 수 있도록 지원합니다.



지속가능 SW 생사

"Jeju Coding Hackathon"은 Facebook 그룹 "제주코딩해커톤"으로 매년 발표자료 모집하여 지속적인 모임을 제공합니다.

꿈을 이루려면 동료가 있어야 합니다.



투피 신장
상디 요리사
조로 무사

좋아하는 애니메이션이나 영화에서 여러분들이 꿈을 함께 해줄 동료를 떠올려 보세요.



PORTFOLIO

이곳에 넣고 싶은 사진과 링크를 넣어주세요.



원하시는 이미지를 넣으세요.
바로 적용 됩니다.



원하시는 이미지를 넣으세요.
바로 적용 됩니다.



원하시는 이미지를 넣으세요.
바로 적용 됩니다.



원하시는 이미지를 넣으세요.
바로 적용 됩니다.



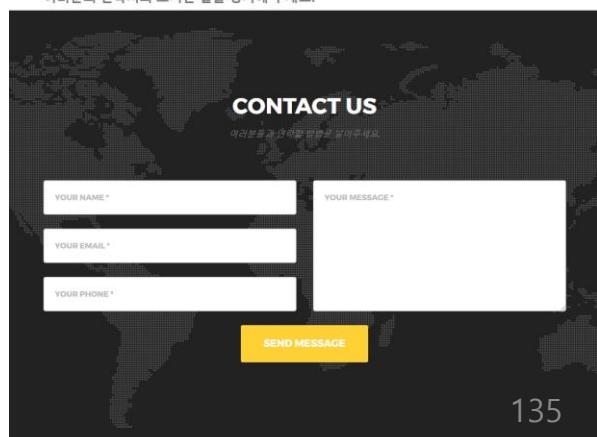
원하시는 이미지를 넣으세요.
바로 적용 됩니다.



원하시는 이미지를 넣으세요.
바로 적용 됩니다.

연락처

여러분의 연락처와 오시는 길을 명시해 주세요.



CONTACT US

여러분과 함께 일할 것을 기다리겠습니다.

YOUR NAME *

YOUR EMAIL *

YOUR PHONE *

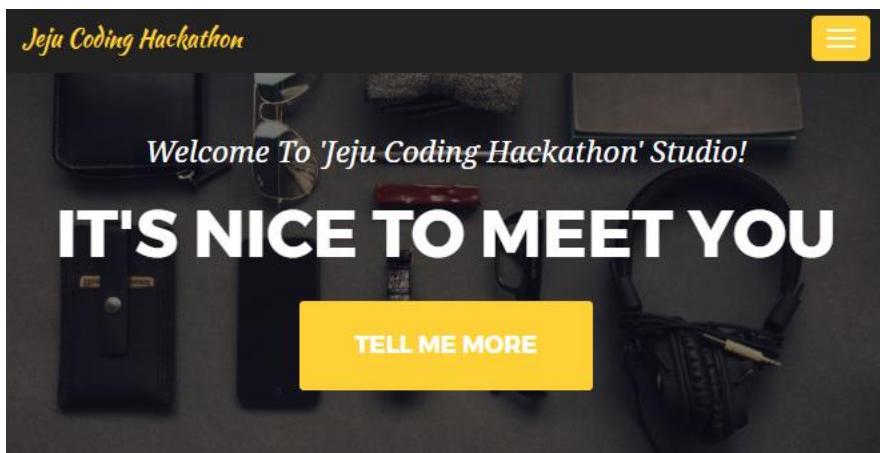
YOUR MESSAGE *

SEND MESSAGE

135

부트스트랩 설정_텍스트&아이콘

1. 수정할 문구만 찾아서 수정하는 방법(아무것도 모르실 경우 사용)
2. 부트스트랩 기본 문법을 알고 커스터マイ징하는 방법(권장해 드립니다)



SERVICES

Jeju Coding Hackathon은 무료로 진행되는 코딩 해커톤입니다.

```
Step_3_4 - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<!DOCTYPE html>
<html lang="ko">

<head>
```

찾기

찾을 내용(N): |

다음 찾기(F) 취소

방향

위로(U) 아래로(D)

대/소문자 구분(C)

```
atible
nt="wi
ntent=
="">
```

```
<title>Jeju Coding Hackathon</tit
```

```
<!-- Bootstrap Core CSS -->
```

부트스트랩 설정 _ 텍스트&아이콘



Font Awesome is fully open source and is GPL friendly. You can use it for commercial projects, open source projects, or really just about whatever you want.

★ JUPITER
The World's Favorite Font Icons
26,000 users can't be wrong. Join them and download Jupiter today.
ads via Carbon

<http://fontawesome.io/> 이용하세요.

- 이미지가 아닌 폰트 아이콘으로 되어 있기 때문에 글자만 수정해주시면 모습이 변합니다. <http://fontawesome.io/> > icon에서 원하시는 아이콘을 클릭하세요



After you get [up and running](#), you can place Font Awesome icons just about anywhere with the `<i>` tag:

`<i class="fa fa-coffee" aria-hidden="true"></i>`

```
<i class="fa fa-coffee" aria-hidden="true"></i>
```

Note: to improve [web accessibility](#), we recommend using `aria-hidden="true"` to hide icons used purely for decoration.

부트스트랩 설정 _ 텍스트&아이콘

```
<div class="col-md-4">
  <span class="fa-stack fa-4x">
    <i class="fa fa-circle fa-stack-2x text-primary"></i>
    <i class="fa fa-laptop fa-stack-1x fa-inverse"></i>
  </span>
  <a href='http://www.paullab.co.kr/'><h4 class="service-heading"><span style="color:black">SW역량강화</span></h4></a>
  <p class="text-muted">“Jeju Coding Hackathon”은 환상섬 제주도에서 그동안 만나보지 못한 고급 SW 개발 경험을 제공합니다.</p>
</div>
```

Step_3_4.html

- 앞서 복사한 text를 배울간색 사각형이 있는 곳에 넣어주세요. 그러면 아래와 같이 아이콘이 바뀝니다.



SW역량강화

“Jeju Coding Hackathon”은 환상섬 제주도에서 그동안 만나보지 못한 고급 SW 개발 경험을 제공합니다.

해커톤 과제

1. 어떤 서비스를 만들 것인지 문서로 작성
2. 홈페이지는 원하는 모습으로 꾸려오기
(이미지와 컨텐츠 준비)
3. 그 밖에 필요한 기능적인 측면 정의해 오기
(요구사항 정의서 준비)

1. 자료 찾는 법 : google hacking
2. 5월 6일까지 paul-lab@naver.com로 보내주시면 의견 보내드리겠습니다.

Step 4

Product page 만드기

Step4_product 페이지 만들기

```
16 from django.conf.urls import url
17 from django.contrib import admin
18 from product.views import index, product
19
20 urlpatterns = [
21     url(r'^admin/', admin.site.urls),
22     url(r'^$', index),
23     url(r'^product', product),
24 ]
```

이번에는 product페이지를 만들어 보도록 하겠습니다.

```
1 from django.shortcuts import render
2
3 # Create your views here.
4 def index(request):
5     return render(request, 'product/index.html')
6
7 def product(request):
8     return render(request, 'product/product.html')
```

한번 해보셨으니 product.html을 만들어야 하실다는 것을 알 수 있습니다.

Step4_product 페이지 만들기

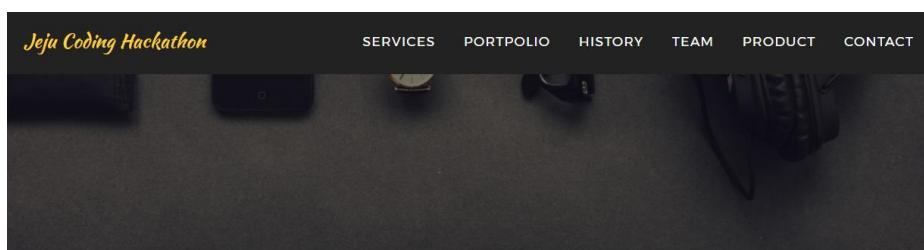
jejuhackathon/mysite/product/templates/product.html

```
1 <!DOCTYPE html>
2 <html lang="ko">
3
4 <head>
5
6     <meta charset="utf-8">
7     <meta http-equiv="X-UA-Compatible" content="IE=edge">
8     <meta name="viewport" content="width=device-width, initial-scale=1">
9     <meta name="description" content="">
10    <meta name="author" content="">
11
12    <title>Jeju On Coding</title>
```

Step_3_5.html

이미 만들어진 html 파일을 복여주세요.

설명화면(<https://jejuhackathon.run.goorm.io/product>)



PRODUCT 소개 페이지입니다.

Step4_Product 테이블 만들기

소스코드

jejuhackathon/mysite/product/models.py

```
1 from django.db import models  
2  
3 # Create your models here.  
4 class Product(models.Model):  
5     productname = models.CharField(max_length=50)  
6     contents = models.TextField()
```

(myvenv)root@goorm:/workspace/여행날ID/mysite# python manage.py makemigrations product

(myvenv)root@goorm:/workspace/여행날ID/mysite# python manage.py migrate

jejuhackathon/mysite/product/admin.py

```
1 from django.contrib import admin  
2 from .models import Product  
3  
4 admin.site.register(Product)
```

Admin 화면에 등록합니다. 이 공간에서 admin 사이트를 꾸밀 수 있습니다.

