2021.06.11 HTML/JS/CSS

⟨application.yaml vs application.properties⟩

Spring에서 configuration data을 properties file에 둔다. Spring Boot에서는 properties file 대신 YAML파일을 사용할 수 있다.

- application.properties 예시

```
spring.datasource.url=jdbc:h2:dev
spring.datasource.username=SA
spring.datasource.password=password
```

- application.yaml 예시

```
spring:
    datasource:
        password: password
        url: jdbc:h2:dev
        username: SA
```

그럼 Spring Boot는
무엇일까?????????????
스프링의 생태계부터 알아보면.

- application.properties vs application.yml for Spring Boot

1.) .properties 파일 : 데이터를 순차적으로 저장.
.yml 파일 : 계층 적 형식으로 데이터를 저장.

2.) .properties 파일 : 기본적으로 문자열 값의 키-값 쌍만 지원.
.yml 파일 : 맵, 목록, 스칼라 유형 값뿐만 아니라 키-값 쌍을 지원.

3.) .properties 파일 : JAVA 전용으로 사용되는 파일.
.yml 파일 : 이 파일 유형은 JAVA, Python, ROR 등과 같은 많은 언어에서 사용.

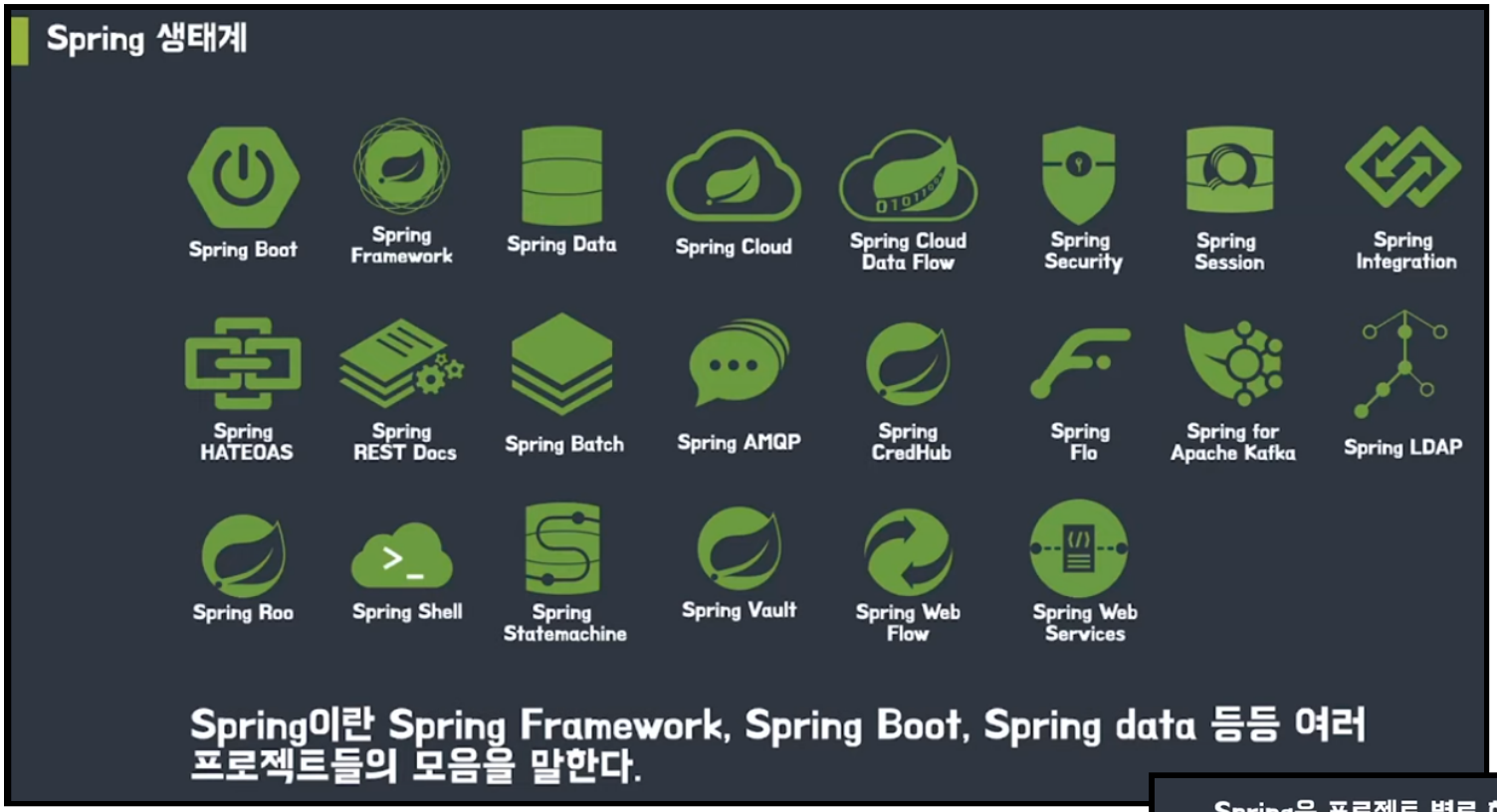
4.) 여러 프로필을 처리하고자하는 경우
.properties 파일 : 이 경우 각 프로필에 대해 개별 파일을 관리해야함.
.yml 파일 : 이 파일 유형에서는 단일 파일을 관리하고 그 안에 특정 프로필의 구성 데이터를 배치하면됨.

5.) Spring 프로젝트의 경우
.properties 파일 : @PropertySource 지원함.
.yml 파일 : @PropertySource 지원할 수 없다.

