

Web Hacking

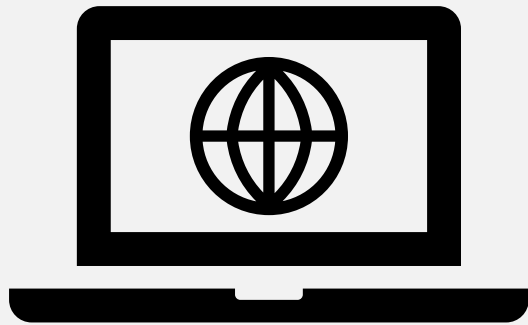
CONTENTS

Contents 1

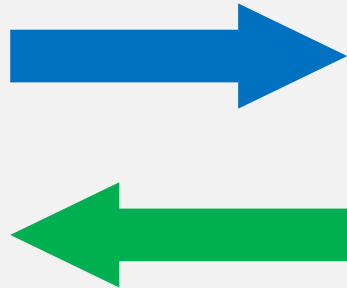
- bitnami WAMP
- html, php
- 홈페이지 구축

Contents 1
홈페이지 구축

Apache

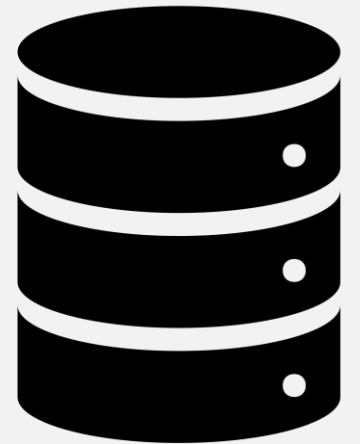
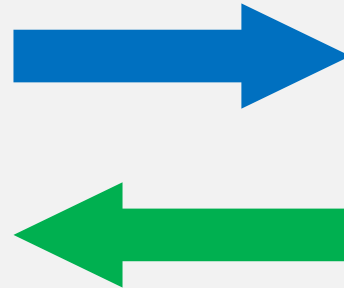


클라이언트



서버

Apache



데이터베이스

MySQL

Contents 1

홈페이지 구축

웹 서버 루트

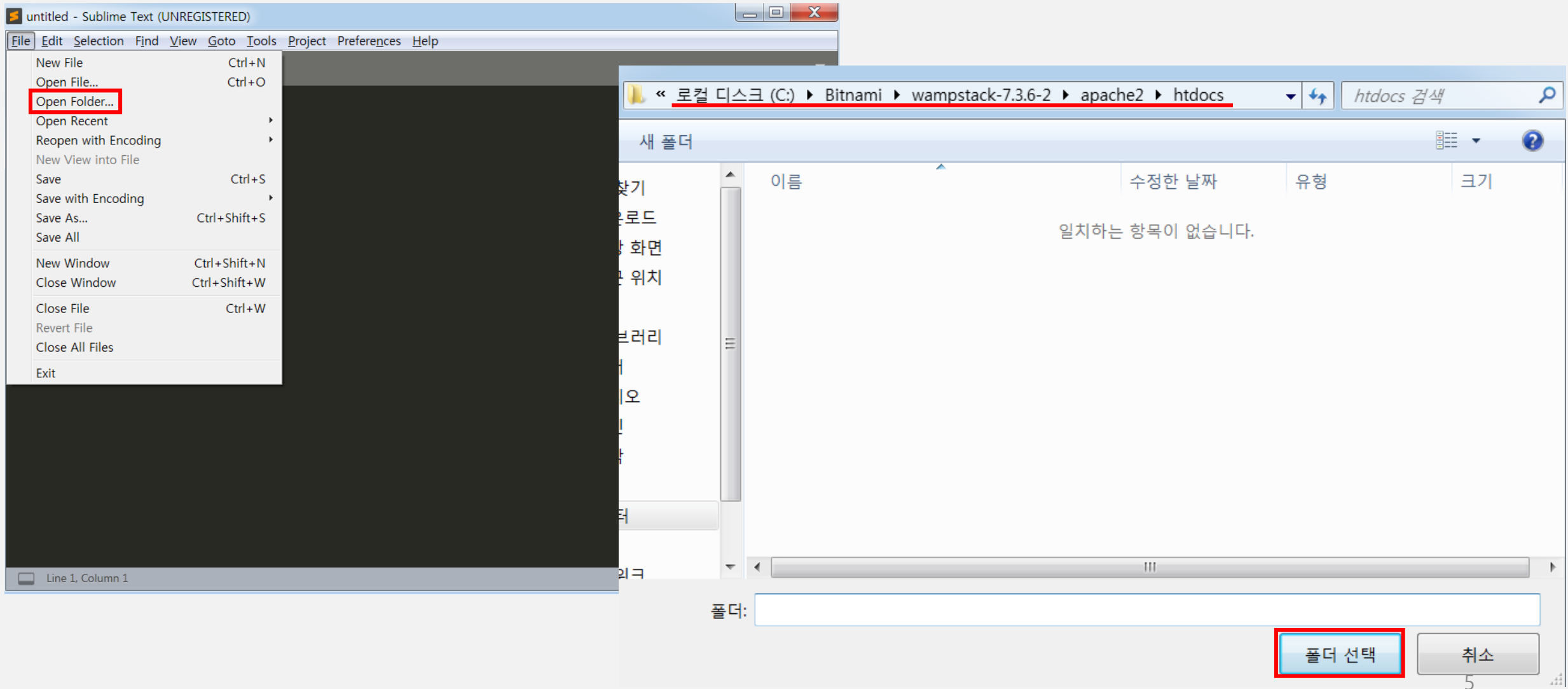
로컬 디스크 (C:) > Bitnami > wampstack-7.3.6-2 > apache2 > htdocs					htdocs 검색	
이름	수정한 날짜	유형	크기			
process	2019-07-08 오전...	파일 폴더				
sample	2019-07-12 오후...	파일 폴더				
template	2019-07-08 오전...	파일 폴더				
board	2019-07-08 오전...	Chrome HTML D...	2KB			
index	2019-07-12 오후...	PHP 파일	1KB			
loginForm	2019-07-08 오전...	Chrome HTML D...	1KB			
modify	2019-07-12 오후...	Chrome HTML D...	2KB			
signup	2019-07-05 오전...	Chrome HTML D...	1KB			
view	2019-07-12 오후...	Chrome HTML D...	2KB			
write	2019-07-08 오전...	Chrome HTML D...	1KB			

