

로그인 페이지

로그인 구현하기

로그인을 구현하려고 한다. 일단 어떻게 동작하게 만들어야 하는지 생각해보자.

1. 사용자가 아이디, 비밀번호를 입력한다. (form input)
2. 입력된 데이터를 서버에 요청한다. (action="page.php")
3. 서버는 사용자가 요청한 데이터를 받는다. (\$_POST['name'])
4. 아이디, 비밀번호를 가지고 인증을 진행한다. (DB연결)
5. 응답을 보낸다. (echo)

필요한 것이 무엇인지 생각해보자.

1. 사용자가 아이디, 비밀번호를 입력하는 페이지.
2. 입력된 데이터를 처리하는 페이지.
3. 응답에 대한 결과 페이지.

사용자가 아이디, 비밀번호를 입력하는 페이지부터 만들어보자.

로그인 페이지

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>로그인</title>
</head>
<body>
  <form action="로그인을-처리하는-페이지.php" method="post">
    <label for="id">아이디:</label><input type="text" id="id" name="id"><br>
    <label for="password">비밀번호:</label><input type="password" id="password"
name="password"><br>
    <input type="submit" value="전송">
  </form>
</body>
</html>
```

위와 같이 구성하면, 사용자가 input에 데이터를 입력하고 전송버튼을 누르면, '로그인을-처리하는-페이지.php'로 브라우저는 서버에 요청을 보낸다. 이제, 로그인을 처리하는 페이지는 사용자가 요청한 데이터를 받아서 처리할 수 있다.

로그인을 처리하는 페이지

로그인을 처리하는 과정을 생각해보자.

1. DB에 연결한다.
 2. 사용자가 요청한 데이터를 가져온다.
 3. DB에서 아이디를 가지고 데이터를 가져온다.
- 만약 아이디가 존재하지 않으면 없는 아이디라고 응답을 보낸다.
4. 가져온 데이터의 비밀번호와 입력된 비밀번호를 확인한다.
- 만약 비밀번호가 맞지 않으면 틀린 비밀번호라고 응답을 보낸다.
5. 로그인 성공 처리를 진행한다.

우선 DB에 연결부터 해보자.

DB 연결하기

php에서 DB를 연결하는 함수는 `mysqli_connect()` 함수가 있다. 이 함수의 매개변수에는

- ip주소
- db아이디
- db패스워드
- db이름
- 포트번호

이렇게 들어간다. 그리고 함수의 리턴값으로 접속된 정보를 담은 객체가 리턴된다.

만일 접속에 실패하면, `false`값이 리턴되므로 이를 이용해 접속을 성공했는지 실패했는지 검사할 수 있다.

```
$link = mysqli_connect('127.0.0.1', 'root', 'password', 'mydb', '3306');

if (!$link) {
    echo '서버 접속 오류<br/>';
    echo mysqli_connect_error(); // 접속 에러 메시지 출력
    exit;
}
```

리턴된 `$link`를 가지고 DB에 쿼리를 날릴 수 있게 된다.

사용자가 요청한 데이터 가져오기

사용자가 `form`에 데이터를 입력하고 `submit`을 했을 테니까, 입력된 데이터들이 포함된 HTTP 요청이 왔을 것이다.

이를 php페이지에서 `$_GET` 혹은 `$_POST`라는 연관배열 변수로 받아올 수 있다.

위의 로그인 페이지에서 폼의 HTTP 메서드가 `post`방식이었으므로 php페이지에서 `$_POST`를 이용해 값을 받아올 수 있다.

아이디의 input name은 `id`였고, 비밀번호는 `password`였으므로 각각 `$_POST['id']`와 `$_POST['password']`로 얻어올 수 있다.

```
$id = $_POST['id'];
$password = $_POST['password'];
```

DB에서 데이터 가져오기

일단 사용자 `id`값으로 DB에서 데이터를 검색해본다. 만약 존재한다면, 일단 사용자 아이디는 존재하는 것이고 존재하지 않는다면 사용자 아이디가 존재하지 않는다는 오류 메시지를 응답으로 보내주면 될 것이다.

그럼 일단 쿼리문을 구성한다.

쿼리문은 `id`값을 조건으로 유저 테이블에서 검색을 하면 될 것이다.

```
$query = "SELECT * FROM `users` WHERE id = '{$id}'";

$result = mysqli_query($link, $query);

// 만약 쿼리문법에 오류가 있어서 에러가 발생하면, $result 변수에는 FALSE값이 들어간다.
if (empty($result)) {
    echo 'DB 오류<br>';
    echo mysqli_error($link); // 에러 메시지 출력
    exit;
}
```

`$result` 변수에는 DB에서 가져온 데이터가 들어있다. 그러면 이 데이터를 php 데이터에 맞게 가져오는 작업이 필요하다. 보통은 연관배열로 가져오므로, 연관배열 데이터로 받아와보자.

```
$data = mysqli_fetch_assoc($result);

// 만일 데이터가 존재하지 않으면 false값이 리턴된다.
if (empty($data)) {
    echo '존재하지 않는 아이디입니다.';
    exit;
}
```

데이터를 성공적으로 받아오면, 연관배열 형태로 받아왔을텐데 이 때 `key`값은 칼럼 이름이고, `value`값은 데이터 값이다. 따라서 비밀번호를 비교해준다.

```
if ($password !== $data['password']) {
    echo '비밀번호가 다릅니다.';
    exit;
}
```

위의 조건도 통과하면 아이디도 존재하며 비밀번호도 같으므로, 로그인을 진행하면 된다.

로그인을 성공하면, 세션에 로그인 성공에 대한 값을 넣어주면 된다.

```
$_SESSION['login'] = true;
```

이제 앞으로, 해당 유저는 어떤 php 페이지에서든 세션에 'login'값이 true이면 로그인이 되어있음을 알 수 있게 된다.

그리고 이제 로그인 성공을 했으니, 메인페이지로 리다이렉트를 시켜주면 된다.

```
header('Location: /index.php');
```

- header 내장함수 기능 알아보기
-

과제1

앞서 배운 내용을 토대로 로그인페이지 로직을 짜시오.

- 로그인 후 메인페이지로 리다이렉트 하면 된다.
- 세션을 기능을 추가할 것.

과제2

로그아웃 기능 만들기

- 세션을 파괴하는 페이지를 만들면 된다.
- 세션을 파괴하고 메인 페이지로 리다이렉트시키시오.