출처: https://github.com/heyKim/TIL/blob/main/spring/application.yaml_application.properties.md

Conclusion

- yaml 파일이 좀더 가독성 높다.
- Spring Boot 2.4.0 이전에는 properties 파일에는 제약들이 있었는데 이제는 다 해결됨
- 다만 yaml은 python, Ruby, Kubernetes 등에서 많이 쓰니 Java에서도 yaml쓰면 호환도 되고.. 좋을 듯 하다.



스프링 프로젝트들은 스프링 프레임워크를 기반으로 돌아감.

))스프링 프레임워크는 스프링의 핵심

스프링 프레임워크는 객체 지향의 특징을 잘 활용할 수 있게 해주며 개발자들은 핵심 비지니스 로직 구현에만 집중할 수 있게 해주는 프레임워크

"The Spring Framework provides a comprehensive programming and configuration model for modern Java-based enterprise applications - on any kind of deployment platform."

"A key element of Spring is infrastructural support at the application level: Spring focuses on the "plumbing" of enterprise applications so that teams can focus on application-level business logic, without unnecessary ties to specific deployment environments."

Spring은 어떤 종류의 배포 플랫폼에서도 최신 자바 기반 기업용 애플리케이션을 위한 종합적인 프로그래밍 및 구성 모델을 제공해준다.

Spring의 핵심 요소는 애플리케이션 수준에서의 인프라 지원이다. Spring은 기업용 애플리케이션의 plumbing에 초점을 맞춰 팀이 특정 배포 환경과 불필요한 시도 없이 애플리케이션 수준의 비즈니스 로직에 집중할 수 있게 해준다.

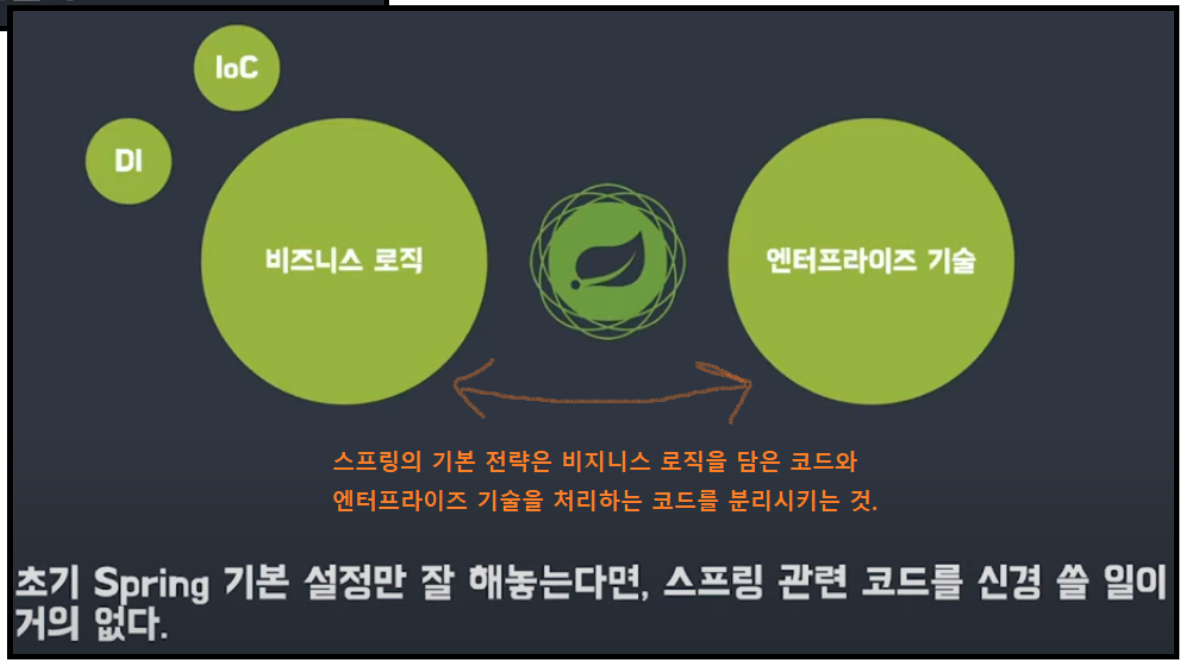
스프링 프로젝트들은

스프링 프레임워크를 기반으로 돌아감.

>>스프링 프레임워크는 스프링의 핵심

스프링 프레임워크는 아래의 사진과 같이 분리를 통해 객체 지향의 특징을 잘 활용할 수 있게 해주며 개발자들은

핵심 비지니스 로직 구현에만 집중할 수 있게 해주는 프레임워크



스프링의 기본 전략은 비지니스 로직을 담은 코드와 엔터프라이즈 기술을 처리하는 코드를 분리시키는 것.

초기 Spring 기본 설정만 잘 해놓는다면, 스프링 관련 코드를 신경 쓸 일이 거의 없다.

Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run".

We take an opinionated view of the Spring platform and third-party libraries so you can get started with minimum fuss. Most Spring Boot applications need minimal Spring configuration.

Spring Boot는 독립적이며, 운영 할 수 있는 수준의 Spring 기반 애플리케이션을 쉽게 만들 수 있게 해준다. 그냥 실행해라.

최소한의 설정으로 Spring 플랫폼과 서드파티 라이브러리를 사용할 수 있다. 대부분 Spring Boot 애플리케이션은 최소한의 Spring 설정을 필요로한다.

》 즉, Spring Boot는 Spring Framework 기반의 App을 쉽게 만들 수 있게 해준다.