C:\Bitnami\wampstack-7.3.6-2\apache2\htdocs가 기본적으로 Bitnami에서 제공하는 웹 서버의 루트

Contents 1

홈페이지 구축

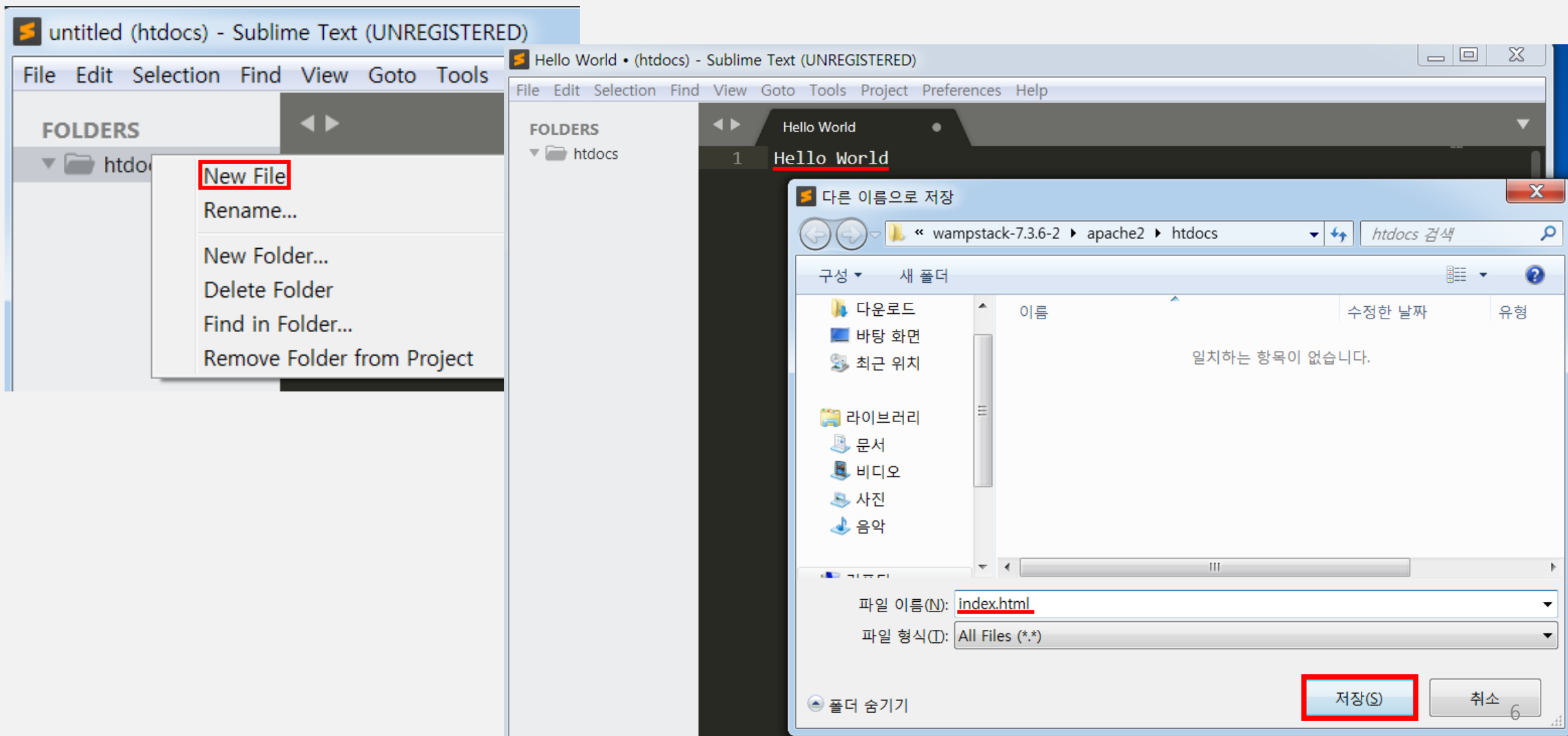
경로 설정



Contents 1

홈페이지 구축

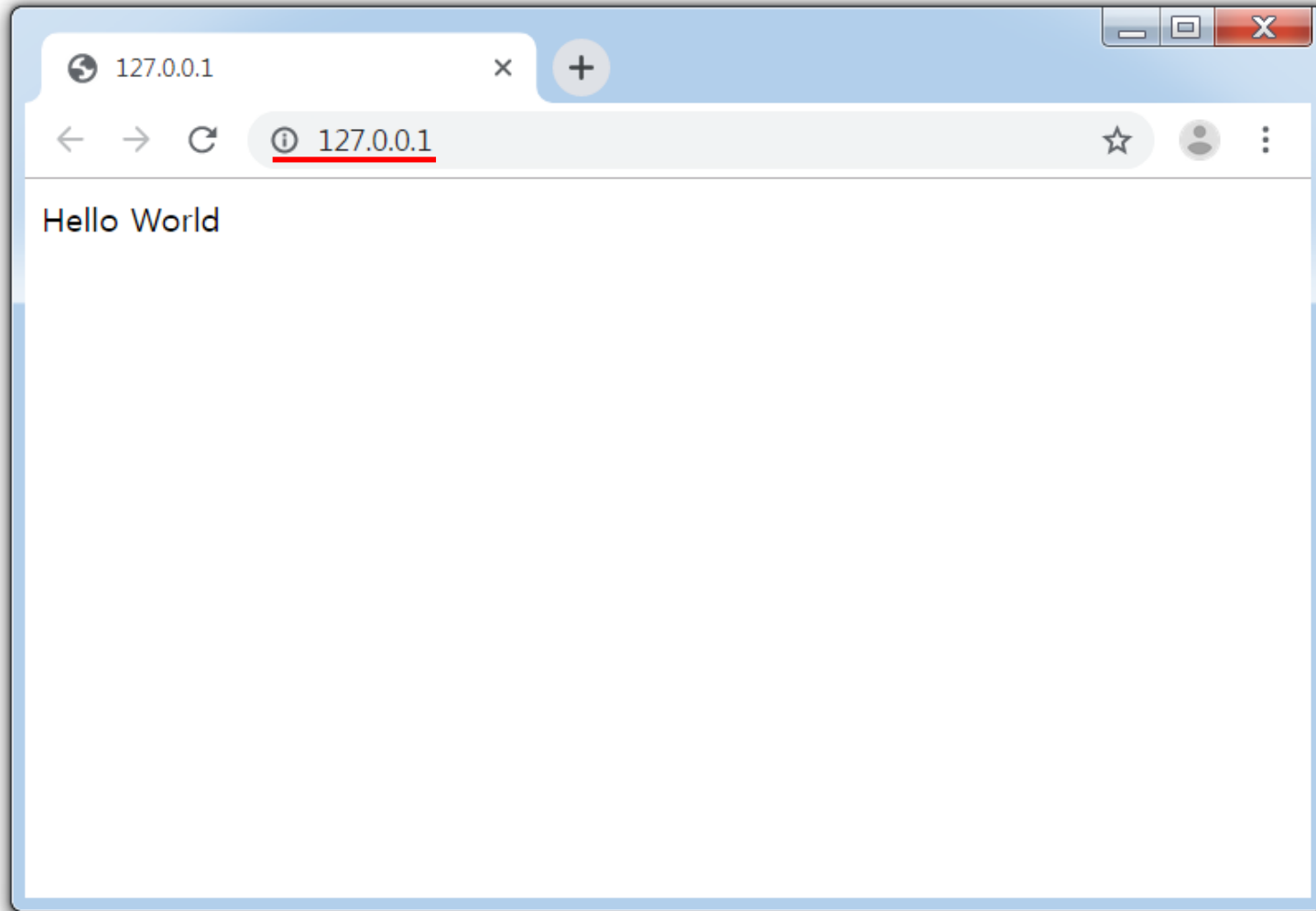
경로 설정



Contents 1

홈페이지 구축

Chrome 127.0.0.1 접속 (local)

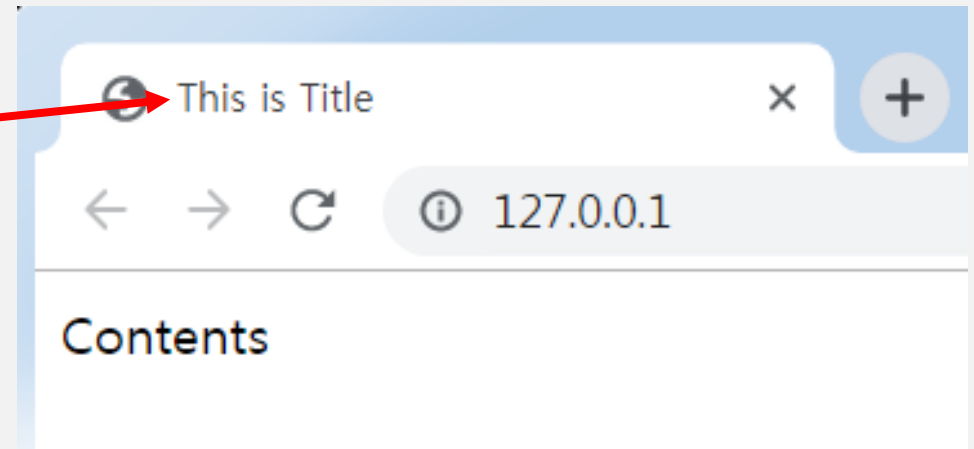


```
<!DOCTYPE html>
<html>
<!-- created 2010-01-01 -->
<head>
  <title>sample</title>
</head>
<body>
  <p>Voluptatem accusantium
    totam rem aperiam.</p>
</body>
</html>
```

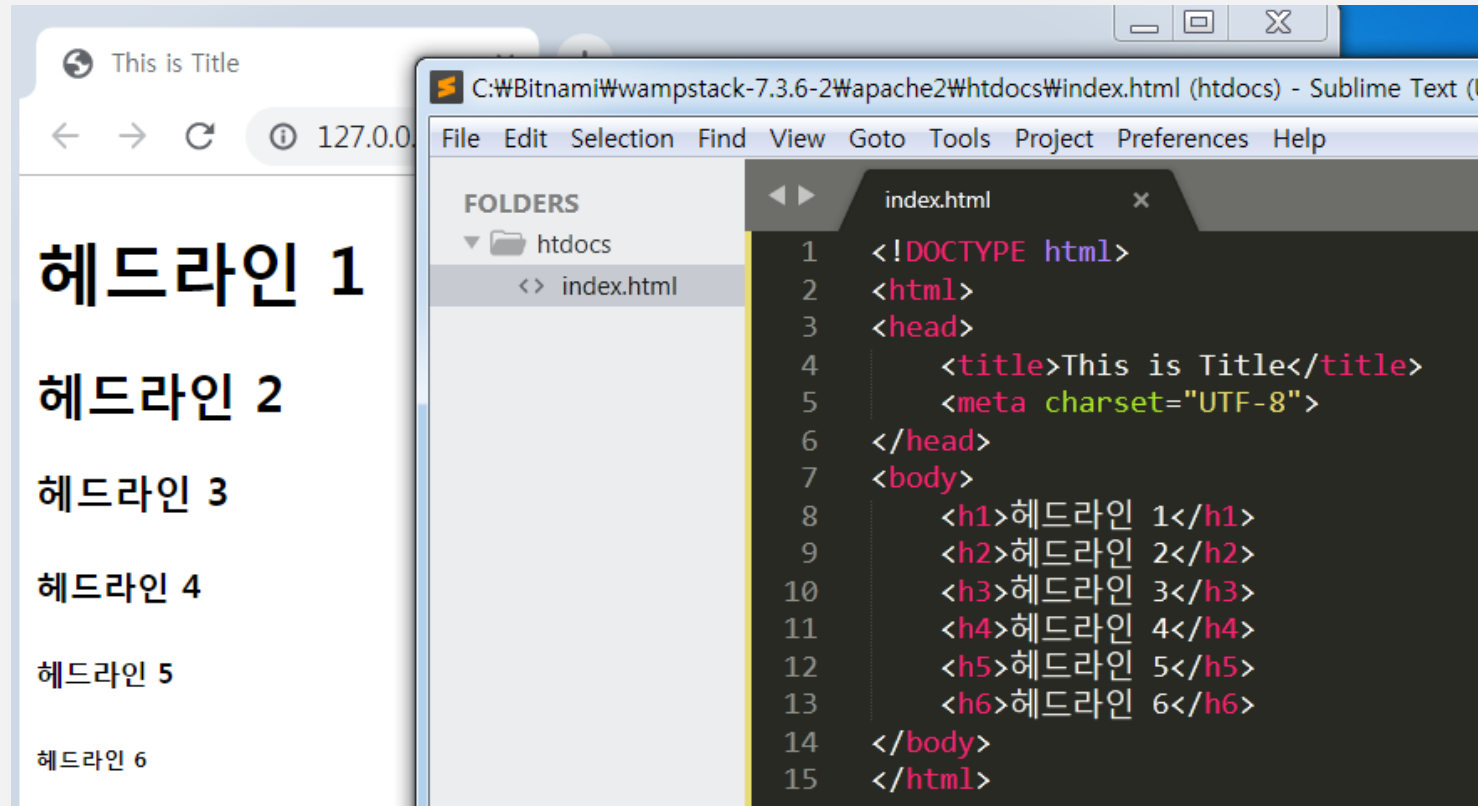
HTML

- HyperText Markup Language
- 태그와 콘텐츠로 구성
- 정적인 콘텐츠(이미지, 텍스트)를 표현


```
index.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>This is Title</title>
5 </head>
6 <body>
7   Contents
8 </body>
9 </html>
```



