

(디지털 컨버전스) 스마트 콘텐츠와 웹 융합 응용 SW개발자 양성과정

훈련기간 : 2021.05.07 ~ 2021.12.08

Modify 기능 설정

```
@GetMapping("/modify")
public String getModify(int boardNo, Model model) throws Exception {
    log.info("getModify()");

    //수정을 눌렀을때 기존 정보를 가져와야함
    model.addAttribute(service.read(boardNo));

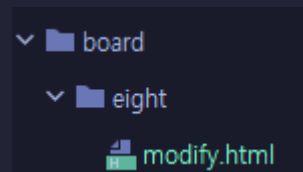
    return "board/eight/modify";
}

@PostMapping("modify")
public String postModify(Board board, Model model) throws Exception {
    log.info("postModify()" + board);

    service.modify(board);

    model.addAttribute(attributeName: "msg", attributeValue: "수정이 성공적으로 완료되었습니다");

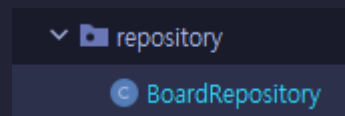
    return "/board/fourth/success";
}
```



주의 : boardNo 가져올것! 안가져오면 No가 0됨

```
<input type="hidden" name="boardNo" th:field="*{boardNo}"/>
```

마찬가지로 인터페이스와 구현부 만들고 저장소에 적용



```
public void update(Board board) throws Exception {
    String query = "update board set title = ?, content = ? where board_no = ?";

    jdbcTemplate.update(query, board.getTitle(), board.getContent(), board.getBoardNo());
}
```

HTML에서 기능을 분리시킨다 == CSS

결론 : 깔끔해짐

방법 : class(.), id(#), style

classSelect.css

```
.greenBoldText {  
    color: green;  
    font-weight: bold;  
}
```

```
<link rel="stylesheet" href="/css/basic/classSelect.css">  
</head>  
<body>  
<div id="container">  
    <h1>Introduction</h1>  
  
    <h2 class="greenBoldText">Mission</h2>  
  
    <h3>Acheivement</h3>  
</div>
```

idSelect.css

```
#container {  
    width: 600px;           /* 너비 */  
    padding: 15px;         /* 테두리와 내용 사이의 여백 */  
    border: 1px dotted gray; /* 테두리 스타일 */  
    margin: 20px;  
}
```

```
<link rel="stylesheet" href="/css/basic/idSelect.css">  
</head>  
<body>  
<div id="container">  
    <h1>Introduction</h1>  
  
    <h2>Mission</h2>  
  
    <h3>Acheivement</h3>  
</div>
```

style.css

```
h2 {  
    color: blue;  
}
```

```
<link rel="stylesheet" href="/css/basic/style.css">  
</head>  
<body>  
    <h1>CSS 테스트</h1>
```

정렬

```
.align-left {  
    text-align: left;  
}
```

```
.align-right {  
    text-align: right;  
}
```

```
.align-center {  
    text-align: center; /*가운데 정렬*/  
}
```

```
.align-justify {  
    text-align: justify; /*양쪽정렬*/  
}
```

BoxHover



```
.box:hover {  
    -webkit-transform: rotate(15deg);  
    -moz-transform: rotate(15deg);  
    -o-transform: rotate(15deg);  
    -ms-transform: rotate(15deg);  
    transform: rotate(60deg);  
}
```

shadow



```
}  
.movieFeelShadow {  
    color: #000000;  
    text-shadow: 0 0 4px #fcfcfc, 0 -5px 4px #ffff33, 2px -10px 6px #ffee33, -5px -10px 6px #ffff33;  
}
```

font 구글폰트사용

Use on the web

To embed a font, copy the code into the <head> of your html

☐ <link> ☒ @import

```
<style>
@import url('https://fonts.googleapis.com/css2?family=Roboto&display=swap');
</style>
```

CSS rules to specify families

```
font-family: 'Roboto', sans-serif;
```

```
@import url('https://fonts.googleapis.com/css2?family=Roboto&display=swap');
.ng-font {
  font-family: 'Roboto', '로보토';
}
p {
  font-size: 30px;
}
```

Browser Basic Font

Roboto Google web Font - Browser Basic Font

margin

CSS 테스트

웹 개발 패턴은 계속 똑같아요.

한번 익히고 사골국을 우립니다.

물론 변하지 않는 형식과 변화가 필요한 형식이 있습니다.

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    <h1>CSS 테스트</h1>
    <h2>웹 개발 패턴은 계속 똑같아요.</h2>
    <p>한번 익히고 사골국을 우립니다.</p>
    <p>물론 변하지 않는 형식과 변화가 필요한 형식이 있습니다.</p>
  </body>
</html>
```

Styles Computed Layout Event Listener

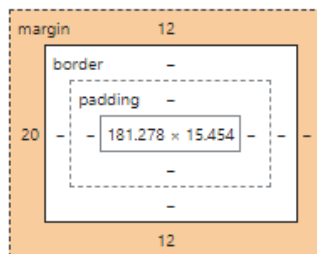
Filter

:hov .cls

```
element.style {
}
```

```
p {
  font-size: 12px;
  margin-left: 20px;
}
```

```
p {
  display: block;
  margin-block-start: 1em;
  margin-block-end: 1em;
  margin-inline-start: 0px;
  margin-inline-end: 0px;
}
```



```
h2 {
  color: blue;
}
p {
  font-size: 12px;
  margin-left: 20px;
}
```

```
<link rel="stylesheet" href="/css/basic/margin.css">
</head>
<body>
  <h1>CSS 테스트</h1>
  <h2>웹 개발 패턴은 계속 똑같아요.</h2>
  <p>한번 익히고 사골국을 우립니다.</p>
  <p>물론 변하지 않는 형식과 변화가 필요한 형식이 있습니다.</p>
</body>
</html>
```