Step4_product 페이지 만들기

```
(myvenv)root@goorm:/workspace/여기분ID/mysite# python manage.py createsuperuser
```

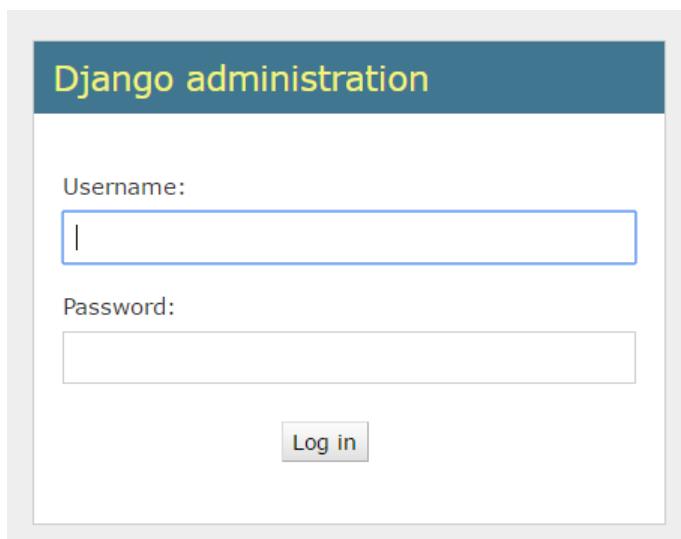
```
root@goorm:/workspace/jejuhackathon/mysite# python manage.py createsuperuser
System check identified some issues:

WARNINGS:
?: (1.7.W001) MIDDLEWARE_CLASSES is not set.
    HINT: Django 1.7 changed the global defaults for the MIDDLEWARE_CLASSES. django.contrib.sessions.middleware.SessionMiddleware,
          django.contrib.auth.middleware.AuthenticationMiddleware, and django.contrib.messages.middleware.MessageMiddleware were removed from the defaults. If your project needs these middleware then you should configure this setting.
Username (leave blank to use 'root'): jejuhackathon
Email address: jejuhackathon@gmail.com
Password:
Password (again):
Superuser created successfully.
root@goorm:/workspace/jejuhackathon/mysite#
```

Admin user를 만들 것입니다. 비밀번호는 원래 콘솔창에 안뜨니 알려주시고 엔터를 치시면 됩니다.

설명화면

<http://jejuhackathon.run.goorm.io/admin>



Step4_Product 페이지 만들기

실행화면

The screenshot shows the Django admin dashboard. In the top left, it says "Django administration". On the right, it says "Welcome, jejuhackathon. View site / Change password / Log out". Below that is a sidebar titled "Site administration" with sections for "Authentication and Authorization" (Groups, Users), "Product" (Products), and "Recent Actions" (My Actions: None available). Under "Product", there is a table with one row: "Products" with "Add" and "Change" buttons. The "Add" button is highlighted with a red rectangle.

제시물을 한번 살펴보도록 하겠습니다.

Add product

The screenshot shows the "Add product" form. It has two fields: "Productname" with the value "Productname_orange_1" and "Contents" with the value "content_orange_1". At the bottom, there are three buttons: "Save and add another", "Save and continue editing", and a blue "Save" button.

구글을 위해 꼭 productname과 contents의 내용을 다르게 해주세요.

Step4_Product 페이지 만들기

그런데 제목이 productname_onrange_1이 아닙니다. 제목을 title로 출력하라는 명령이 있었기 때문입니다. Models.py로 들어가서 다시 정의해줄게요.

The screenshot shows the Django admin 'Select product to change' page. At the top, it says 'Django administration' and 'Welcome, jejuhackathon'. Below that, there's a green success message: 'The product "Product object" was added successfully.' On the left, there's a sidebar with 'Action:' dropdown, 'Go' button, and two checkboxes: 'Product' and 'Product object'. The 'Product object' checkbox is checked. At the bottom left, it says '1 product'. On the right, there's a 'Add product +' button.

jejuhackathon/mysite/product/models.py

```
1 from django.db import models
2
3 # Create your models here.
4 class Product(models.Model):
5     productname = models.CharField(max_length=50)
6     contents = models.TextField()
7
8     def __str__(self):
9         return self.productname
```

Step4_Product 페이지 만들기

설명화면

Django administration

Welcome, jejuhackathon. View site / Change password / Log out

Home > Product > Products

Select product to change

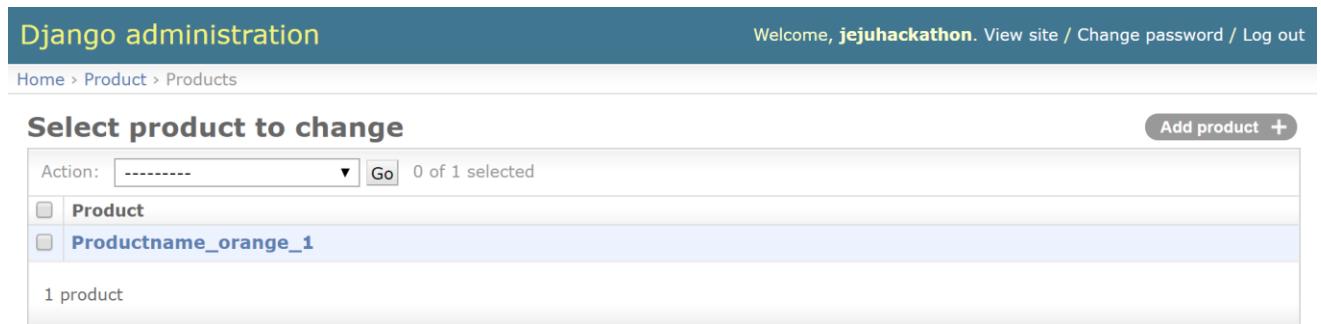
Add product +

Action: ----- ▼ Go 0 of 1 selected

Product

Productname_orange_1

1 product



제작물 추가

Django administration

Welcome, jejuhackathon. View site / Change password / Log out

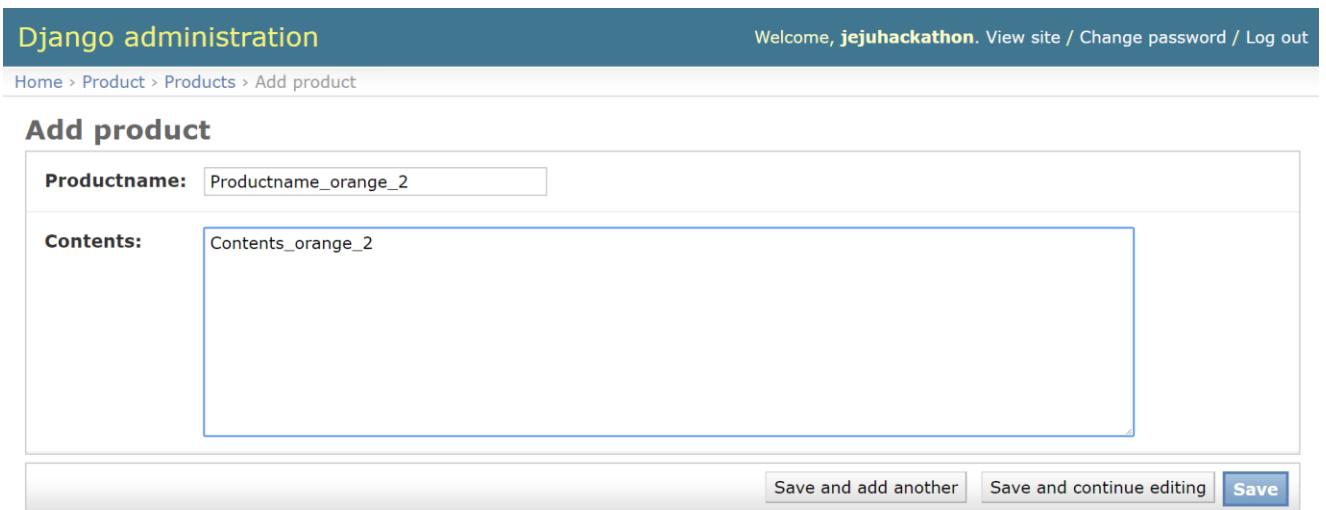
Home > Product > Products > Add product

Add product

Productname: Productname_orange_2

Contents: Contents_orange_2

Save and add another Save and continue editing Save



Step4_Product 페이지 만들기

Test를 위해 총 3개의 게시물을 작성해주세요.

The screenshot shows the Django admin 'Products' list page. At the top, it says 'Django administration' and 'Welcome, jejuhackathon'. Below that, there's a success message: 'The product "Productname_orange_3" was added successfully.' On the left, there's a sidebar with an 'Add product +' button. The main area lists three products:

Action	Product
<input type="checkbox"/>	Productname_orange_3
<input type="checkbox"/>	Productname_orange_2
<input type="checkbox"/>	Productname_orange_1

Below the table, it says '3 products'.

jejuhackathon/mysite/product/views.py

```
1 from django.shortcuts import render
2 from .models import Product
3
4 # Create your views here.
5 def index(request):
6     return render(request, 'product/index.html')
7
8 def product(request):
9     productlist = Product.objects.all()
10    return render(request, 'product/product.html', {'productlist':productlist,})
```

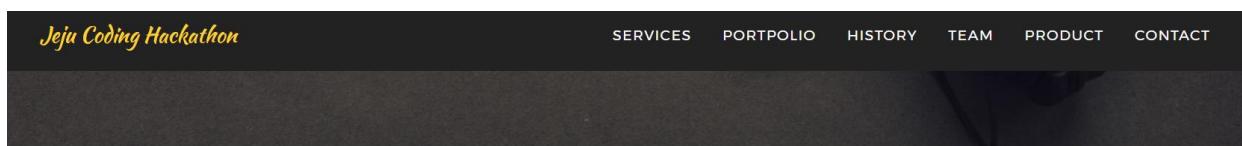
Step4_Product 페이지 만들기

```
95      <!-- Section -->
96  <section>
97    <div class="container">
98      <div class="row">
99        <div class="col-lg-12 text-center">
100          <table class="table table-hover table-bordered">
101            <tbody>
102              {% for list in productlist %}
103                <tr>
104                  <td>{{ list.productname }}</td>
105                  <td>{{ list.contents }}</td>
106                </tr>
107              {% endfor %}
108            </tbody>
109          </table>
110        </div>
111      </div>
112    </div>
113  </section>
```

Step_4_1.html

Template tag를 이용하여 DB에 입력된 내용들을 전부 가져옵니다. 자세한 내용은 <https://docs.djangoproject.com/en/1.11/ref/templates/builtins/>를 참고해주세요.