- `<!DOCTYPE html>`은 해당 문서의 유형이 무엇인지와 버전을 알려주기 위해 사용
- `<html>`은 html 문서의 처음과 끝을 나타내기 위하여 사용하는 태그
- `<head>`는 문서의 제목, 문서가 사용할 Java script 및 CSS 등 문서의 속성을 나타내는 태그
- `<body>`는 실질적인 콘텐츠가 들어가는 태그

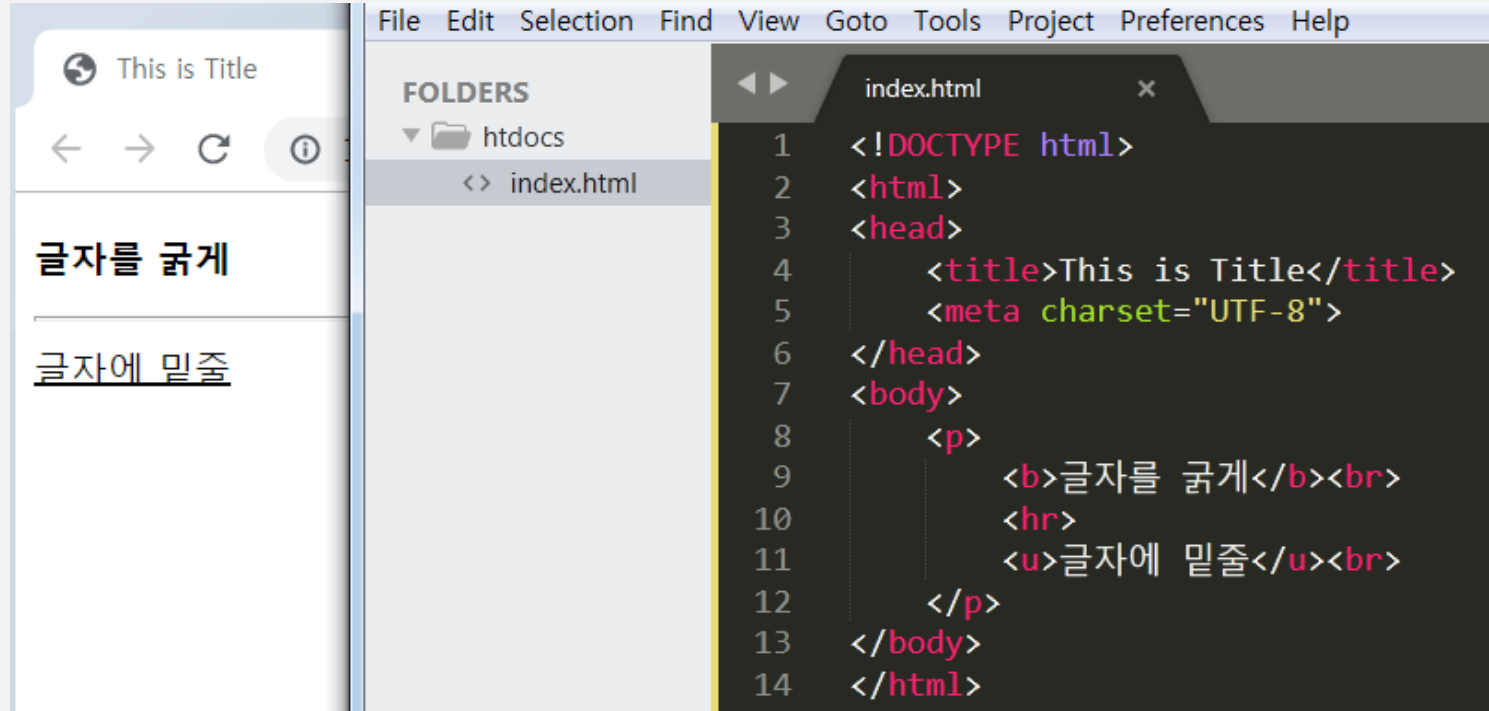


- `<h1>` 태그는 문서의 헤드라인을 나타내는 태그이다. h1 ~ h6까지 존재하며 숫자가 클수록 글자가 작음
- `<meta charset="UTF-8">`은 UTF-8로 인코딩하여 한글을 표현하기 위해 사용

Contents 1

홈페이지 구축

HTML

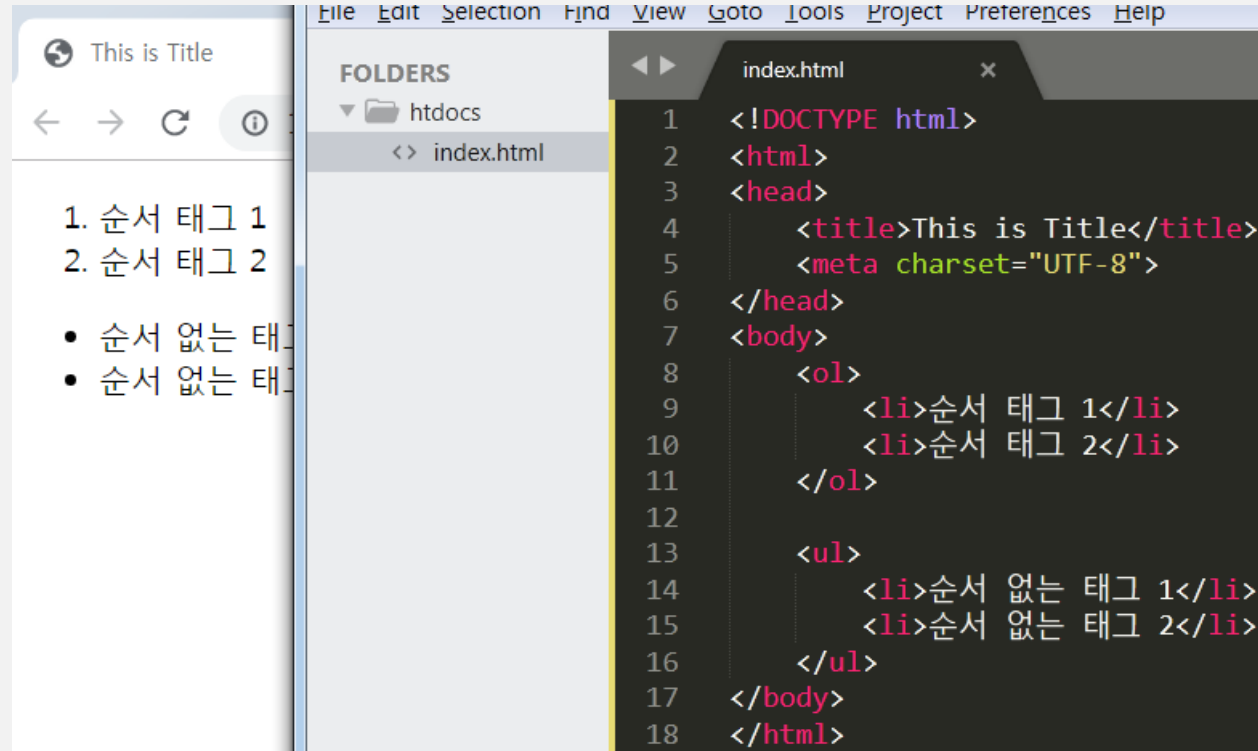


- <p> 태그는 문단을 나타내는 태그
- 태그 사이에 입력한 텍스트는 굵게 나타냄
- <u> 태그 사이에 입력한 텍스트는 밑줄을 그음
-
 태그는 줄바꿈을 나타냄
- <hr> 태그는 수평선을 그림

Contents 1

홈페이지 구축

HTML



The screenshot shows a web browser on the left and an HTML editor on the right. The browser's address bar displays 'This is Title'. The editor's 'FOLDERS' pane shows a project structure with 'htdocs' and 'index.html'. The editor window displays the following HTML code:

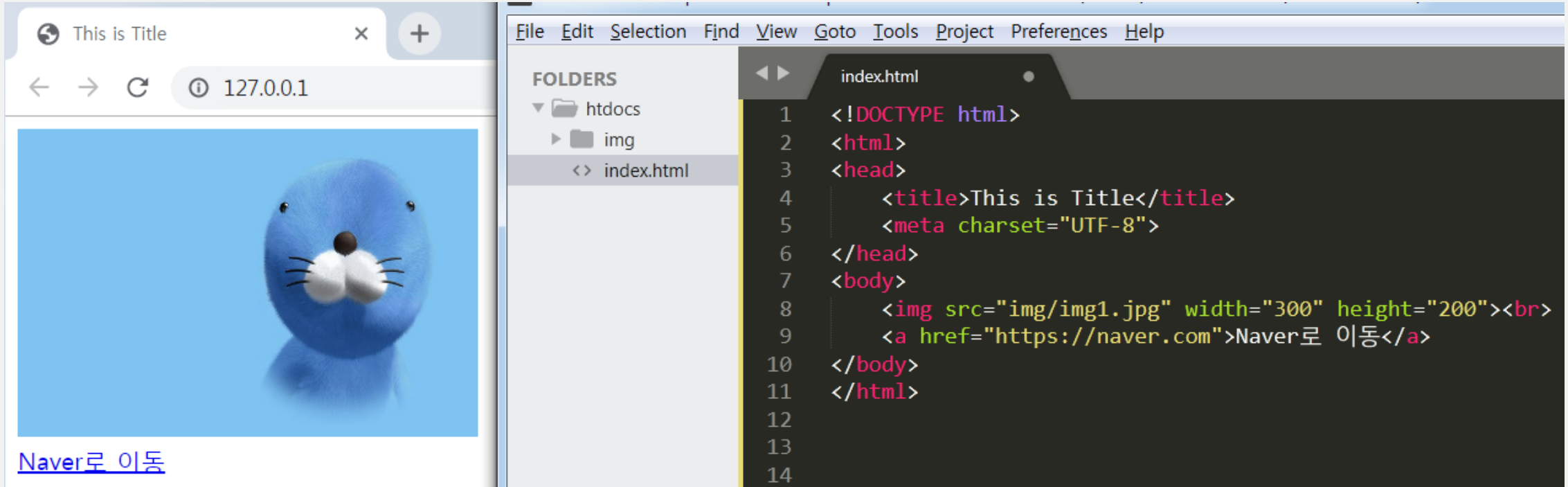
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>This is Title</title>
5   <meta charset="UTF-8">
6 </head>
7 <body>
8   <ol>
9     <li>순서 태그 1</li>
10    <li>순서 태그 2</li>
11  </ol>
12
13  <ul>
14    <li>순서 없는 태그 1</li>
15    <li>순서 없는 태그 2</li>
16  </ul>
17 </body>
18 </html>
```

- 은 순서가 있는 목록을 의미
- 은 순서가 없는 목록을 의미
- 는 목록의 요소를 의미

Contents 1

홈페이지 구축

HTML

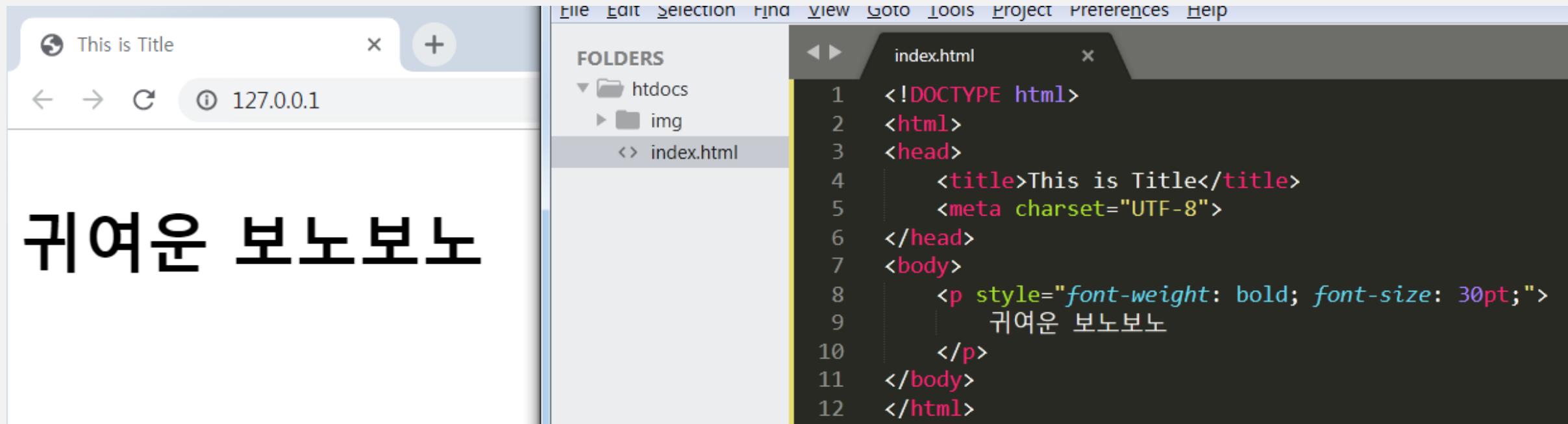


- `` 태그는 사진을 불러올 수 있다. `src`로 경로를 지정하고 `width`와 `height`로 크기를 조절
- `<a>` 태그는 하이퍼 링크 태그이다. 클릭 시 `href` 내에 명시한 URL로 이동

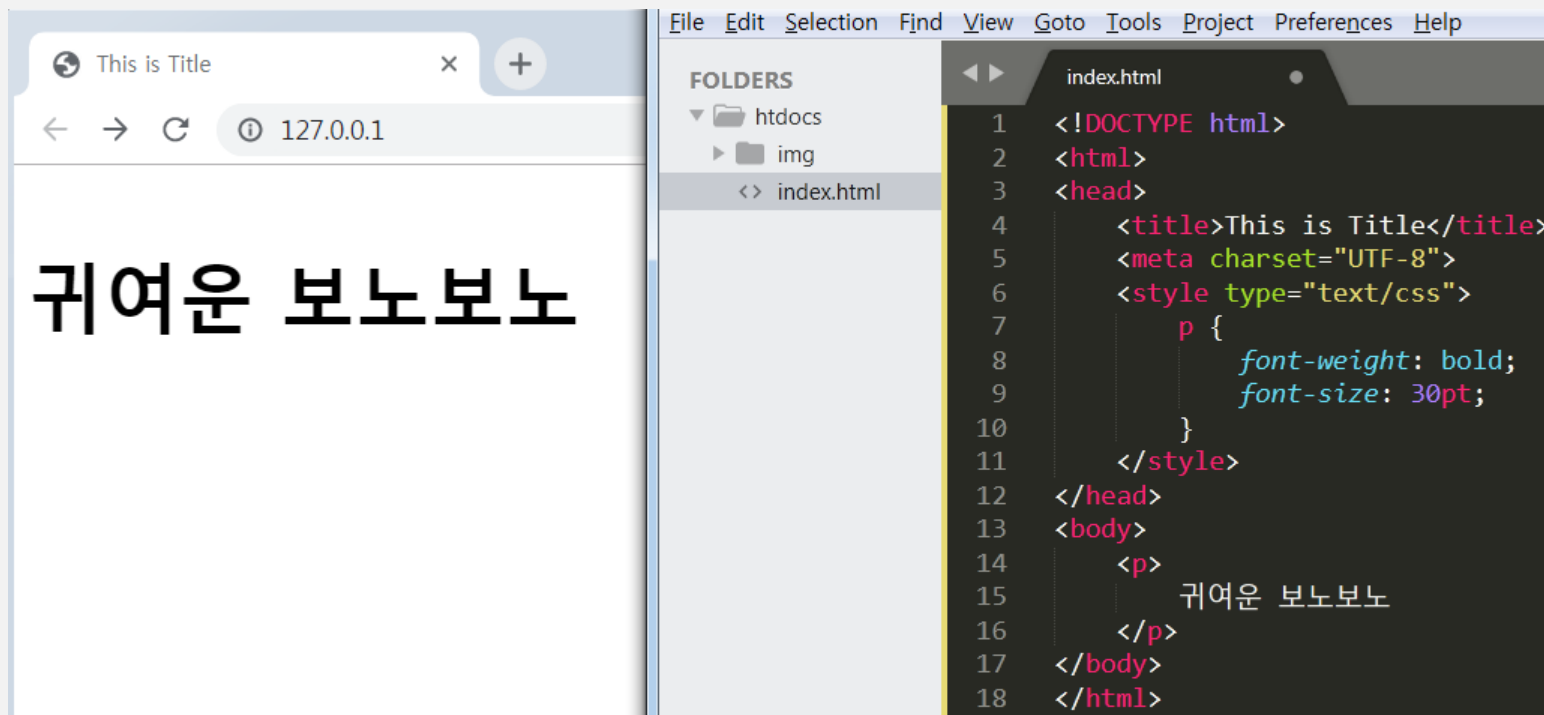
CSS



- Cascading Style Sheets
- 스타일 시트 언어
- 사이트의 전체 스타일을 손쉽게 제어



- style을 이용하여 글자 크기, 글자 굵기, 글자 색 등 다양한 스타일을 적용시킬 수 있음
- 위의 예시와 마찬가지로 원하는 태그 내에 직접 삽입하는 것도 가능

