



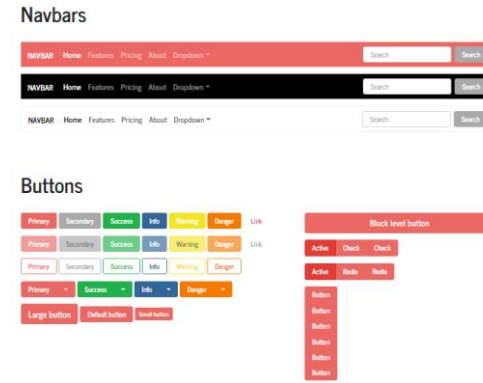
Spring Boot

소프트웨어융합공학과
웹서버 프로그래밍
허우행

Theme vs Template

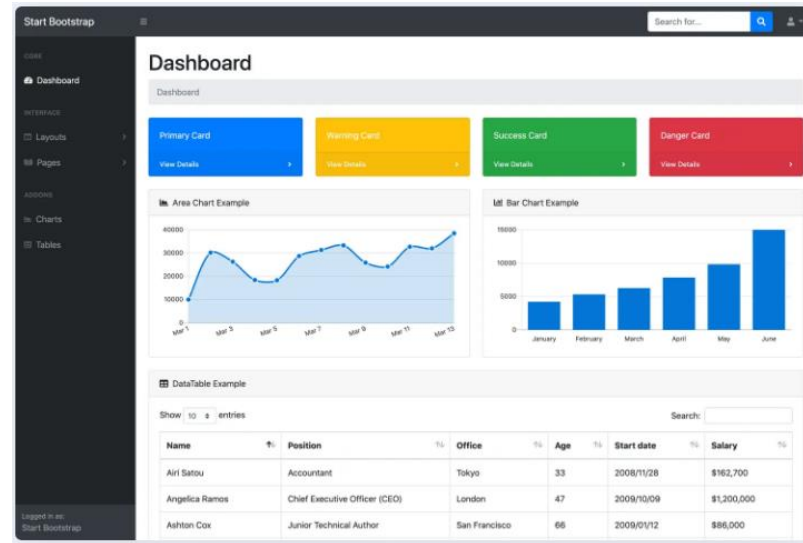
▶ Theme란?

- ▶ Theme는 스타일 레이어 또는 "스킨"을 의미.
- ▶ Theme는 색상, 글꼴 및 스타일을 포함.
- ▶ 단순 css 파일을 의미
- ▶ <https://bootswatch.com/>



▶ Template란?

- ▶ Template은 페이지 레이아웃 및 해당 HTML 구조.
- ▶ <https://startbootstrap.com/>



Thymeleaf

- ▶ <https://www.thymeleaf.org/>
- ▶ HTML, XML, JavaScript, CSS 및 일반 텍스트를 처리 할 수 있는 Java 템플릿 엔진
- ▶ Thymeleaf는 html파일을 가져와서 파싱해서 분석 후 정해진 위치에 데이터를 치환해서 웹 페이지를 생성
- ▶ Layout 기능 제공(독립적 구동 가능 형태로 존재)

Build.gradle(의존성 추가)

```
implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'  
implementation 'org.thymeleaf.extras:thymeleaf-extras-springsecurity5'  
implementation 'org.springframework.boot:spring-boot-starter-security'  
implementation 'nz.net.ultraq.thymeleaf:thymeleaf-layout-dialect'
```

Thymeleaf 기본

Thymeleaf 권한처리

스프링부트 권한처리

Thymeleaf layout 처리

Thymeleaf

default_layout.html

```
<!DOCTYPE html>

<html lagn="ko"
  xmlns:th="http://www.thymeleaf.org"
  xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
  xmlns:sec="http://www.thymeleaf.org/extras/spring-security"

  <head>
    <meta charset="UTF-8" />
    <title layout:title-pattern="$LAYOUT_TITLE - $CONTENT_TITLE">인하대학교</title>

    <!-- config fragment 사용 -->
    <th:block th:replace="fragment/config :: configFragment" ></th:block>
  </head>

  <body>
    <!-- header fragment 사용 -->
    <th:block th:replace="fragment/header :: headerFragment"></th:block>

    <th:block layout:fragment="content"></th:block>

    <!-- footer fragment 사용 -->
    <th:block th:replace="fragment/footer :: footerFragment"> </th:block>
  </body>

</html>
```