설명화면

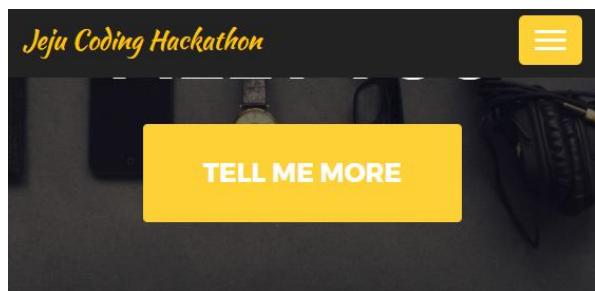


Productname_orange_1	content_orange_1
Productname_orange_2	Content_orange_2
Productname_orange_3	Content_orange_3

Step4_Product 페이지 만들기

설명화면

소스보기



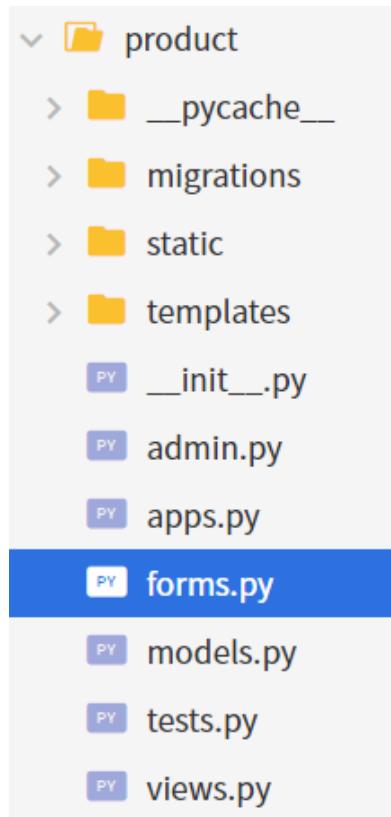
Productname_orange_1	content_orange_1
Productname_orange_2	Contents_orange_2
Productname_orange_3	Content_orange_3

```
<!-- Section -->
<section>
  <div class="container">
    <div class="row">
      <div class="col-lg-12 text-center">
        <table class="table table-hover table-bordered">
          <tbody>
            <tr>
              <td>Productname_orange_1</td>
              <td>content_orange_1</td>
            </tr>
            <tr>
              <td>Productname_orange_2</td>
              <td>Contents_orange_2</td>
            </tr>
            <tr>
              <td>Productname_orange_3</td>
              <td>Content_orange_3</td>
            </tr>
          </tbody>
        </table>
      </div>
    </div>
  </div>
</section>
```

우리가 원래 작성했던 template tag가 보이지 않습니다. 이렇게 사용자에게 보여질 때에는 html 형식으로 넘겨주게 됩니다.

Product 프로젝트에서 모델을 작성하는게 핵심

(이모든 것은 이 부분은 중요이 필요하지만 끝만 따라하기면)



App 폴더에 form.py 파일을 만들고 그 안에 아래와 같이
작성합니다.

```
1 from django import forms
2 from .models import Product
3
4 class PostForm(forms.ModelForm):
5     class Meta:
6         model = Product
7         fields = ('productname', 'contents',)
```

Product 페이지에서 게시물을 작성하거나 하기

(이전과는 다른 부분은 응용이 필요하지 않은 부분만 다루겠습니다.)

```
1 from django.shortcuts import render
2 from .models import Product
3 from .forms import PostForm
4
5 # Create your views here.
6 def index(request):
7     return render(request, 'product/index.html')
8
9 def product(request):
10    productlist = Product.objects.all()
11    if request.method == "POST":
12        form = PostForm(request.POST)
13        if form.is_valid():
14            post = form.save(commit=False)
15            post.author = request.user
16            post.save()
17            return render(request, 'product/product.html', {'productlist':productlist})
18    else:
19        form = PostForm()
20    return render(request, 'product/product.html', {'form':form, 'productlist':productlist})
```

form으로 전달된 값이 POST인지 확인하여 처리합니다.

Product 페이지에서 게시물을 작성하거나 하기

(이전과는 다른 부분은 응용이 필요하지만 끝만 바꿀 수 있습니다.)

```
<form method="POST">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">submit</button>
</form>
```

form.as_p 템플릿 태그를 사용하면 Django가 폼에 있는 form을 템플릿으로 전달해줍니다.

Product 페이지에서 제시물을 작성하거나 하기

(이전과는 - 이 부분은 유통이 필요하지만 브랜드 전략화하기.)

Productname:

Contents:

form.as_p

• Productname:

• Contents:

form.as_ul

Productname: Contents:

form.as_table

Step 5

DBI Template

구름 IDE 가상환경 설정

이 명령어는 구름 IDE 컨테이너를 다시 실행할 때마다 쳐주어야 합니다.

```
root@goorm:/workspace/여러분ID# python --version
```

```
root@goorm:/workspace/여러분ID# cd mysite
```

```
root@goorm:/workspace/여러분ID/mysite# source myvenv/bin/activate
```

```
(myvenv)root@goorm:/workspace/여러분ID/mysite#
```

myvenv가 붙지 않은 상태에서 그동안 명령어를 치렀다면 지금이라도 컨테이너를 삭제해버리시고 처음부터 다시 하시길 권장해 드립니다.

productdetails

```
16  from django.conf.urls import url
17  from django.contrib import admin
18  from product.views import index, product, productdetails
19
20 urlpatterns = [
21     url(r'^admin/', admin.site.urls),
22     url(r'^$', index),
23     url(r'^product', product),
24     url(r'^^(?P<pk>\d+)/$', productdetails),
25 ]
```

현재 '홈페이지 구성'에 3번을 작업하는 과정 중에 있습니다.

3. <http://사용자이름.run.goorm.io/제품번호>

- product 페이지에서 제품을 눌렀을 때 이동
- Template : productdetails.html

뒤에 보이시는 r'내용' 부분에 내용은 정규표현식이 들어가요. 이 부분은 마지막 괄호에서 다루겠지만 간단하게 먼저 설명을 해드리고자 합니다.

url(r'^^(?P<pk>\d+)/\$')

파턴의 시작

한자리 숫자가 반복

이전 또는 한번 반복

파턴의 끝

변수명 pk는 primary key의 약자, 변수명은 마음 대로 변경 가능. 뷰에서 아규먼트(쉽게 말해 인자값)를 받아 해당 값을 사용자에게 전달

Post_detail

```
1 from django.shortcuts import render
2 from .models import Product
3
4 # Create your views here.
5 def index(request):
6     return render(request, 'product/index.html')
7
8 def product(request):
9     productlist = Product.objects.all()
10    return render(request, 'product/product.html', {'productlist':productlist,})
11
12 def productdetails(request, pk):
13     productlist = Product.objects.get(pk=pk)
14     return render(request, 'product/productdetails.html', {'productlist':productlist,})
```

Post_detail

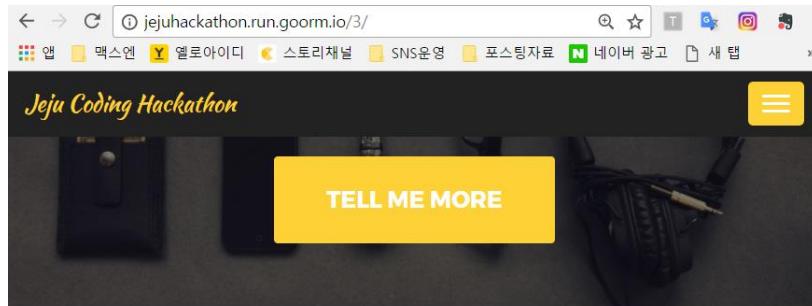
jejuhackathon/mysite/product/templates/product/productdetails.html

```
<!-- Section -->
<section>
    <div class="container">
        <div class="row">
            <div class="col-lg-12 text-center">
                <h2>{{ productlist.productname }}</h2>
                <p>{{ productlist.contents }}</p>
            </div>
        </div>
    </div>
</section>
```