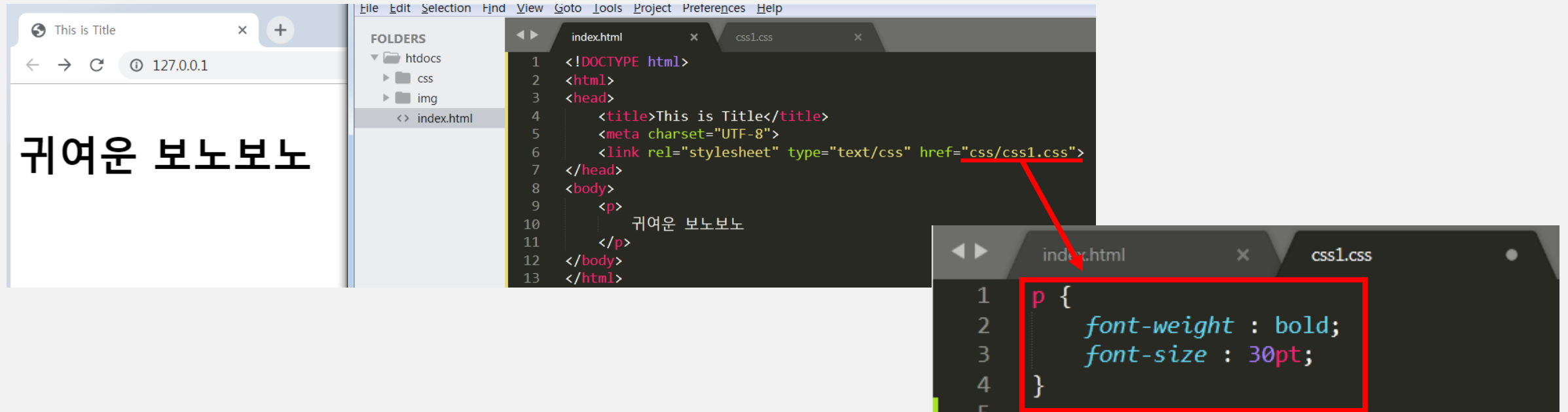


- style을 태그 내에 직접 삽입하는 것이 아니라 head 태그 내에 삽입하는 것도 가능
- 태그, id, class 등을 지정하여 스타일을 적용시킬 수 있음

Contents 1

홈페이지 구축

CSS



- style을 css파일에 쓴 뒤 해당 css파일을 html에서 불러와도 동일한 효과를 볼 수 있음

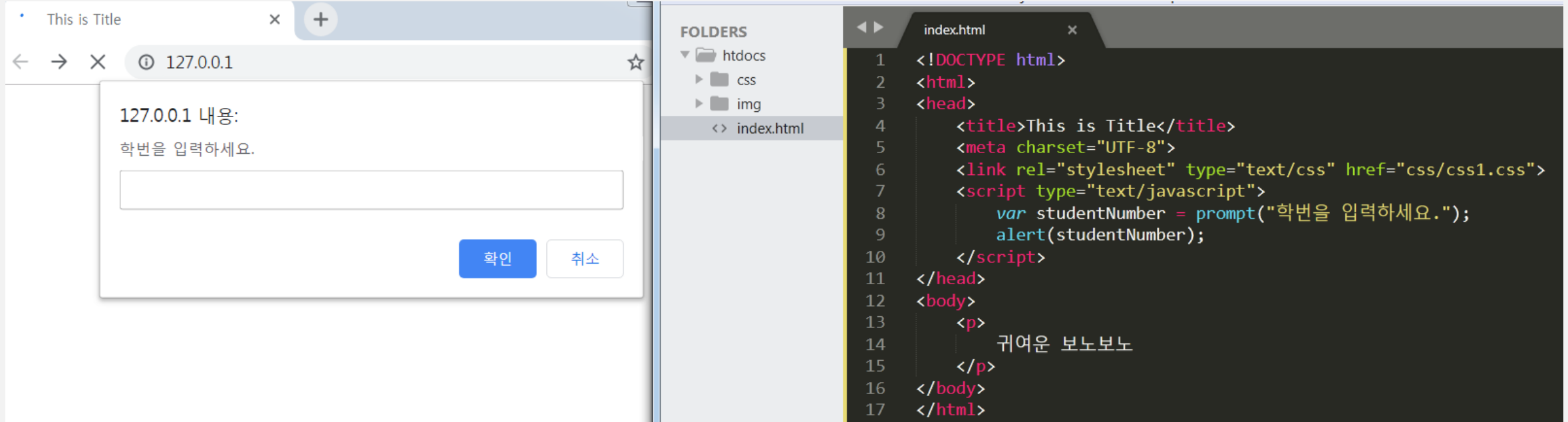


- 객체 지향형 스크립트 언어
- HTML에서 <script> 태그를 이용하여 삽입
- 웹을 동적으로 만들어 줌

Contents 1

홈페이지 구축

Java Script

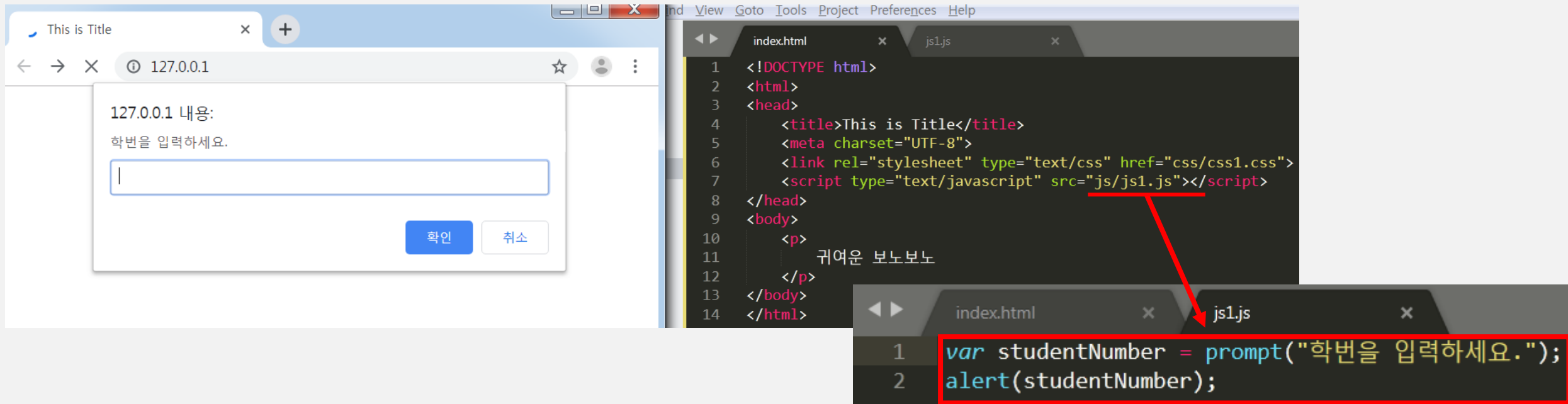


- `<script>` 태그 내에 코드를 작성하여 실행할 수 있음

Contents 1

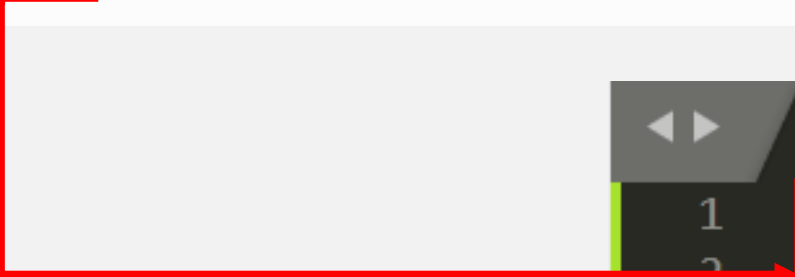
홈페이지 구축

Java Script



- script 코드를 js 파일에 작성하고 해당 js파일을 불러와서 사용할 수도 있음

css	2019-06-30 오전 2:57	파일 폴더	
img	2019-06-30 오전 1:23	파일 폴더	
js	2019-06-30 오전 4:00	파일 폴더	
html1	2019-06-30 오전 4:09	Chrome HTML Do...	1KB
<u>index.php</u>	2019-06-30 오전 4:10	PHP 파일	1KB

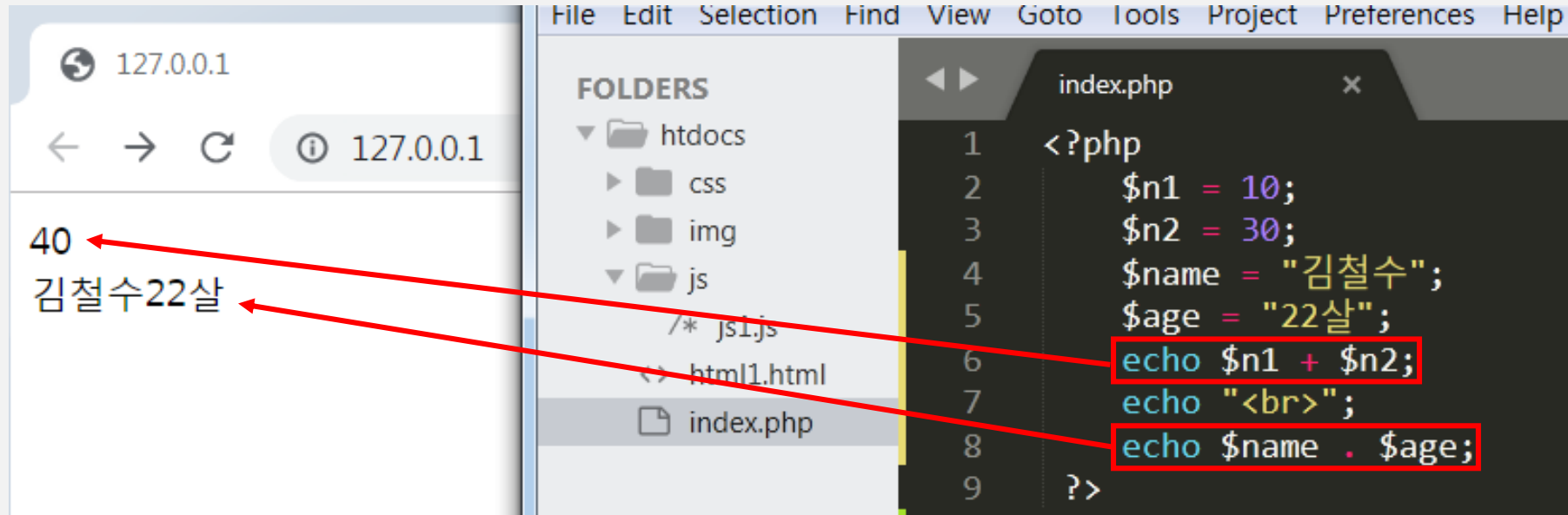