Thymeleaf 기본

Thymeleaf layout 처리

Thymeleaf 권한처리

- layout:title-pattern : layout 타이틀과 홈의 타이틀 조합하여 사용
- th:replace= " fragment경로 :: fragment이름 " 속성은 해당 영역을 fragment로 치환
- layout:fragment="content"는 해당 layout을 사용하는 content의 내용을 불러옴

Thymeleaf

home.html(실제 보여질 화면)

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org"
      xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
      layout:decorator="layout/default_layout">

  <!-- index.html 고유 CSS 추가 -->
  <th:block layout:fragment="css">
    <link rel="stylesheet" th:href="@{/css/page/home.css}" >
  </th:block>

  <!-- index.html 고유 스크립트 추가 -->
  <th:block layout:fragment="script">
    <script th:src="@{/js/page/home.js}"></script>
  </th:block>

  <div layout:fragment="content">
    <h1>content</h1>
  </div>
</html>
```

header.html, footerFragment.html, configfragment.html

```
<html lagn="ko"
      xmlns:th="http://www.thymeleaf.org">

  <!--headerFragment 선언-->
  <div th:fragment="headerFragment">
    <h1>HEADER</h1>
  </div>

</html>
```

```
<html lagn="ko"
      xmlns:th="http://www.thymeleaf.org">

  <!--footerFragment 선언-->
  <div th:fragment="footerFragment">
    <h1>FOOTER</h1>
  </div>

</html>
```

```
<html lagn="ko"
      xmlns:th="http://www.thymeleaf.org"
      xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout">

  <!--configFragment 선언-->
  <th:block th:fragment="configFragment">

    <!-- 이 영역에 공통으로 사용할 css, js library를 선언한다. -->
    <link rel="stylesheet" th:href="@{/css/common/common.css}" >

    <script th:src="@{/js/common/common.js}"></script>

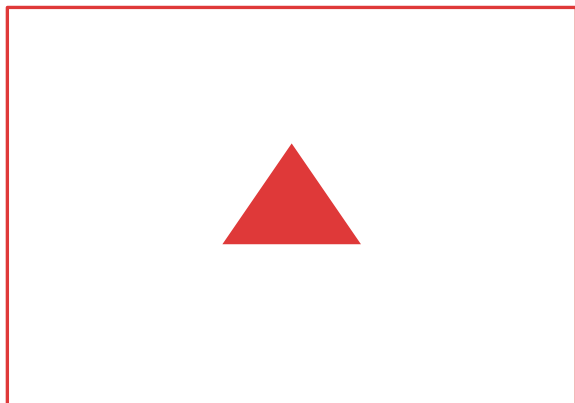
    <!-- Content Page의 CSS fragment 삽입 -->
    <th:block layout:fragment="css"></th:block>

    <!-- Content Page의 script fragment 삽입 -->
    <th:block layout:fragment="script"></th:block>