Step_4_3.html

Post_detail

<http://jejuhackathon.run.goorm.io/3/>



PRODUCTNAME_ORANGE_3

Content_orange_3

Post

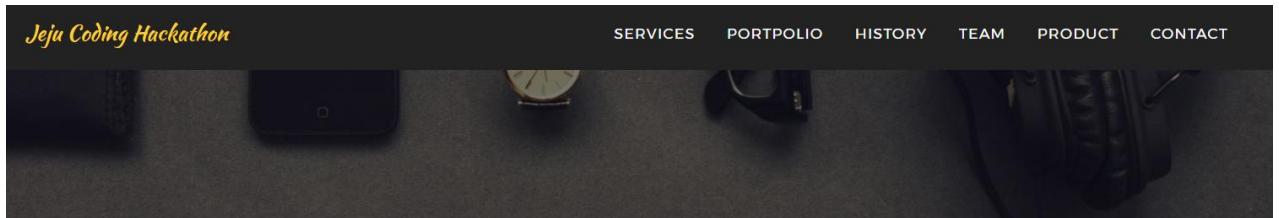
jejuhackathon/mysite/product/templates/product/product.html

```
{% for list in productlist %}  
|
    <td>{{ list.productname }}</td>  
    <td>{{ list.contents }}</td>  
</tr>  
{% endfor %}
```

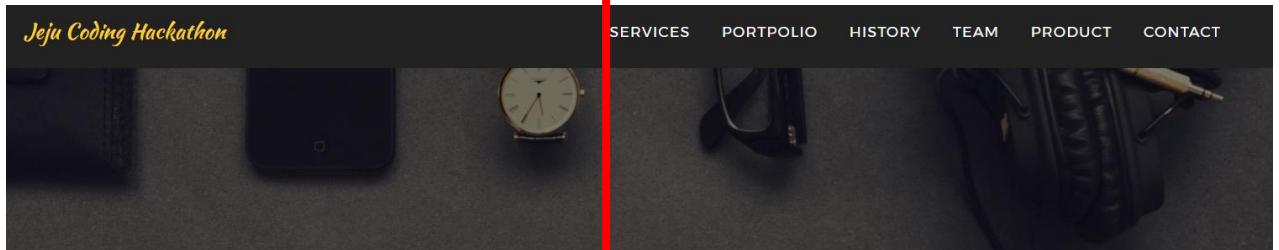
Step_4_4.html

'<http://jejuhackathon.run.goorm.in/>' 이 부분은 여러분의 URL로 교체해 주세요.

Post 77



Productname_orange_1	content_orange_1
Productname_orange_2	Contents_orange_2
Productname_orange_3	Content_orange_3



PRODUCTNAME_ORANGE_1

content_orange_1

jejuhackathon/mysite/product/templates/product/productdetails.html

```
95 <!-- Section -->
96 <section>
97 <div class="container">
98 <div class="row">
99 <div class="col-lg-12 text-center">
100 <a href="https://jejuhackathon.run.goorm.io/product" class="btn btn-default">목록</a>
101 <h2>{{ productlist.productname }}</h2>
102 <p>{{ productlist.contents }}</p>
103 </div>
104 </div>
105 </div>
106 </section>
```

Step_4_5.html

Post photo field ↗↑↑

jejuhackathon/mysite/product/models.py

```
1 from django.db import models
2
3 # Create your models here.
4 class Product(models.Model):
5     productname = models.CharField(max_length=50)
6     photo = models.ImageField(blank=True, null = True)
7     contents = models.TextField()
8
9     def __str__(self):
10        return self.productname
```

여기서 null True를 주게 되면 기존에 있는 게시판 게시물은 null 값으로 들어가게 됩니다.

Post photo DB 생성

```
(myvenv)root@goorm:/workspace/毗邻ID/mysite# pip install pillow==2.9.0
```

```
root@goorm:/workspace/jejuhackathon/mysite# python manage.py makemigrations
Migrations for 'product':
  0002_product_photo.py:
    - Add field photo to product
root@goorm:/workspace/jejuhackathon/mysite# python manage.py migrate
Operations to perform:
  Synchronize unmigrated apps: staticfiles, messages
  Apply all migrations: admin, contenttypes, product, auth, sessions
Synchronizing apps without migrations:
  Creating tables...
    Running deferred SQL...
  Installing custom SQL...
Running migrations:
  Rendering model states... DONE
  Applying product.0002_product_photo... OK
root@goorm:/workspace/jejuhackathon/mysite# █
```

Post

* change post

Django administration

Welcome, **jejuhackathon**. View site / Change password / Log out

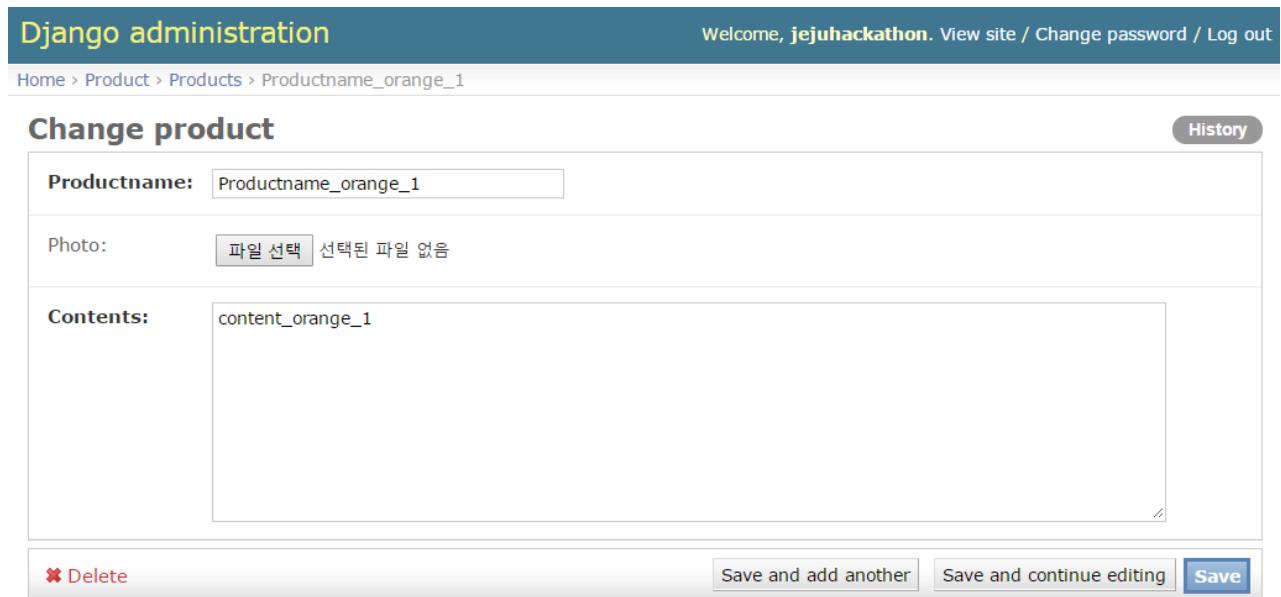
Home > Product > Products > Productname_orange_1

Change product

History

Productname:	Productname_orange_1
Photo:	<input type="button" value="파일 선택"/> 선택된 파일 없음
Contents:	content_orange_1

Save and add another Save and continue editing



* Add post

Add post

Title:	<input type="text"/>
Photo:	<input type="button" value="파일 선택"/> 선택된 파일 없음
Content:	<input type="text"/>

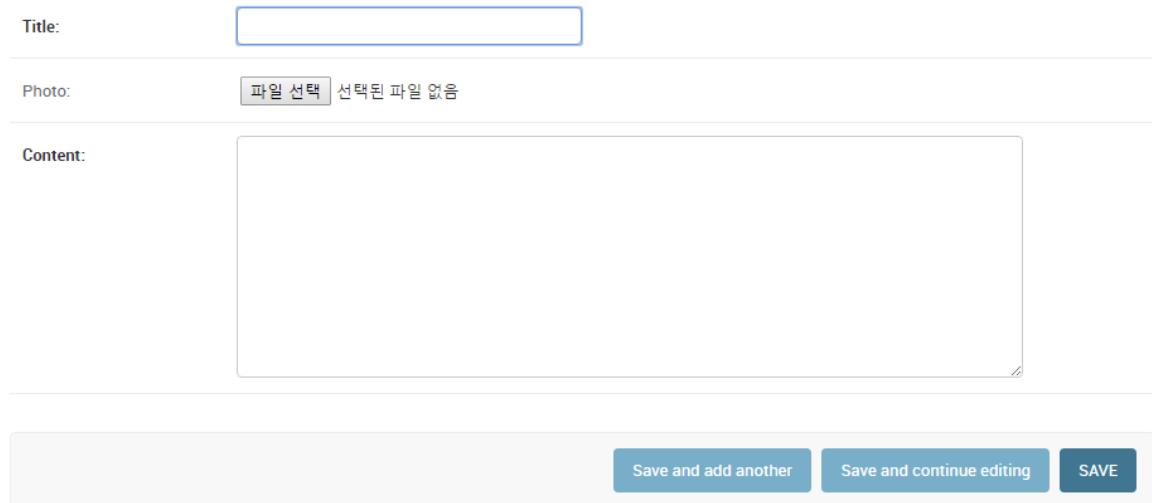
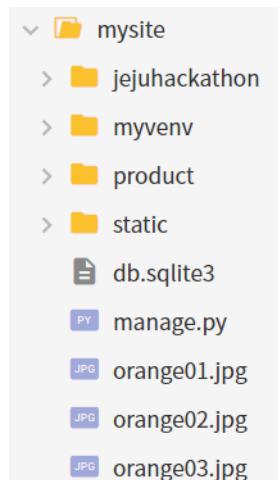


Photo 저장 공간 확보

jejuhackathon/mysite/jejuhackathon/settings.py

```
126     MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
127     MEDIA_URL = '/media/'
```

* 설정을 안해주시면 이곳으로 들어가옵니다.