Spring Boot는 Spring 프로젝트 중 하나로, Spring Framework를 쉽게 사용하게 해주는 도구이지, Spring Framework와 별개로 사용할 수 있는 것이 아니다.

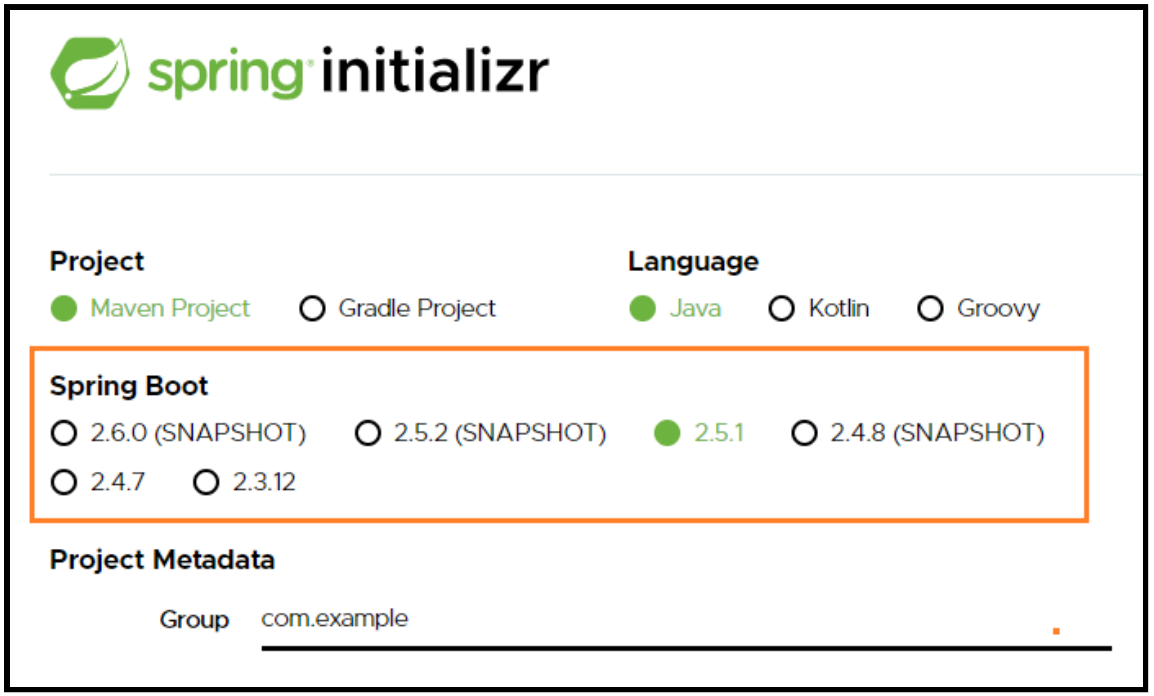
왜 Spring Boot가 개발되었는가??

》 Spring Framework가 점점 다양한 기술들을 지원하게 되면서

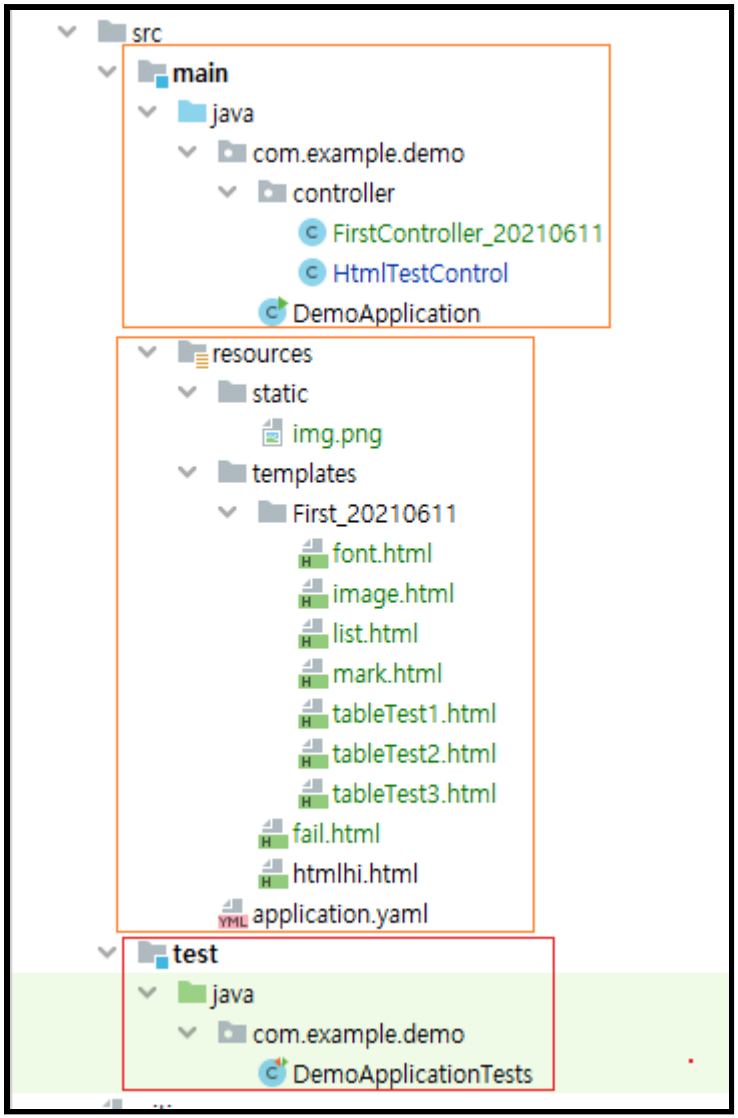
'App개발에 필요한 객체 생성 + 객체 사이의 의존성을 제공'하는 초창기 Spring Framework의 역할에 비해 개발자가 처리해야되는 설정도 많아지고 복잡해졌기 때문.

