

Quiz #1 웹에 접속한 사용자의 정보를 사용자 컴퓨터에 저장하면 이 장소를 ()라고 하며, 서버에 저장하면 이 장소를 ()이라고 합니다.

Quiz #2

주소창에 localhost:8080/bank 로 접속하면 a.html이 화면이 뜨며, a.html 안에 폼이 있는데 `<form action="/bank/deposit">` 라고 정의되어 있습니다.

이 폼에서 입력한 데이터를 전송받아서 session 에 넣을 수 있는 메소드의 annotation은 몇 번일까요?

- ① @GetMapping("/bank")
- ② @PostMapping("/bank")
- ③ @GetMapping("/bank/deposit")
- ④ @PostMapping("/bank/deposit")

Quiz #3 car.html 화면에서 입력한 차, 색상 정보를 전송 받아서 세션에 넣은 후 carSave.html 화면을 띄워주는 컨트롤러 메소드를 작성하세요.



car.html
(보내는 쪽)

```
<form method="post" action="/car/save" >
  차: <input type="text" name="car"> <p>
  색상: <select name="color">
    <option> black <option> yellow
    <option> red   <option> white </select>
  <button type="submit">선택</button>
</form>
```

Quiz #4

- (1) 컨트롤러의 @GetMapping, @PostMapping 괄호 안의 주소가 동일해도 될까요?
- (2) 컨트롤러의 메소드명이 동일해도 될까요?
- (3) 컨트롤러의 모델이나 세션을 매개변수 말고 일반 지역변수로 선언해도 될까요?
- (4) 컨트롤러의 모델이나 세션을 여러 개 선언해도 될까요?
- (5) 컨트롤러의 모델이나 세션 매개변수명을 model, session 이라고 하지 않고 mo, se 하고 하는 이유는?

Quiz #5 new Actor("spider","Tom Holland", 28)를 하면 아이디가 spider
이름이 Tom Holland, 나이가 28인 Actor객체가 생성되도록
Actor.java를 완성하세요.

```
public class Actor {
```

```
} // class
```

Quiz #6 로그인 체크 소스의 빈 부분을 채우세요.

```
@GetMapping("/login/check")
public String loginCheck( ) {
    var arr = new ArrayList<Member>();
    arr.add(new Member("고흐", "g"));
    arr.add(new Member("dooli", "d"));

    arr.add(new Member("james", "j"));
    arr.add(new Member("iu", "i"));

    boolean ourMember=false;
    for(var a: arr) {
        if(a.mid.equals(mid) && a.pw.equals(pw))
            { ourMember = true; break; }
    }

    if( ourMember ) {
        /* 아이디를 세션에 저장 후 */
        /* 메뉴 메소드가 실행되도록 함 */
    } else {
        se.setAttribute("msg", mid + "는 미등록 아이디이거나 혹은 패스워드가 일치하지 않습니다.\n"
            + "확인 후 로그인 부탁드립니다.");
        return "redirect:/popup";
    }

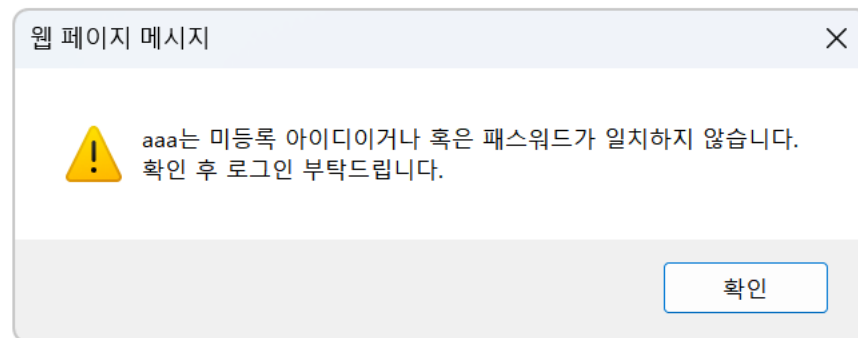
} // method
```

Quiz #7 모델에 있는 msg를 팝업창에 띄우는 아래 소스에서 오류를 모두 찾으세요.

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head><meta charset="UTF-8">
<title>안내글</title></head>
<body>

<style th:texts="javascript">
    popup([[${"msg"}]]);
    location.href="/login";
</style>

</body>
</html>
```



Quiz #8 메뉴 소스의 빈 부분을 채우세요.

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head><meta charset="UTF-8"> <title>MENU</title>
<link rel="stylesheet" href="/style.css">
<style>
    body {padding: 0px 40px 0px 40px;
          background:lightgray; }
</style>
</head><body>
```



```
<ul>
    <li>bread
    <li><a href="/coffee">coffee</a>
    <li>salad
    <li><a href="/logout">logout</a>
</ul>
</body></html>
```

MENU

dooli님, 오늘도 즐겁게 쇼핑하세요!

- bread
- [coffee](#)
- salad
- [logout](#)

```
@GetMapping("/menu")
public String menu( ) {
    mo.addAttribute("mid", se.getAttribute("mid"));
    return "menu";
}
```


Quiz #9

세션이나 쿠키를 사용하는 이유 중 하나는 "고객님이 어느 화면을 먼저 방문하실 지 아무도 모른다" 입니다.

그런데 예를 들어 쇼핑몰에서 클릭해야하는 화면 순서가 미리 정해졌다고 가정(예를 들어 로그인후 무조건 티셔츠 → 바지 → 운동화 순서대로 쇼핑해야 한다고 가정)하는 경우에도 세션/쿠키를 사용합니다.

순서가 정해졌으므로 화면끼리 토스를 계속 하면 되는데 왜 세션/쿠키를 사용할까요?

Quiz #10 세션을 이용해서 설문조사 시스템을 완성하세요.



- (1) 화면은 총 5개
- (2) 화면1에서는 아이디 입력
- (3) 화면2, 화면3에서 여러분께서 평소 하시고 싶었던 설문조사 구현
- (4) 화가, 음악가는 위한 예시일 뿐입니다. 똑같이 하시지 말고 개성 있게 구현해 보세요!!
- (5) 화면4에서는 감사의 멘트
- (6) 화면 5에서는 최종 결과 보여줌
- (7) 디자인은 자유 (단, 너무 무성의하지만 않으면 됩니다.)
- (8) SurveyController.java 별도 코딩
- (9) 컨트롤러 및 화면 5개 html 소스를 txt파일 하나에 넣어서 제출
- (10) 실행 동영상 짧게 녹화해서 제출