Django administration

Home > Product > Products > Productname_orange_1

Change product

Productname:	Productname_orange_1
Photo:	Currently: ./orange01.jpg <input type="button" value="Clear"/> Change: <input type="button" value="파일 선택"/> 선택된 파일 없음
Contents:	content_orange_1

Post

Page not found (404)

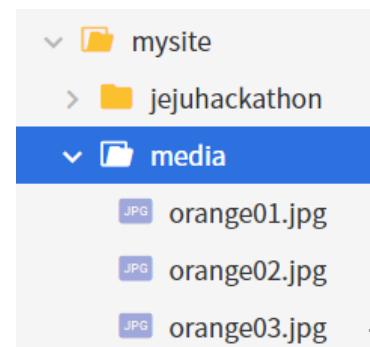
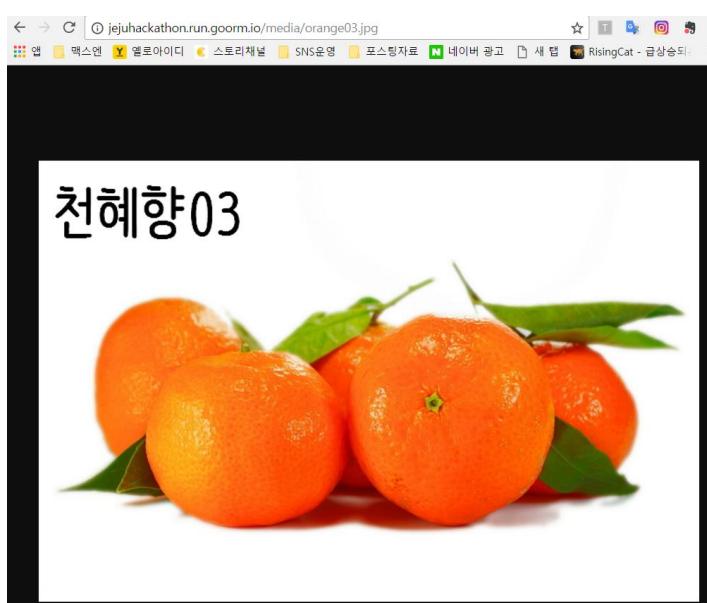
Request Method: GET
Request URL: http://jejuhackathon.run.goorm.io/admin/product/product/1/orange01.jpg/
Raised by: django.contrib.admin.options.change_view

product object with primary key u'1/orange01.jpg' does not exist.

You're seeing this error because you have DEBUG = True in your Django settings file. Change that to False, and Django will display a standard 404 page.

```
16 | from django.conf.urls import url
17 | from django.contrib import admin
18 | from product.views import index, product, productdetails
19 | from django.conf.urls.static import static
20 | from django.conf import settings
21 |
22 | urlpatterns = [
23 |     url(r'^admin/', admin.site.urls),
24 |     url(r'^$', index),
25 |     url(r'^product$', product),
26 |     url(r'^^(?P<pk>\d+)/$', productdetails),
27 | ]
28 |
29 | urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Post image



Post image

jejuhackathon/mysite/product/templates/product/productdetails.html

```
1  <!-- Content -->
2  <div class="container">
3  <div class="row">
4      <h1> {{ productlist.productname }} </h1>
5      <div class="col-md-8">
6          {% if productlist.photo %}
7              
8          {% endif %}
9      </div>
10     </div>
11
12     <div class="row">
13         <div class="col-md-8">
14             <p> {{ productlist.contents|linebreaks }} </p>
15             <a href="https://jejuhackathon.run.goorm.io/product" class="btn btn-default">목록</a>
16         </div>
17     </div>
18 </div>
19 <!-- Content end-->
```

Step_4_6.html

제작한 사이트 네이버 뉴스는 여기를 홈페이지 주소로 바꿔주세요.

실행화면

The screenshot shows a website header with "Jeju Coding Hackathon" and navigation links for SERVICES, PORTFOLIO, HISTORY, TEAM, PRODUCT, and CONTACT. Below the header, the product name "PRODUCTNAME_ORANGE_1" is displayed in bold. The main content area features the text "천혜향01" above a large image of several ripe oranges with green leaves. At the bottom left, there is a link labeled "content_orange_1" and a button labeled "목록".

Step 7

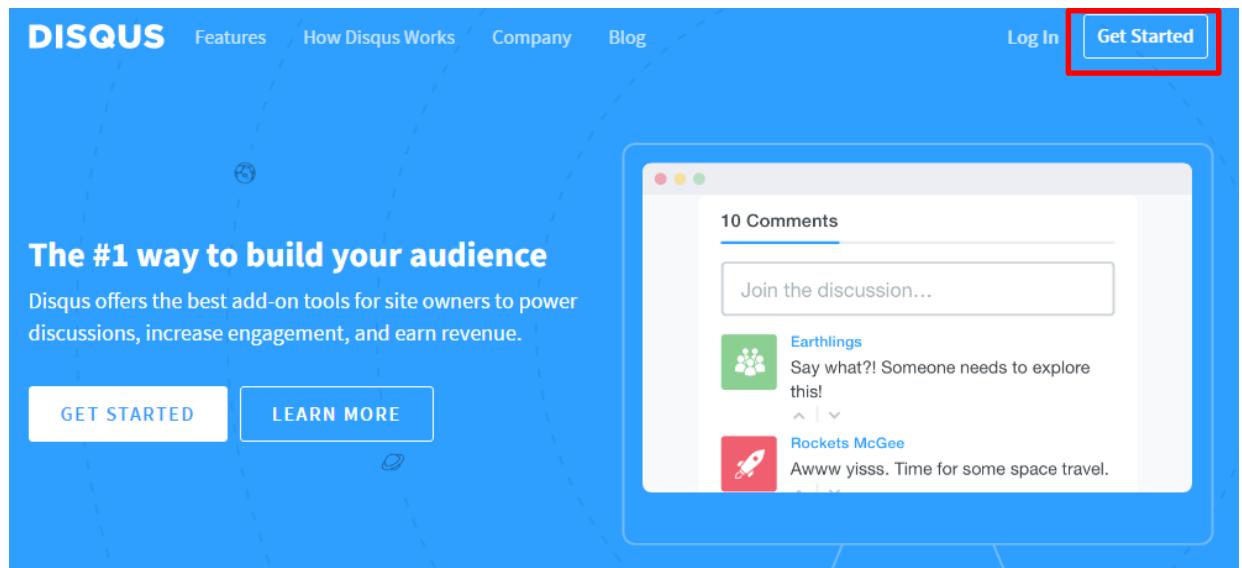
가상환경에 Django-disqus 설치하기(댓글 달기 기능 구현)

댓글 달 수 있는 패키지인 disqus 설치하기

```
(myvenv)root@goorm:/workspace/여러분ID/mysite# pip install django-disqus
```

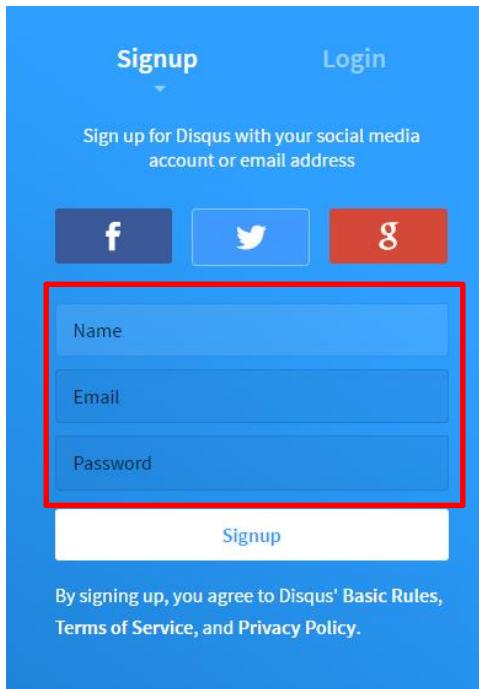
Post

댓글 및 좋아요를 누를 수 있는 Framework인 DISQUS 연결!

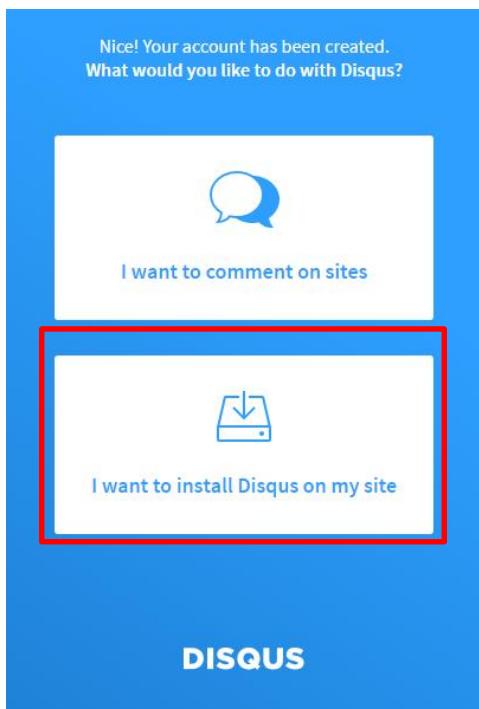


<https://disqus.com/>

Post



Name, E-mail 등은 관리하기 쉽도록
구글 IDE ID, PW 사용



Create a new site

All fields are required.

Site Owner  jejuhackathon
To associate a different account as the site owner,
[login with a different account](#)

Website Name

Shortname
Your unique disqus URL will be: jejuhackathon-django.disqus.com

Category

Language

Create Site

Select a plan

Select your Disqus plan for this organization

Your organization will be on the ads-supported Disqus Basic plan:

Basic
Free, Ads Supported

Get all of the core Disqus features including: Engage comments plug-in, advanced spam filtering, moderation tools, basic analytics, configurable ads, and more. [Learn more about Disqus' features.](#)

Continue on Basic

Post

면 아래 있는 것 클릭



I don't see my platform listed, install
manually with Universal Code

설치한 내용은 Settings.py 파일에 등록