상단에 framework 설명시 '초기설정'을 잘 해야 한다고 했는데 그 초기설정이 너무 어려워졌다는 것.

그것을 해결하기 위해 Spring Boot를 개발. 》 즉, Boot가 초기설정 다 해준다. 넌 써라



내가 수업시간에 사용했던 Spring Initializr도 Spring Boot를 사용



《Project 구조》

《 main

controller: Mapping하는(?) controller들을 배치

xxxApplication: `@SpringBootApplication`

《 resources

static: 자료들 배치

templates: html파일들 배치

《 application: configuration data를 포함

《 Q. test하는 class는 정확히 어떤 기능을 하나요???

Q. lombok은 이클립스용 라이브러리인지?? 그래서 현재 우리는 따로 설치를 안 한 상태가 맞는지 ??

```java
package com.example.demo.controller;
import lombok.extern.slf4j.Slf4j;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
@Slf4j
@Controller
public class FirstController_20210611 {
```

《@Slf4j》

logging: 운영체제나 소프트웨어가 실행 중에 발생하는 event를 기록하는 행위.

logging은 점을 치는 행위와 유사하다고 하는데
그 이유는. error가 생겼을 때 미리 예상해서 그 때 필요할 것 같은 정보를 짐작해서 logging해야 하기 때문.

》 서비스 동작 상태 파악 / Error파악 / ErrorAlarm

그렇다면 Logging은 어떻게 하는가?

로깅 라이브러리를 활용 》 JCL/slf4j/log4j/logback 등...》 SpringBoot에서는 Slf4j를 활용

Slf4j: 로깅 추상화 라이브러리

- logging level

FATAL: 매우 심각한 ERROR가 발생한 경우. 이 레벨이 사용될 경우 프로그램이 종료되는 경우가 많다. 프로그램이 정상적으로 종료되지 않는 경우라 로그가 남는다는 보장하기 어려움. 따라서 이 레벨은 최대한 사용하지 않는게 좋다고 한다.

ERROR: ERROR가 발생했지만, 프로그램이 종료되지 않는 경우.

--------- ERROR / FATAL의 경우, 의도하지 않은 exception경우에 사용하는 것이 좋다. ----------

WARN: ERROR가 될 수 있는 잠재적 가능성이 있는 경우. 알람이 오도록 설정하여 ERROR가 나기 전 조치를 취하거나 ERROR가 나면 그전의 상황을 알 수 있다.

INFO: APP의 상태를 간결하게 보여주는 경우. '서비스가 시나리오대로 잘 동작하고 있는가?' 같은 상황에 그 상태를 간결하게 보여주는 사용.

DEBUG: INFO레벨 보다 더 자세한 정보가 필요한 경우. 권한이 없어 디버깅이 불가능한 경우에 유용하다.

TRACE: DEBUG 레벨 보다 더 자세한 정보가 필요한 경우. 개발환경에서 버그를 해결하기 위해 사용한다. 최종 프로덕션이나 커밋에 포함하면 안된다.

```java
@GetMapping("/htmlList")
public String doHtmlList(){
    log.info("doHtmlList()");
    return "First_20210611/list";
}
```

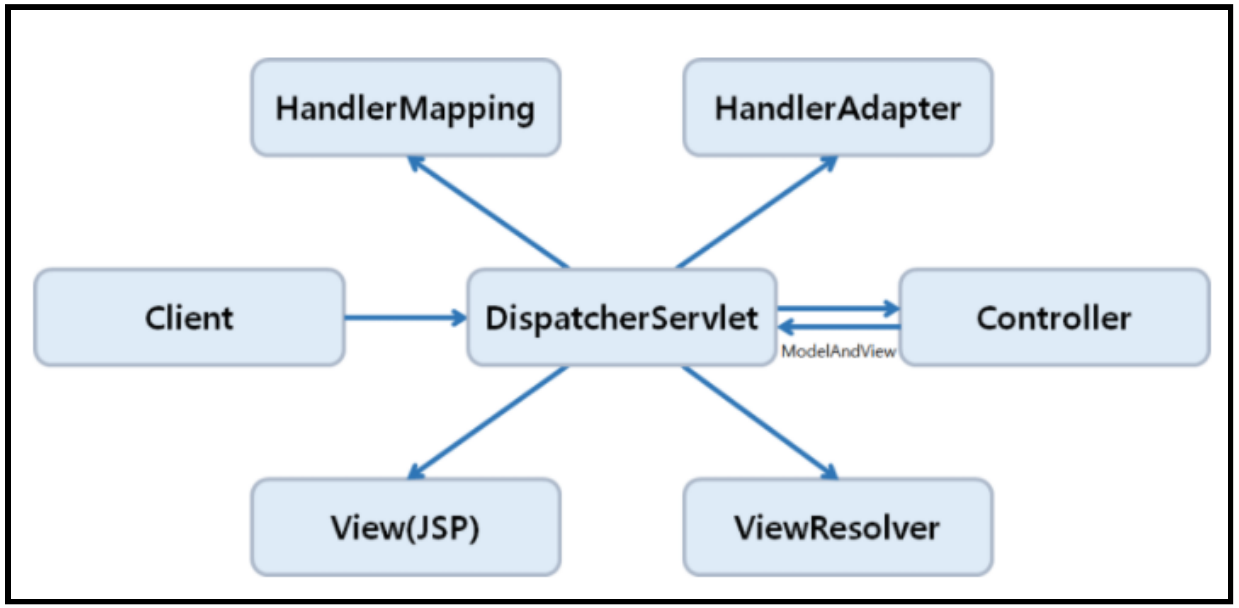
출처: https://www.youtube.com/watch?v=MxxeKXydn4A&t=670s

```
@Slf4j
@Controller
public class FirstController_20210611 {
```

그렇다면 @Controller는 무엇인가

《@Controller》

먼저 SPRING의 MVC를 보면



웹 브라우저를 통해 클라이언트 요청이 들어오면 모든 요청을 DispatcherServlet이 받는다.
그 후 이 작업을 HandlerMapping, HandlerAdapter 등..
다른 곳으로 보내준다.