```
index.php x
1 <?php
2     echo "귀여운 보노보노";
3 ?>
```

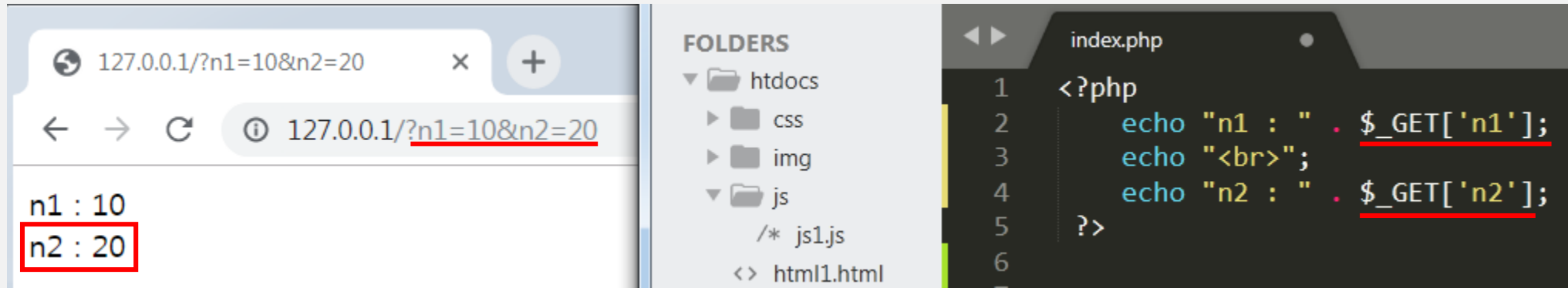
- 지금까지 실습한 index.html파일을 html1.html로 이름을 변경하고 index.php라는 파일을 새로 만들어서 위와 같은 코드를 작성한 후 웹 페이지에서 확인



- Hypertext Preprocessor (Personal Home Page)
- HTML에 포함되어 동작하는 스크립팅 언어
- 별도의 실행 파일 없이 HTML 문서 안에 포함
- 동적인 웹 문서를 빠르고 쉽게 작성



- php 코드는 **<?php** 로 시작하며 **?>** 로 끝냄
- 변수 이름 앞에 **\$**를 붙임
- 문자열을 붙일 때는 **.**을 이용하여 붙임

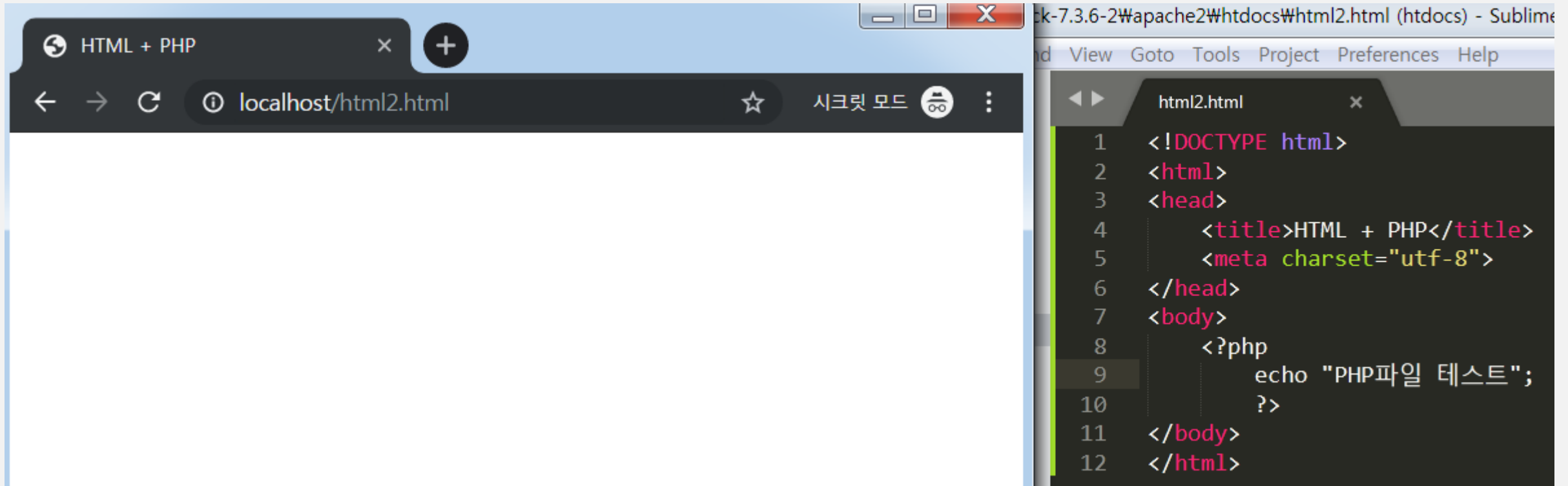


- PHP에서 GET Method의 인자는 **`$_GET['인자 이름']`**로 가져올 수 있음
- N1과 n2가 인자 이름이고 `$_GET['n1']`을 통해 n1의 값을 가져오며 `$_GET['n2']`를 통해 n2의 값을 가져옴
- POST Method의 인자는 **`$_POST['인자 이름']`**을 이용하여 가져올 수 있음

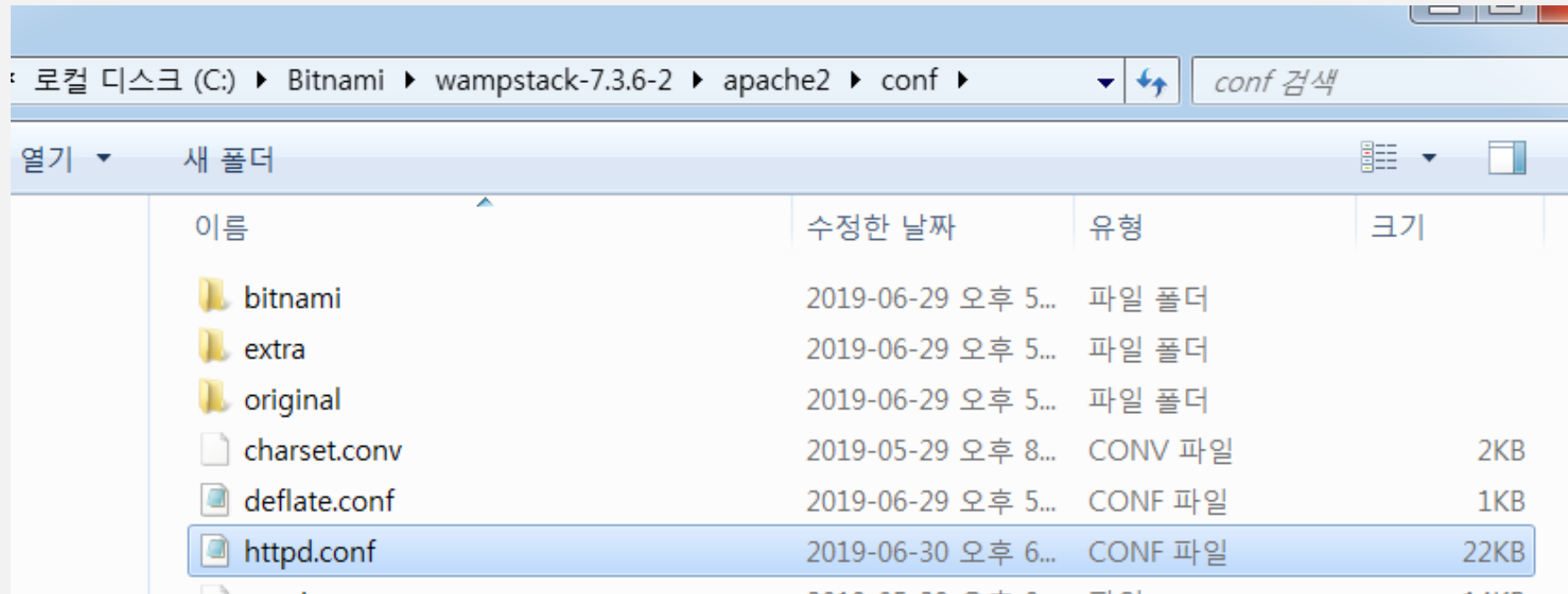
Contents 1

홈페이지 구축

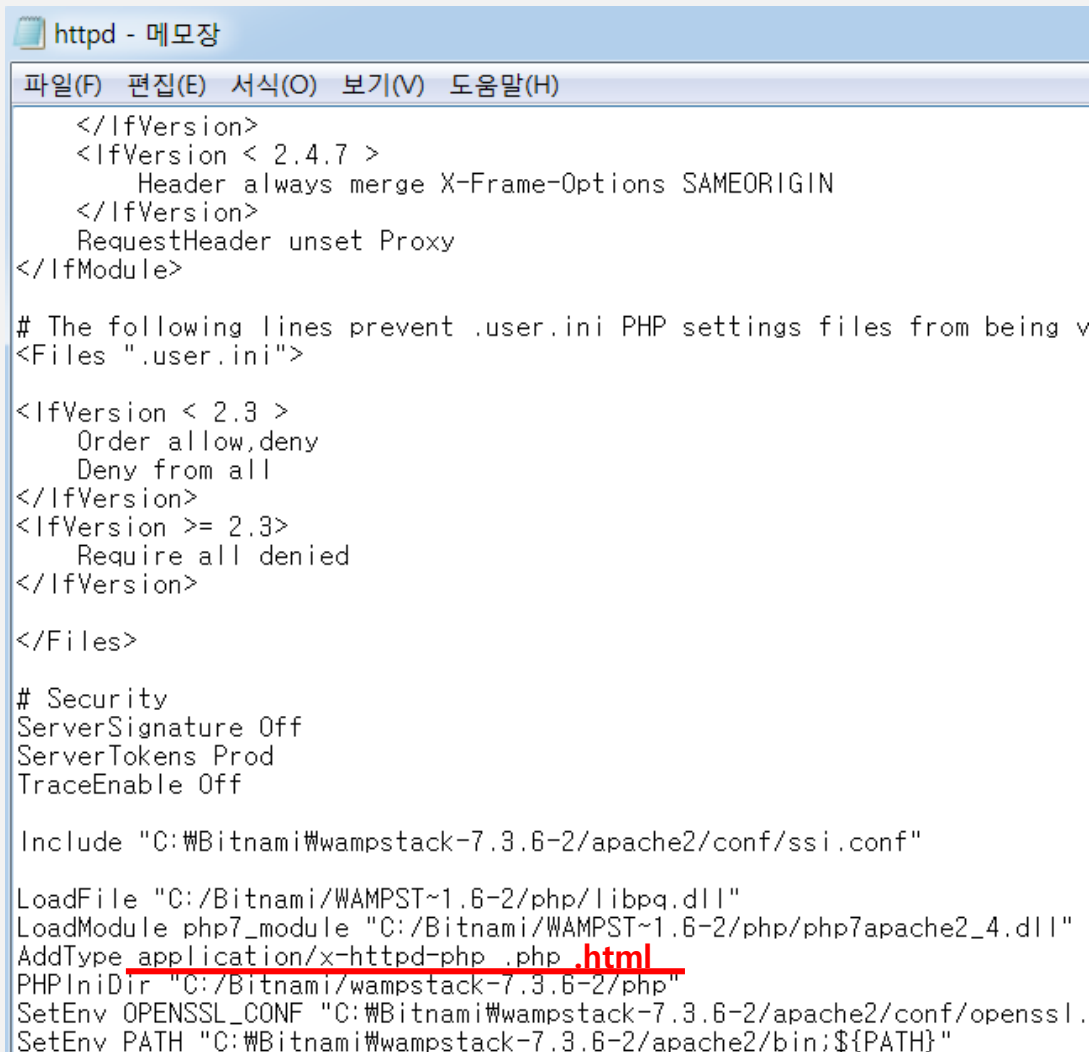
PHP



- html 코드 내에 php 코드를 삽입했으나 실행되지 않음
- Apache 웹 서버 설정에서 이를 해결해야함



- C:\Bitnami\wampstack-7.3.6-2\apache2\conf에 위치한 httpd.conf는 Apache 웹 서버에 대한 설정을 저장하고 있는 파일
- 해당 파일에서 php를 실행할 수 있는 확장자를 설정할 수 있음



The screenshot shows a text editor window titled 'httpd - 메모장' (httpd - Notepad). The menu bar includes '파일(F)', '편집(E)', '서식(O)', '보기(V)', and '도움말(H)'. The main text area contains an Apache configuration snippet. It starts with an `</IfVersion>` block for versions less than 2.4.7, setting the `Header` to `always merge X-Frame-Options SAMEORIGIN`. This is followed by an `</IfVersion>` block for versions 2.4.7 and above, where `RequestHeader unset Proxy` is set. Then, there's a `</IfModule>` block. Below this, a comment states that the following lines prevent .user.ini PHP settings files from being loaded. This is followed by another `</IfVersion>` block for versions less than 2.3, setting `Order` to `allow,deny` and `Deny` to `from all`. For versions 2.3 and above, `Require` is set to `all denied`. The `</Files>` block ends. A `# Security` section follows, with `ServerSignature Off`, `ServerTokens Prod`, and `TraceEnable Off`. Then, `Include` is used to load `"C:\wampstack-7.3.6-2\apache2\conf\ssi.conf"`. The `LoadFile` directive loads `"C:/Bitnami/WAMPST~1.6-2/php/libpq.dll"`. The `LoadModule` directive loads `"C:/Bitnami/WAMPST~1.6-2/php/php7apache2_4.dll"`. The `AddType` directive is used to add `application/x-httpd-php.php.html` to the `text/html` MIME type. The `PHPIniDir` is set to `"C:/Bitnami/wampstack-7.3.6-2/php"`. The `SetEnv` directive sets `OPENSSL_CONF` to `"C:\wampstack-7.3.6-2\apache2\conf\openssl.conf"`. Finally, the `SetEnv` directive sets `PATH` to `"C:\wampstack-7.3.6-2\apache2\bin;%PATH%"`.