```
33 INSTALLED_APPS = [  
34     'django.contrib.admin',  
35     'django.contrib.auth',  
36     'django.contrib.contenttypes',  
37     'django.contrib.sessions',  
38     'django.contrib.messages',  
39     'django.contrib.staticfiles',  
40     'product',  
41     'disqus',  
42     'django.contrib.sites',  
43 ]
```

```
73 DISQUS_WEBSITE_SHORTNAME = 'jejuhackathon-django'  
74 SITE_ID = 1
```

Post

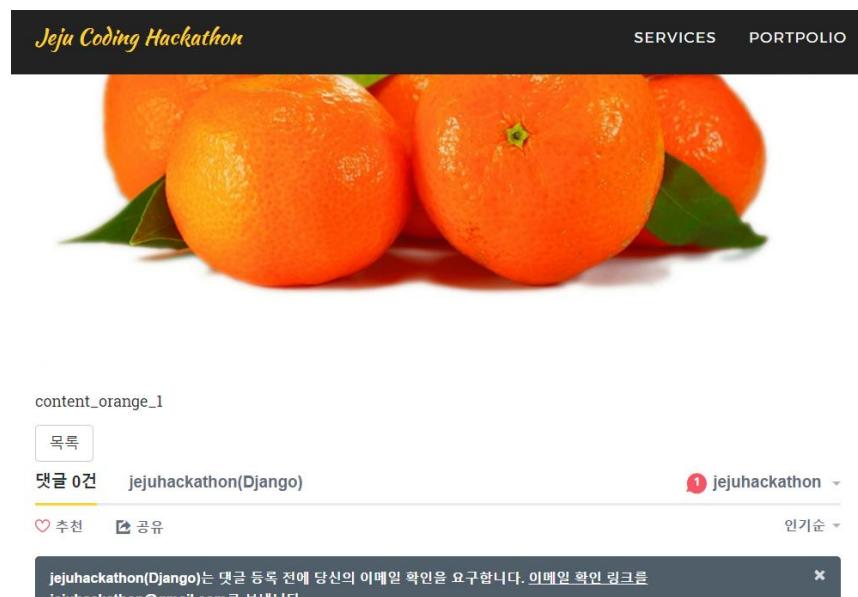
DB에 박스팅

```
^Croot@goorm:/workspace/jejuhackathon/mysite# python manage.py migrate
Operations to perform:
  Synchronize unmigrated apps: staticfiles, disqus, messages
  Apply all migrations: product, sessions, admin, sites, auth, contenttypes
Synchronizing apps without migrations:
  Creating tables...
    Running deferred SQL...
  Installing custom SQL...
Running migrations:
  Rendering model states... DONE
  Applying sites.0001_initial... OK
root@goorm:/workspace/jejuhackathon/mysite# python runserver 0.0.0.0:80
```

테플릿에 박스팅

djangotest/mysite/product/templates/product/productdetails.html

```
<div class="row">
  <div class="col-md-8">
    <p> {{ productlist.contents|linebreaks }} </p>
    <a href="https://jejuhackathon.run.goorm.io/product" class="btn btn-default">목록</a>
    <div>
      {% load disqus_tags %}
      {% disqus_show_comments %}
    </div>
  </div>
</div>
```



토론 시작

Post

* 문제 해결 시

djangotest/mysite/djangotest/settings.py

45 DISQUS_API_KEY = '3T9JjvNUqEmBhmBIa4EoK7WNh0Nh71KF0xAZGhrkwMz5cQiNHmF1xs5CxHhgwaNW'

<https://disqus.com/api/applications/>에 링크되어 있어요

```
1 from django.db import models
2
3 # Create your models here.
4 class Product(models.Model):
5     productname = models.CharField(max_length=50)
6     photo = models.ImageField(blank=True, null = True)
7     published_date = models.DateTimeField(blank=True, null=True)
8     modified_date = models.DateTimeField(blank=True, null=True)
9     contents = models.TextField()
10
11     def __str__(self):
12         return self.productname
```

```
root@goorm:/workspace/jejuhackathon/mysite# python manage.py makemigrations
Migrations for 'product':
  0003_auto_20170502_0300.py:
    - Add field modified_date to product
    - Add field published_date to product
root@goorm:/workspace/jejuhackathon/mysite# python manage.py migrate
Operations to perform:
  Synchronize unmigrated apps: staticfiles, disqus, messages
  Apply all migrations: product, sessions, admin, sites, auth, contenttypes
Synchronizing apps without migrations:
  Creating tables...
    Running deferred SQL...
    Installing custom SQL...
Running migrations:
  Rendering model states... DONE
  Applying product.0003_auto_20170502_0300... OK
root@goorm:/workspace/jejuhackathon/mysite#
```

Django administration

Welcome, **jejuhackathon**. View site / Change password / Log out

Home > Product > Products > Productname_orange_1

Change product

History

Productname:

Photo: Currently: ./orange01.jpg
Change: 파일 선택 선택된 파일 없음

Published date: Date: Today |
Time: Now |

Modified date: Date: Today |
Time: Now |

Contents:

Django administration

Welcome, **jejuhackathon**. View site / Change password / Log out

Home > Product > Products > Productname_orange_3

Change product

History

Productname:

Photo: Currently: ./orange03.jpg
Change: 파일 선택 선택된 파일 없음

Published date: Date: 2017-05-02 Today |
Time: 03:05:25 Now |

Modified date: Date: 2017-05-02 Today |
Time: 03:05:27 Now |

Contents:

3개 모두 수정 완료

jejuhackathon/mysite/product/templates/product/product.html

```
<table class="table table-hover table-bordered">
  <tbody>
    <tr>
      <th>Name</th>
      <th>Contents</th>
      <th>Date</th>
    <tr>
      {% for list in productlist %}
      <tr onclick="location.href='http://jejuhackathon.run.goorm.io/{{ list.pk }}'">
        <td>{{ list.productname }}</td>
        <td>{{ list.contents }}</td>
        <td>{{ list.published_date }}</td>
      </tr>
      {% endfor %}
    </tbody>
</table>
```

Name	Contents	Date
Productname_orange_1	content_orange_1	May 2, 2017, 3:05 a.m.
Productname_orange_2	Contents_orange_2	May 2, 2017, 3:05 a.m.
Productname_orange_3	Content_orange_3	May 2, 2017, 3:05 a.m.

<https://jejuhackathon.run.goorm.io/product>

```
root@goorm:/workspace/jejuhackathon/mysite# pip install django-taggit
Downloading/unpacking django-taggit
  Downloading django_taggit-0.22.1-py2.py3-none-any.whl (45kB): 45kB downloaded
Installing collected packages: django-taggit
Successfully installed django-taggit
Cleaning up...
```

```
1  from django.db import models
2  from taggit.managers import TaggableManager
3
4  # Create your models here.
5  class Product(models.Model):
6      productname = models.CharField(max_length=50)
7      photo = models.ImageField(blank=True, null = True)
8      published_date = models.DateTimeField(blank=True, null=True)
9      modified_date = models.DateTimeField(blank=True, null=True)
10     contents = models.TextField()
11     tag = TaggableManager(blank=True)
12
13     def __str__(self):
14         return self.productname
```

```

33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'product',
41     'disqus',
42     'django.contrib.sites',
43     'taggit',
44 ]
45
46 TAGGIT_CASE_INSENSITIVE = True

```

```

root@goorm:/workspace/jejuhackathon/mysite# python manage.py makemigrations
Migrations for 'product':
  0004_product_tag.py:
    - Add field tag to product
root@goorm:/workspace/jejuhackathon/mysite# python manage.py migrate
Operations to perform:
  Synchronize unmigrated apps: staticfiles, disqus, messages
  Apply all migrations: product, sessions, admin, auth, contenttypes, taggit
Synchronizing apps without migrations:
  Creating tables...
    Running deferred SQL...
  Installing custom SQL...
Running migrations:
  Rendering model states... DONE
  Applying product.0004_product_tag... OK
root@goorm:/workspace/jejuhackathon/mysite#

```

Django administration

Welcome, **jejuhackathon**. View site / Change password / Log out

Home > Product > Products > Productname_orange_1

Change product

Productname:

Photo: Currently: ./orange01.jpg Change: 선택된 파일 없음

Published date: Date: Today | Time: Now |

Modified date: Date: Today | Time: Now |

Contents:

Tags:
A comma-separated list of tags.

Home > Product > Products > Productname_orange_3

Change product

[History](#)Productname: Photo: Currently: ./orange03.jpg Change: 선택된 파일 없음Published date: Date: Today | Time: Now | Modified date: Date: Today | Time: Now | Contents: Tags:

A comma-separated list of tags.

Tags all orange

jejuhackathon/mysite/product/templates/product/productdetails.html

```

<!-- Content -->


<div class="row">
        <h1> {{ productlist.productname }} </h1>
        <div class="col-md-8">
            {% if productlist.photo %}
                