개발 시 주로 다루게 될 부분은 Model, View, Controller 부분이며 그 외의 부분은 스프링 프레임워크에서 자동으로 다루어준다.

DispatcherServlet에서 일방적으로 보내는 것이 아닌 요청을 주고받는 곳이 있다. 바로 Controller.

1. DispatcherServlet은 클라이언트로부터 받은 요청을 Controller에게 전송한다.
2. 요청을 받은 Controller는 DispatcherServlet에게 응답을 준다.
3. DispatcherServlet은 ViewResolver를 통해 View를 호출한다.

Spring MVC 구조로 살펴보았듯이 Controller는 클라이언트로부터 요청이 들어왔을 때

DispatcherServlet을 통해 Controller로 진입하게 된다.

그 후 Controller는 해당 요청에 대한 작업을 마친 후 VIEW쪽으로 데이터를 전달한다.

`@Controller`는 Controller의 역할을 할 클래스에 @(어노테이션) + Controller를 사용해서
'이 class가 Controller 역할을 수행할 것이다' 라고 명시하는 것.

출처:

https://blog.naver.com/jjekjjek7/222186372398

https://elfinlas.github.io/2017/12/14/java-annotation/

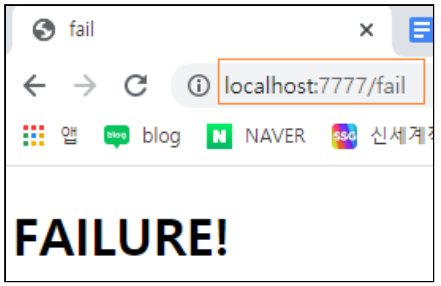
```java
package com.example.demo.controller;
import lombok.extern.slf4j.Slf4j;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
@Slf4j
@Controller
public class FirstController_20210611 {

    // 'Get' / 'Post' > 실제 웹상에서 URL 요청할 때 자주 사용하는 두 가지 방식.
    // 일반적인 URL 입력은 Get이라고 파악해둘 것.
    // @GetMapping("/") > ip:port의 Home > 'naver.com/' = 'naver.com'
    @GetMapping("/fail")
    //GetMapping은 바로 아래의 method까지만 바라봄(?)
    public String dofail(){
        log.info("dofail() 실행");
        return "fail";
    }

    @GetMapping("/htmlList")
    public String doHtmlList(){
        log.info("doHtmlList()");
        return "First_20210611/list";
    }

    @GetMapping("/imageTest")
        public String doImageTest(){
            log.info("doImagetTest");
            return "First_20210611/image";
        }
```

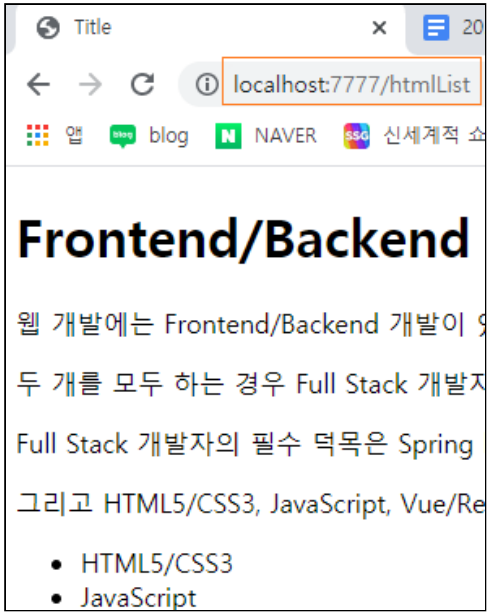
templates
- First_20210611
  - font.html
  - image.html
  - list.html
  - mark.html
  - tableTest1.html
  - tableTest2.html
  - tableTest3.html
- fail.html
- htmlhi.html
- application.yaml

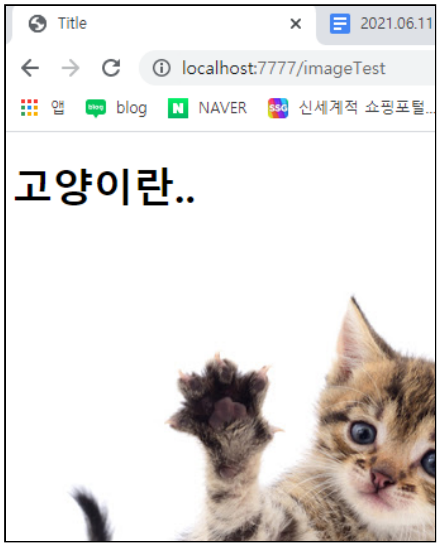
test

〈 App 실행 관련 〉

〈〈 Controller







FirstController_20210611.java | tableTest3.html | DemoApplication.java

```java
package com.example.demo;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class DemoApplication {
    public static void main(String[] args) { SpringApplication.run(DemoApplication.class, args); }
```