- **AddType application/x-httpd-php뒤에 명시한 확장자에서 php 코드를 읽고 실행하게 됨**
- **따라서 현재는 확장자가 php인 파일만 php 코드가 실행이 되는 것임**
- **.php 뒤에 .html을 적어주면 html 문서에서도 php 코드를 실행할 수 있음**
- **변경된 설정을 적용하기 위해 Apache 웹 서버를 재시작 해주어야 함**

Contents 1

홈페이지 구축

PHP

The image shows a Windows file explorer window on the left and the Bitnami WAMP Stack 7.3.6-2 control panel on the right.

File Explorer: The address bar shows the path: 컴퓨터 > 로컬 디스크 (C:) > Bitnami > wampstack-7.3.6-2. The file list includes folders like 'letsencrypt', 'licenses', 'mysql', 'php', 'scripts', 'sqlite' and files like 'changelog.txt', 'manager-windows.exe' (highlighted with a red line), 'properties.ini', and 'README.txt'.

Bitnami WAMP Stack 7.3.6-2: The 'Manage Servers' tab is selected. It displays a table of server statuses:

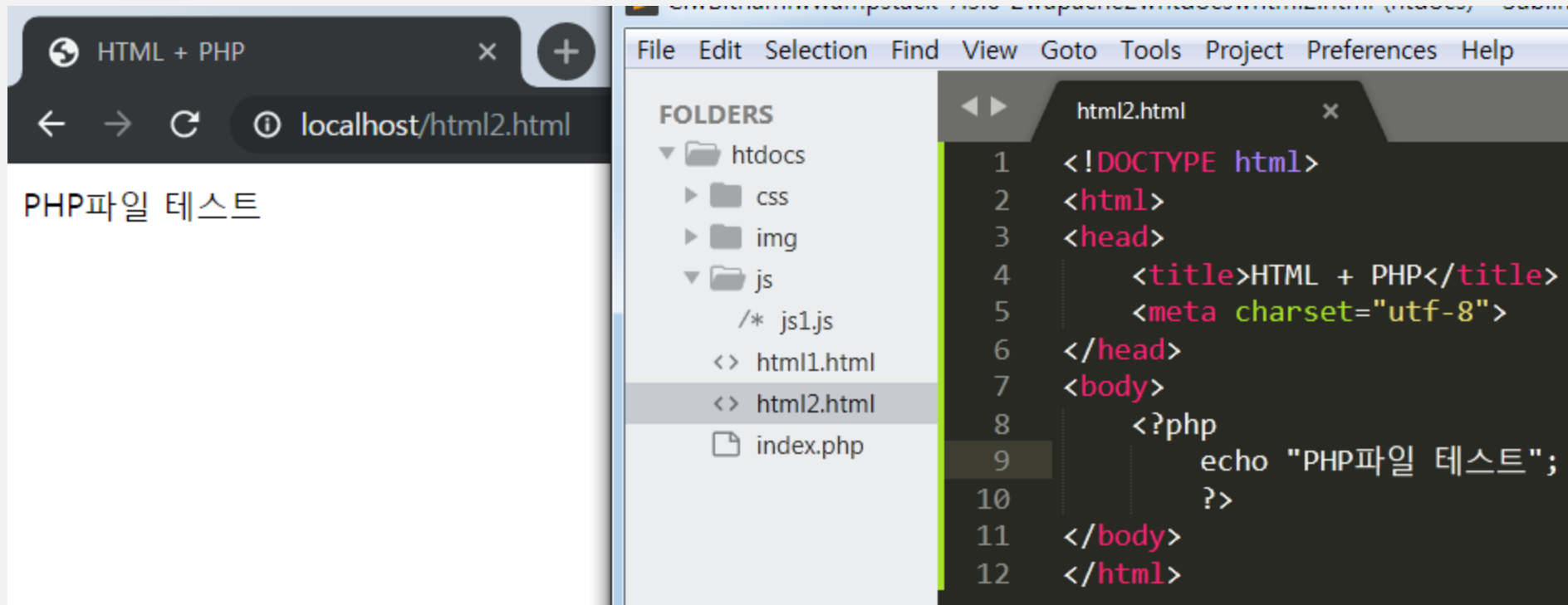
Server	Status
MySQL Database	Running
Apache Web Server	Running

The 'Apache Web Server' row is highlighted with a blue background and a red border. To the right of the table are buttons for 'Start', 'Stop', 'Restart' (highlighted with a red border), and 'Configure'. At the bottom of the window are buttons for 'Start All', 'Stop All', and 'Restart All'.

Contents 1

홈페이지 구축

PHP

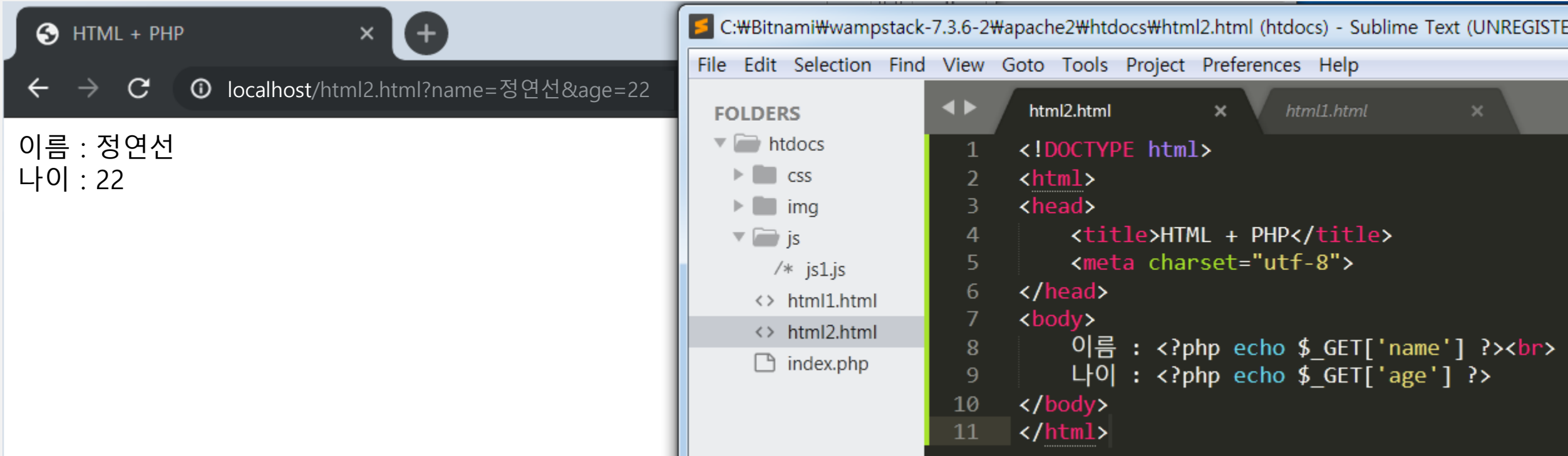


- AddType application/x-httpd-php .php .html 이라고 설정을 변경한 뒤 Apache를 재 시작하면 HTML 문서 내에 있는 PHP 코드가 잘 실행되는 것을 볼 수 있음

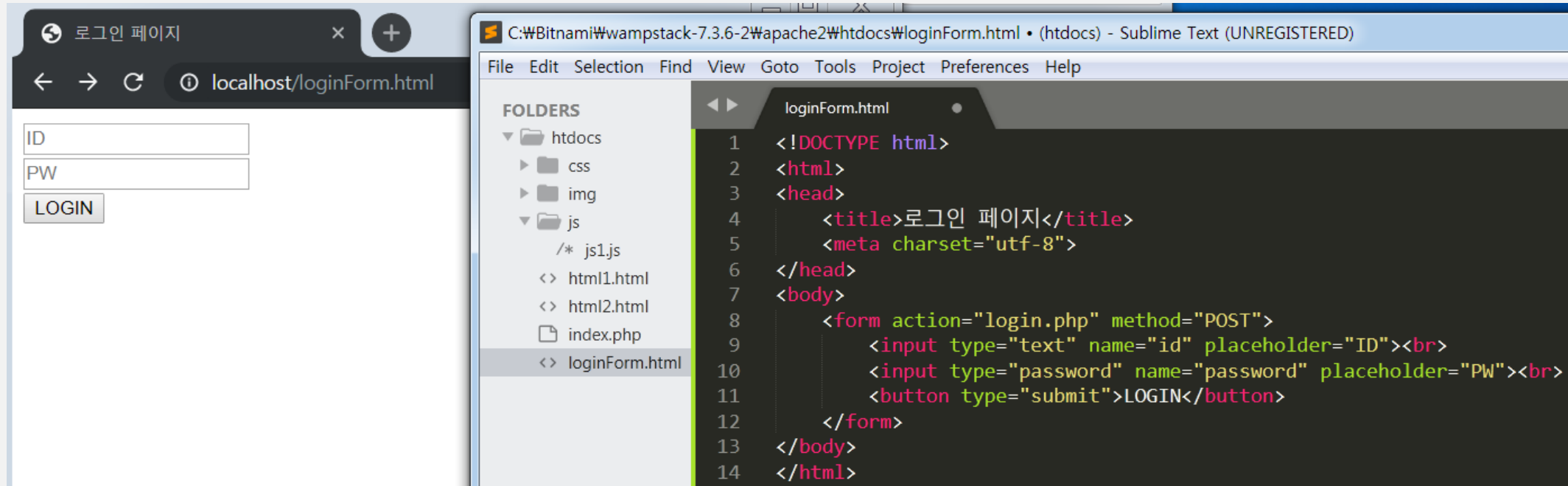
Contents 1

홈페이지 구축

PHP



- 정적인 HTML 문서에 PHP를 더하여 동적인 페이지를 만들 수 있게 됨
- 사용자의 입력 값을 **PHP에서 처리하여 HTML에 더해 웹 페이지를 완성**
- 그리고 Client가 그것을 확인하는 구조로 Client와 Server가 통신

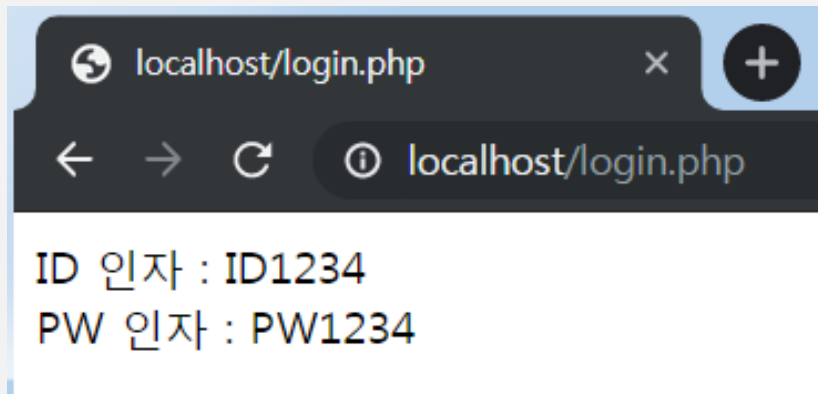
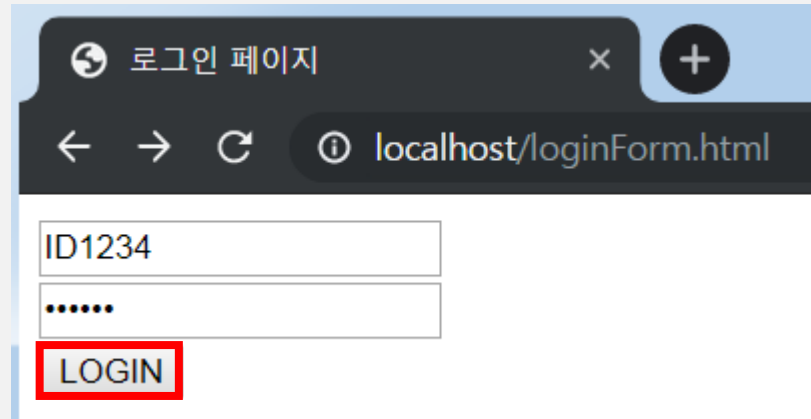


- <form> 태그를 이용하여 사용자에게 입력을 받고 그 값을 login.php로 넘겨서 로그인을 진행
- action에 적은 경로로 값이 전달되고 method는 GET이나 POST 같이 login.php로 인자를 전달할 때 사용할 방식을 의미
- <input> 태그는 입력을 받을 수 있는 영역을 만든다. name에 입력한 값이 form의 action에 적은 경로에 전달되는 인자 명이 됨

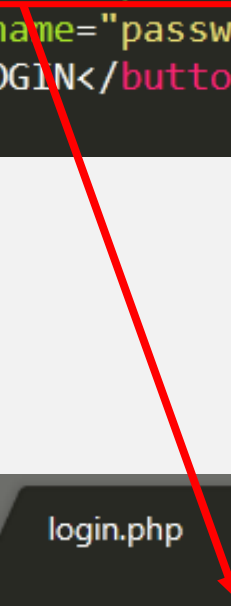
Contents 1

홈페이지 구축

Login Form