            {% endif %}
            {{ productlist.tag.names }}
        </div>
    </div>


```

PRODUCTNAME_ORANGE_1

천혜향01



[u'orange']

content_orange_1

목록

Welcome to Disqus! Discover more great discussions just like this one. We're a lot more than comments.

시작하세요

Dismiss X

```
<!-- Content -->
<div class="container">
    <div class="row">
        <h1> {{ productlist.productname }} </h1>
        <div class="col-md-8">
            {% if productlist.photo %}
                
            {% endif %}
            {{ productlist.tag.names.0 }}
        </div>
    </div>
```

```
16 from django.conf.urls import url
17 from django.contrib import admin
18 from product.views import index, product, productdetails, orange, hanlabong
19 from django.conf.urls.static import static
20 from django.conf import settings
21
22 urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^$', index),
    url(r'^product', product),
    url(r'^(?P<pk>\d+)/$', productdetails),
    url(r'^orange/$', orange),
    url(r'^hanlabong/$', hanlabong),
]
```

```
1 from django.shortcuts import render
2 from .models import Product
3 from taggit.models import Tag
```

```
12 def productdetails(request, pk):
13     productlist = Product.objects.get(pk=pk)
14     return render(request, 'product/productdetails.html', {'productlist':productlist,})
15
16 def orange(request):
17     Tag_list = Tag.objects.all()
18     productlist = Product.objects.all()
19     return render(request, 'product/orange.html',{'Tag_list':Tag_list, 'productlist':productlist,})
20
21 def hanlabong(request):
22     Tag_list = Tag.objects.all()
23     productlist = Product.objects.all()
24     return render(request, 'product/hanlabong.html',{'Tag_list':Tag_list, 'productlist':productlist,})
```

jejuhackathon/mysite/product/templates/product/orange.html

```
<!-- Content -->
<h1>tag를 게시할 곳입니다.</h1>
<div class="container">
    <div class="row">
        <div class="col-sm-12">
            <table class="table table-hover table-bordered">
                {% for taglist in Tag_list %}
                    <p>{{ taglist }}</p>
                {% endfor %}

                {% for list in productlist %}
                    {% if list.tag.names.0 == "orange" %}
                        <h1>{{ list.productname }}</h1>
                    {% if list.photo %}
                        
                    {% endif %}
                    {% endif %}
                    {% endfor %}
                    <a href="https://jejuhackathon.run.goorm.io/product" class="btn btn-default">목록</a>
            </div>
        </div>
    </div>
<!-- Content end-->
```

Step_4_7.html

jejuhackathon/mysite/product/templates/product/hanlabong.html

```
95      <!-- Content -->
96      <h1>tag을 게시할 곳입니다.</h1>
97      <div class="container">
98          <div class="row">
99              <div class="col-sm-12">
100                  <table class="table table-hover table-bordered">
101                      {% for taglist in Tag_list %}
102                          <p>{{ taglist }}</p>
103                      {% endfor %}

104                      {% for list in productlist %}
105                      {% if list.tag.names.0 == "hanlabong" %}
106                          <h1>{{ list.productname }}</h1>
107                      {% if list.photo %}
108                          
110                      {% endif %}
111                      {% endif %}
112                      {% endfor %}
113                  <a href="https://jejuhackathon.run.goorm.io/product" class="btn
114                      btn-default">목록</a>
115                  </div>
116              </div>
117      <!-- Content end-->
```

Step_4_8.html

Django administration

Welcome, **jejuhackathon**. [View site](#) / [Change password](#) / [Log out](#)

Home > Product > Products

✓ The product "Productname_hanlabong_3" was added successfully.

Select product to change

Action: 0 of 6 selected

<input type="checkbox"/>	Product
<input type="checkbox"/>	Productname_hanlabong_3
<input type="checkbox"/>	Productname_hanlabong_2
<input type="checkbox"/>	Productname_hanlabong_1
<input type="checkbox"/>	Productname_orange_3
<input type="checkbox"/>	Productname_orange_2
<input type="checkbox"/>	Productname_orange_1

6 products

<http://jejuhackathon.run.goorm.io/orange/>

Jeju Coding Hackathon SERVICES PORTPOLIO

orange
hanlabong

PRODUCTNAME_ORANGE_1

천혜향01



PRODUCTNAME_ORANGE_2

천혜향02



<http://jejuhackathon.run.goorm.io/hanlabong/>

Jeju Coding Hackathon SERVICES PORTPOLIO

orange
hanlabong

PRODUCTNAME_HANLABONG_1

한라봉01



PRODUCTNAME_HANLABONG_2

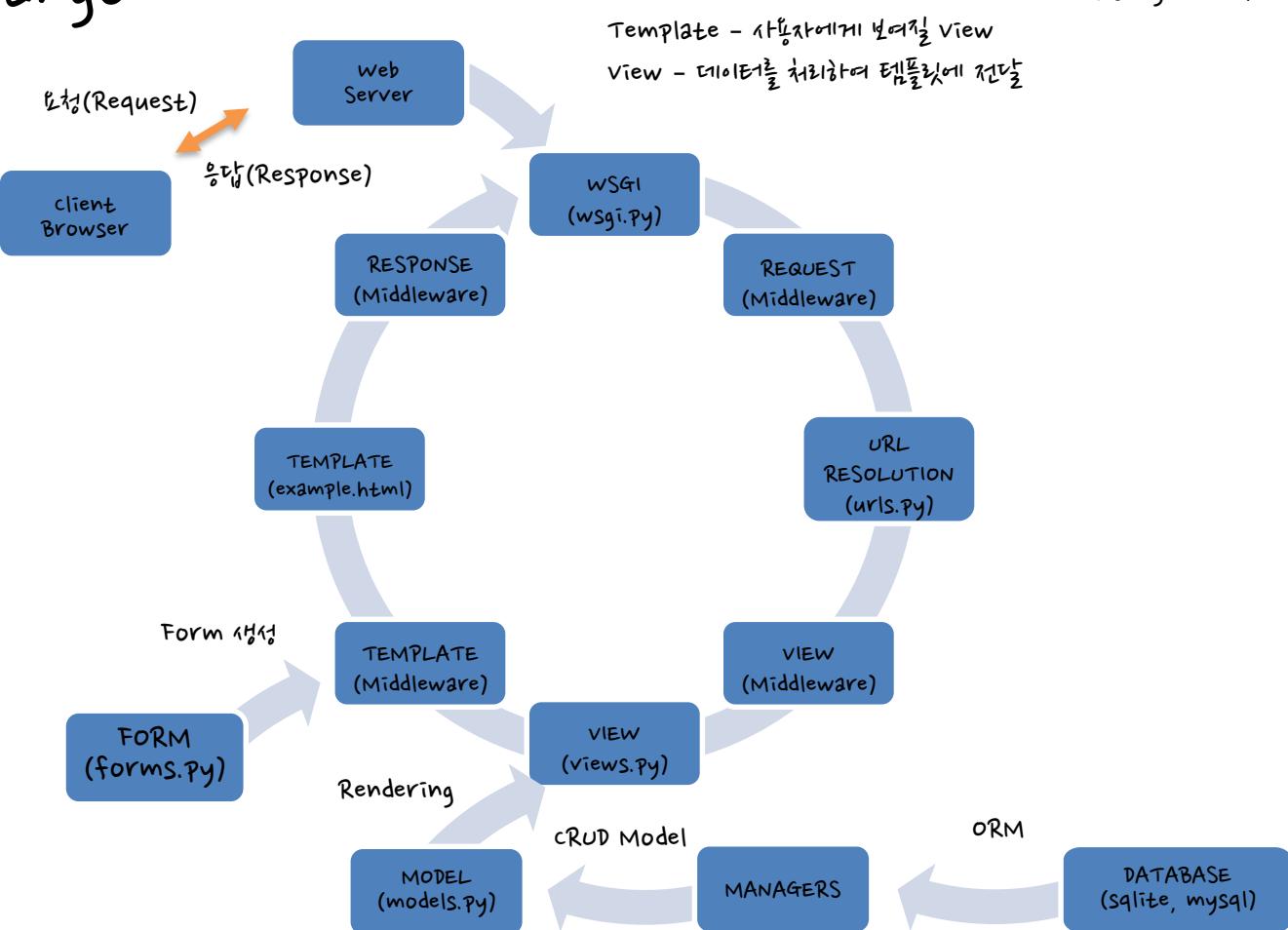
한라봉02



Django

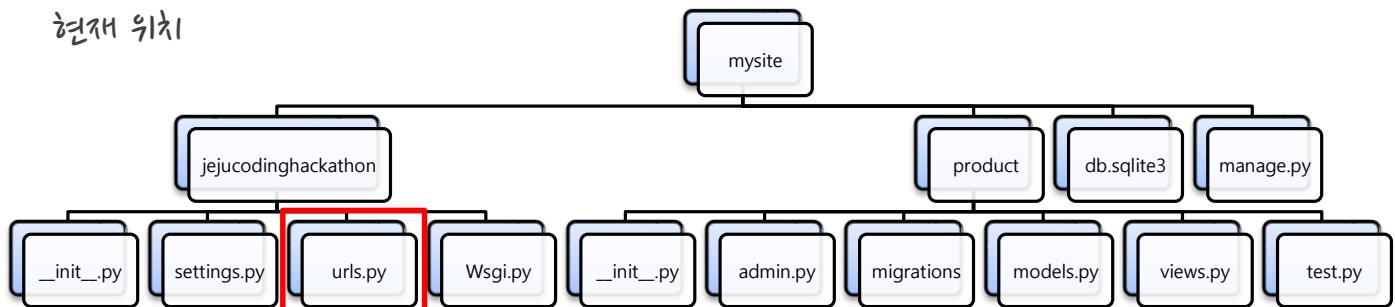
- High-level Python Web Framework
- Develop fast
- Automate the repetitive stuff
- Follow best practices

Django



URL 설정

현재 위치



URLconf - URL 설정

URL 패턴 설정

(별도자료)

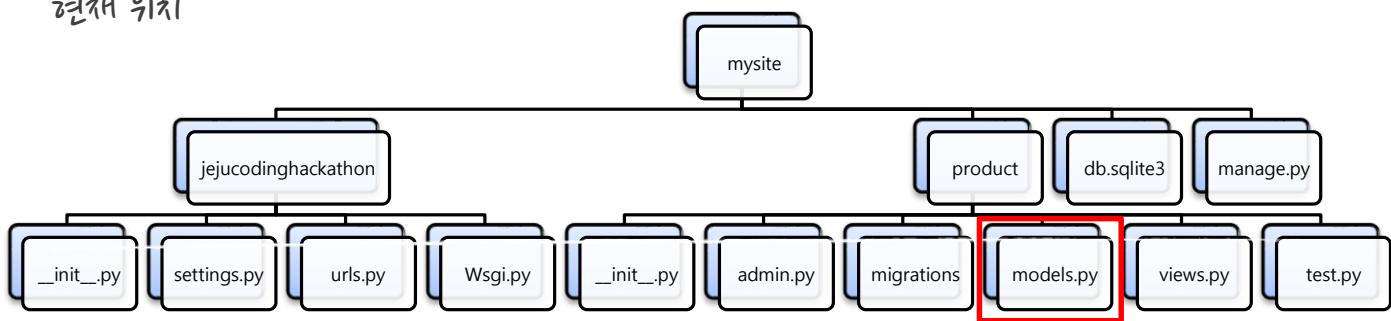
```
16 from django.conf.urls import url
17 from django.contrib import admin
18 from product.views import index, product, productdetails, orange, hanlabong
19 from django.conf.urls.static import static
20 from django.conf import settings
21
22 urlpatterns = [
23     url(r'^admin/', admin.site.urls),
24     url(r'^$', index),
25     url(r'^product$', product),
26     url(r'^^(?P<pk>\d+)/$', productdetails),
27     url(r'^orange/$', orange),
28     url(r'^hanlabong/$', hanlabong),
29 ]
30 urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