App 실행하는 class

Q. `@SpringBootApplication`**이 App 실행하는 class에 사용하는 어노테이션이 맞나요??**

```
2021-06-13 23:03:10.826  INFO 19460 --- [nio-7777-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2021-06-13 23:03:10.827  INFO 19460 --- [nio-7777-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 1 ms
2021-06-13 23:03:10.846  INFO 19460 --- [nio-7777-exec-1] c.e.d.c.FirstController_20210611         : doImagetTest
2021-06-13 23:03:16.935  INFO 19460 --- [nio-7777-exec-4] c.e.d.c.FirstController_20210611         : dofail() ����
2021-06-13 23:05:24.373  INFO 19460 --- [nio-7777-exec-9] c.e.d.c.FirstController_20210611         : doHtmlList()
2021-06-13 23:06:31.428  INFO 19460 --- [nio-7777-exec-2] c.e.d.c.FirstController_20210611         : doImagetTest
2021-06-13 23:07:40.531  INFO 19460 --- [nio-7777-exec-6] c.e.d.c.FirstController_20210611         : doFontTest
2021-06-13 23:19:10.872  INFO 19460 --- [nio-7777-exec-9] c.e.d.c.FirstController_20210611         : doImagetTest
2021-06-13 23:23:19.673  INFO 19460 --- [nio-7777-exec-3] c.e.d.c.FirstController_20210611         : doHtmlList()
2021-06-13 23:26:26.546  INFO 19460 --- [nio-7777-exec-6] c.e.d.c.FirstController_20210611         : domarkTest
```

App을 실행 후 run 부분에 보면 내가 접속 할 때마다 저런 식으로 log가 찍힘.

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <!-- style에 적용하면 전체에 적용됨 -->
    <style>
        p {
            font-size: 32px;  /* 글자 크기 */ /* << css에서 주석다는 법 */
            line-height: 45px; /* 줄 간격 */
        }
    </style>
</head>
<body>
<!-- strong:, b:강조 -->
<!-- em:, i: 이탤릭체 -->
<!-- br: 줄바꿈 -->
    <h2>go Language</h2>
    <p><strong>Rust</strong>, C, <b>jAVA</b> Python</p>
    <p><em>List</em>, Fortran, <i>JavaScript</i>, asm <br> FPGA Verilog</p>
</body>
</html>
```

전체 적용되는 CSS

〈HTML〉

go Language

Rust, C, **jAVA** Python

List, Fortran, *JavaScript*, asm
FPGA Verilog

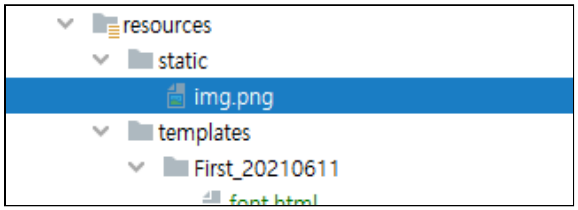
〈!-- HTML 주석 --〉

/* CSS 주석 */

Q. SRC가 RESOURCES 폴더 쪽에서 해당 파일을 찾겠다는 KEYWORD인지??

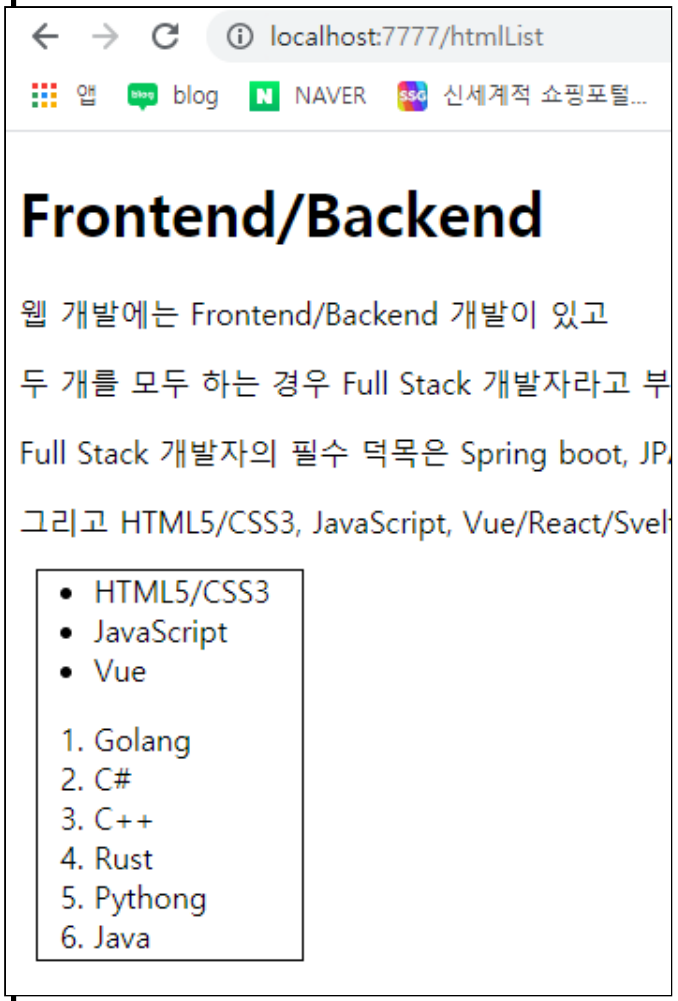
image.html | HtmlTestContro

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
    <h1>고양이란..</h1>
    <img src = "img.png">
    <p>고양이</p>
</body>
</html>
```