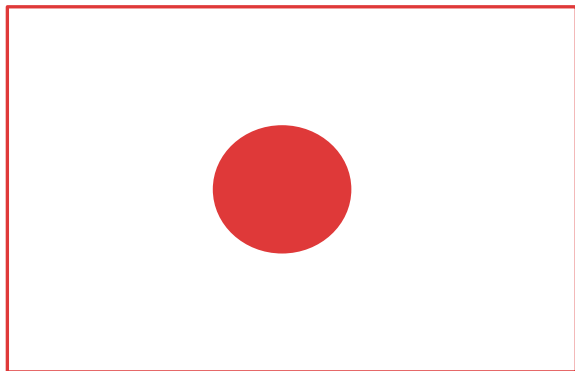
  </th:block>
</html>
```

기존 결합 방식

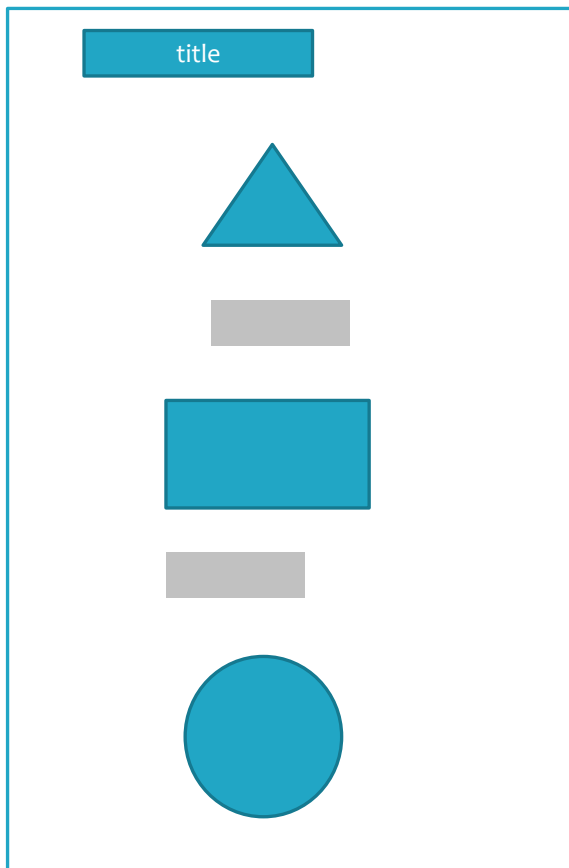
header.html



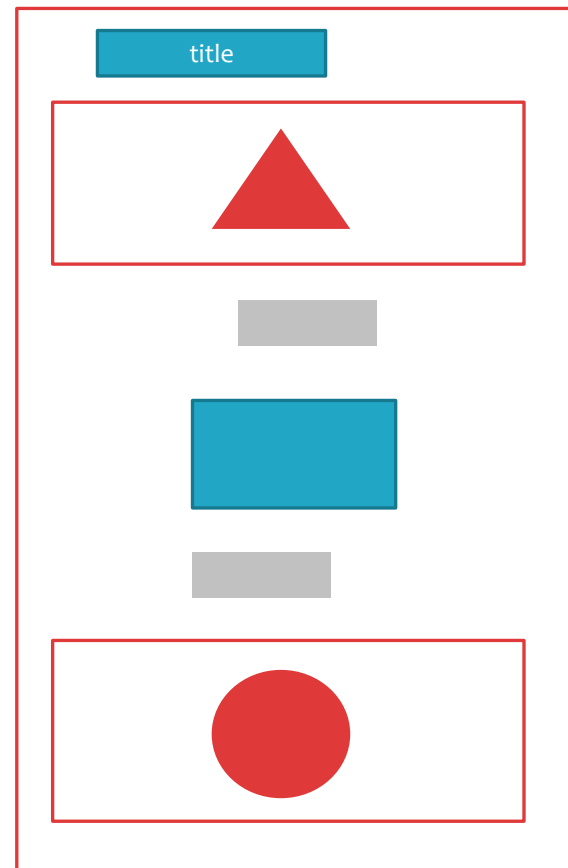
footer.html



home.html



완성(결합)