URL 설정

Symbol	내용
.	(Dot) 모든 단일 문자
^	(Caret) 문자열의 시작
\$	문자열의 끝
[]	[] 괄호에 있는 문자 하나, 예를 들어 [akz]라면 a또는 k또는 z
[^]	[] 괄호에 있는 문자 이외의 문자 하나, 예를 들어 [^ab]라면 a와 b를 제외한 문자 하나
*	0번 이상 반복
+	1번 이상 반복
?	0번 또는 1번 반복
{ n }	n번 반복
{ m, n }	최소 m에서 n번 반복
wd	모든 단일 숫자
[A-Z]	모든 대문자(A~Z)
[a-z]	모든 소문자(a~z)
[A-Z a-z]	모든 대소문자 구별없는 문자(a~z 혹은 A~Z)
[^]+	첫 /가 나타날때까지의 하나이상의 문자

Model - 데이터 베이스 설정

현재 위치

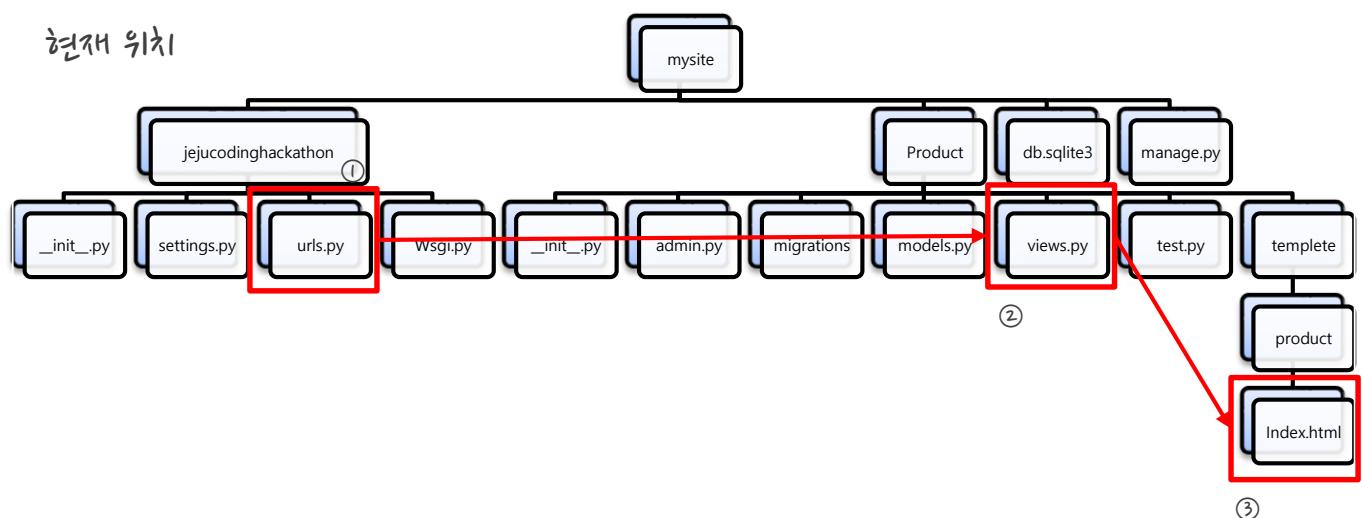


Model - 데이터베이스 모델 설계

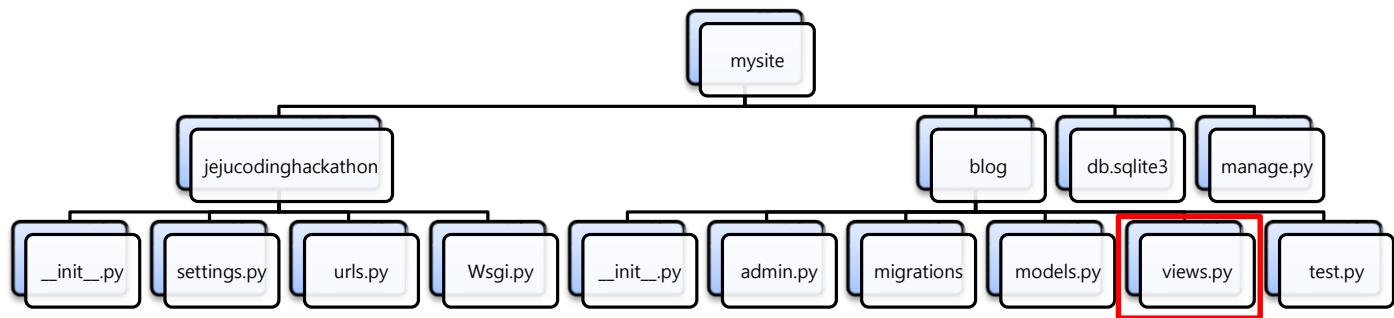
```
1  from django.db import models
2  from taggit.managers import TaggableManager
3
4  # Create your models here.
5  class Product(models.Model):
6      productname = models.CharField(max_length=50)
7      photo = models.ImageField(blank=True, null=True)
8      published_date = models.DateTimeField(blank=True, null=True)
9      modified_date = models.DateTimeField(blank=True, null=True)
10     contents = models.TextField()
11     tag = TaggableManager(blank=True)
12
13     def __str__(self):
14         return self.productname
```

Template - 화면 UI 설계

현재 위치



View - Յ Հ Ա Ր Ա Խ



(코딩 기초에서 서비스 런칭까지 4박 5일) 제주 코딩 베이스 캠프

초판 1쇄 발행 | 2017년 8월 25일

지은이 | 이호준, 이범재, 유준모

편집 | 이호준

총괄 | 이호준

펴낸곳 | 사도출판

주소 | 제주특별자치도 제주시 사라봉 9길 18, 102호

표지디자인 | 사도출판

홈페이지 | <http://www.paullab.co.kr>

E-mail | paul-lab@naver.com

ISBN | 979-11-959952-1-9

Copy right © 2017 by. 사도출판

이 책의 저작권은 사도출판에 있습니다.

저작권법에 의해 보호를 받는 저작물이므로 무단 복제 및 무단 전재를 금합니다.

이 책에 대한 의견을 주시거나 오탈자 및 잘못된 내용의 수정 정보는 사도출판의
이메일로 연락을 주시기 바랍니다.

코딩 기초에서 서비스 런칭까지 4박5일 제주 코딩 베이스캠프

코딩 기초부터 기획, 서비스 배포까지 4박 5일. 88,000원. 우리는 단지 열정 있는 청년들에게 열정만큼 성장할 수 있는 기회를. 서비스 구축 성공경험을 통해 이 취업난 시대에 자신감, 자존감, 성취감, 자립할 수 있는 능력을 배양할 수 있도록. 또, 이러한 경험이 단순히 금액이 높다는 이유로 또 한번에 좌절이 되지 않도록. 이 행사를 진행합니다.

1년 2회, 제주에서 진행되며 facebook 그룹인 제주 코딩 베이스캠프를 통하여 모집합니다.