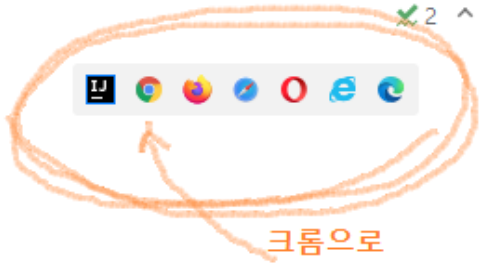
tml × image.html × FirstController_20210611.java × list.html × DemoApplication.java ×

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<!-- h1 ~ h6는 글자의 크기 속성을 지정할 수 있음 -->
  <h1>Frontend/Backend</h1>
  <p>웹 개발에는 Frontend/Backend 개발이 있고</p>
  <p>두 개를 모두 하는 경우 Full Stack 개발자라고 부릅니다.</p>
  <p>Full Stack 개발자의 필수 덕목은 Spring boot, JPA+ Alpha(Kafka, etc...)</p>
  <p>그리고 HTML5/CSS3, JavaScript, Vue/React/Svelte, Typescript 등 입니다.</p>

<!-- 'li*3 + tab'하면 li 3개 한 번에 생김  -->
  <ul> <!-- unordered list -->
    <li>HTML5/CSS3</li>
    <li>JavaScript</li>
    <li>Vue</li>
  </ul> <!-- ordered list -->
  <ol>
    <li>Golang</li>
    <li>C#</li>
    <li>C++</li>
    <li>Rust</li>
    <li>Pythong</li>
    <li>Java</li>
  </ol>
</body>
```



ontroller_20210611.java | mark.html | list.html | DemoApplication.java

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<!-- mark: 형광펜 기능 -->
<!-- span: 줄바꿈 없이 영역을 묶음, style="color: x;" 특정 색상으로 만듬 -->
<!-- 배경색을 설정하고 싶다면 background-color를 사용 -->
  <h2>Python Language: <mark style="color: pink;">"TensorFlow" ===> "Keras"</mark></h2>
  <p>Machine Lerning, <span style="color: blue;">Deep Learning, </span> Statistics</p>
  <h3>Test <mark style = "background-color: orange;">Driven Development</mark></h3>
</body>
</html>
```



크롬으로 TEST 할 수 있음

span

The **HTML element** is a generic inline container for phrasing content, which does not inherently represent anything. It can be used to group elements for styling purposes (using the `class` or `id` attributes), or because they share attribute values, such as `lang`. It should be used only when no other semantic element is appropriate. is very much like a `<div>` element, but `<div>` is a block-level element ↗ whereas a is an inline element ↗.

ontroller_20210611.java × tableTest1.html × tabl

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <style>
      table, th, td { /*
        border: 1px solid #cccccc;
      }
      td {
        padding: 5px;
      }
    </style>
</head>
<body>
  <table>
    <tr> <!-- tr:행 -->
      <th>[0][0]</th> <!-- th/td:열 -->
      <th>[0][1]</th>
      <th>[0][2]</th>
    </tr>
    <tr>
      <th>[1][0]</th>
      <th>[1][1]</th>
      <th>[1][2]</th>
    </tr>
  </table>
</body>
</html>
```

[0][0] [0][1] [0][2]
[1][0] [1][1] [1][2]

Q. 왜 표가 안 생길까요..?

›› A. 멍청하게 CSS에 /* 처리 해놓고 질문하고 있었음 나중에 수정할 것

ntroller_20210611.java | tableTest2.html | tableTest3.html | DemoApplication.java

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
  <style>
    table, th, td {
      border: 1px solid #000000;
    }
    th {
      padding: 15px; /* 셀 테두리와 내용 사이의 간격(패딩) */
    }
    tr > td:nth-child(odd) {
      width: 120px; /* 홀수번째 열의 너비 지정 */
    }
    tr > td:nth-child(even) {
      width: 300px; /* 짝수번째 열의 너비 지정 */
    }
  </style>
</head>
<body>
  <table>
    <tr>
      <th>Name</th> <!-- th는 td의 강조형 -->
      <td></td>
      <th>Contact</th>
      <td></td>
    </tr>
    <tr>
      <th>Address</th>
      <td colspan="3"></td> <!-- colspan="3"은 열 3개를 하나로 묶음 -->
    </tr>
    <tr>
      <th>Self Introduction</th>
      <td colspan="3"></td>
    </tr>
  </table>
</body>
</html>
```