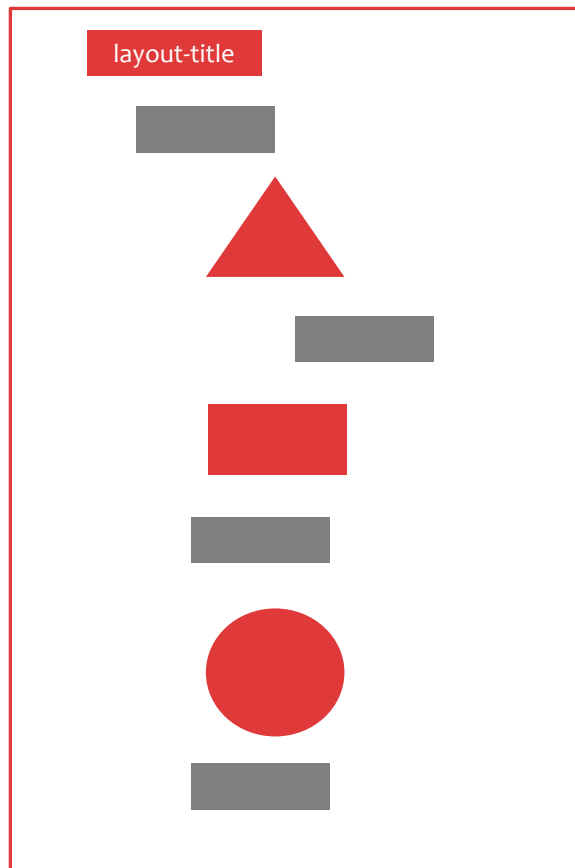




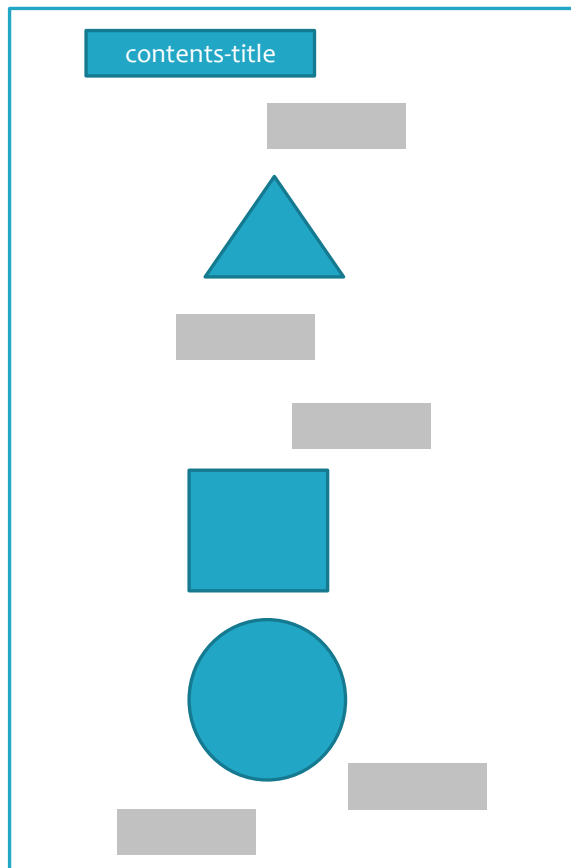
Template 결합



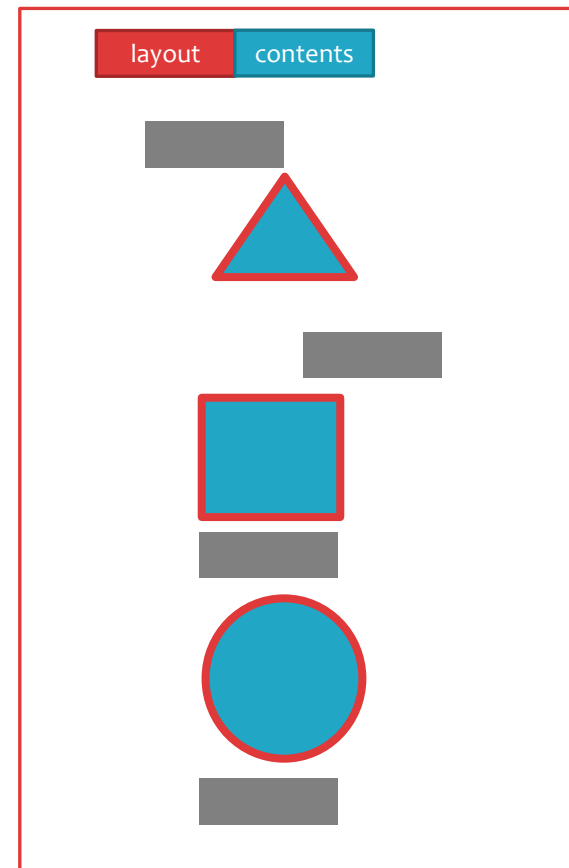
default_layout.html



home.html



완성(결합)



※ 각 파일은 독립적 구동 가능한 완전한 HTML 형태로 존재