Name		Contact	
Address			
Self Introduction			

ontroller_20210611.java × tableTest3.html × DemoApplication.java ×

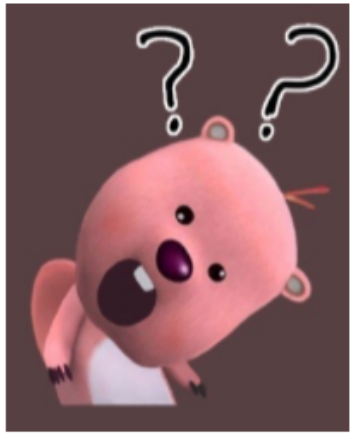
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
  <style>
    table, th, td {
      border: 1px solid #000000;
    }
    td, th {
      padding: 10px; /* 셀 테두리와 내용 사이의 간격(패딩) */
    }
  </style>
</head>
<body>
  <table>
    <caption> <!-- caption, figcaption을 통해 표에 제목을 붙일 수 있음 -->
      <strong>이력서(Resume)</strong>
      <p>i am.....</p>
    </caption>
    <tr>
      <th>Browser</th>
      <th>Vendor</th>
      <th>DownLoads</th>
    </tr>
    <tr>
      <th>chrome</th>
      <td>Google</td>
      <td>https://www.google.com/chrome/</td>
    </tr>
    <tr>
      <th>Firefox</th>
      <td>Mozilla</td>
      <td>https://www.mozilla.org/ko/firefox</td>
    </tr>
  </table>
</body>
```

이력서(Resume)

i am.....

Browser	Vendor	DownLoads
chrome	Google	https://www.google.com/chrome/
Firefox	Mozilla	https://www.mozilla.org/ko/firefox

2_zu Resume



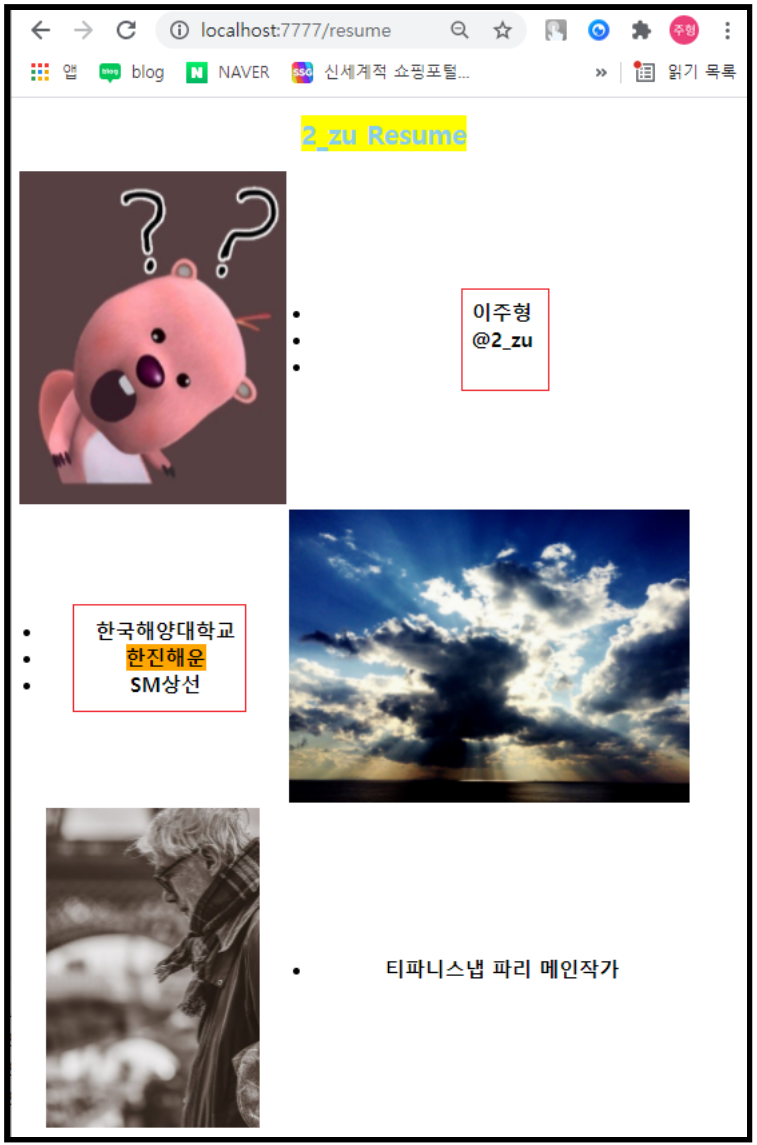
- 이주형
- @2_zu
- spmen12@gmail.com

- 한국해양대학교
- 한진해운
- SM상선

- 티파니스냅 파리 메인작가

〈Quiz.64〉
자기소개서

›› 넣을 것도 없는 것 같고 어떻게 만들어야 할지 감도 안 와서
배운 것 복습 한다는 느낌으로 이것 저것 넣어 봤는데 질문 생김.



```
<table>
    <tr>
        <th><img src = "pic.jpg" width="400" height="500"></th>
        <th>
            <ul>
                <li><span style="red">이주형</span></li>
                <li><span style="red">@2_zu</span></li>
                <li><style color="blue">spmen12@gmail.com</style></li>
            </ul>
        </th>
    </tr>
    <tr>
        <th>
            <ul>
                <li><b>한국해양대학교</b></li>
                <li><mark style="background-color:orange;">한진해운</mark></li>
                <li>SM상선</li>
            </ul>
        </th>
```

Q. li tag 안에 적용 시킨 style 값이 왜 적용이 안될까요?

```html
<head>
    <meta charset="UTF-8">
    <title>2_zu</title>
    <style>
        h1 {
        text-align: center;
        font-size: 40px;
        }
        ul{
        font-size: 30px;
        }
    </style>
</head>
```

CSS 이것저것 넣어보기 전

HTML

```html
<body>
<h1>2_zu Resume</h1>
<table>
    <tr>
        <th><img src = "pic.jpg" width="400" height="500"></th>
        <th>
            <ul>
                <li>이주형</li>
                <li>@2_zu</li>
                <li>spmen12@gmail.com</li>
            </ul>
        </th>
    </tr>
    <tr>
        <th>
            <ul>
                <li>한국해양대학교</li>
                <li>한진해운</li>
                <li>SM상선</li>
            </ul>
        </th>
        <th><img src = "IMG_3017.JPG" width="600" height="440"></th>
    </tr>
    <tr>
        <th><img src = "eee.jpg" width="320" height="480"></th>
        <th>
            <ul>
                <li>티파니스냅 파리 메인작가</li>
            </ul>
        </th>
    </tr>
</table>
</body>
```