```
<form action="login.php" method="POST">
  <input type="text" name="id" placeholder="ID"><br>
  <input type="password" name="password" placeholder="PW"><br>
  <button type="submit">LOGIN</button>
</form>
```

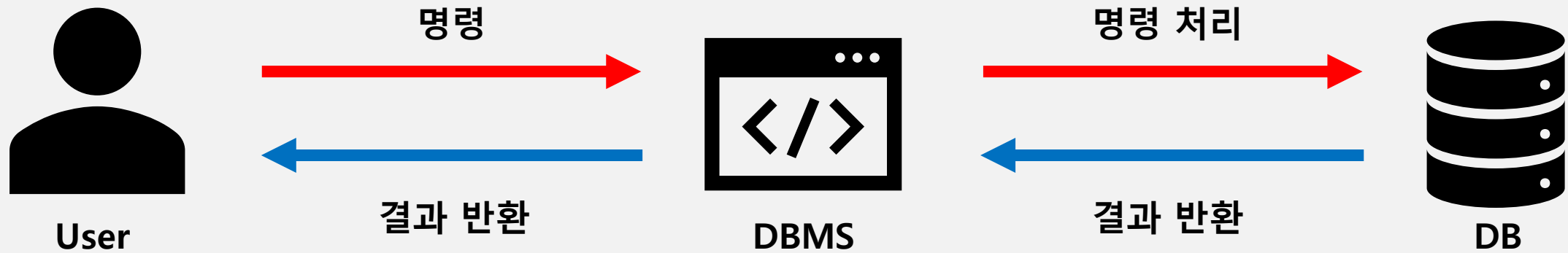


```
<?php
  echo "ID 인자 : " . $_POST['id'] . "<br>";
  echo "PW 인자 : " . $_POST['password'] . "<br>";
?>
```

- 이제 login.php에서 id와 password 인자 값을 이용하여 Database에서 로그인을 진행

- 연관된 데이터들의 집합
- 추가 / 삭제 등으로 인한 변동
- 수 많은 데이터를 효율적으로 관리
- 사이트 내에서 회원 관리 / 게임 정보 관리





- 데이터베이스 관리 시스템

데이터베이스를 제어하는 소프트웨어

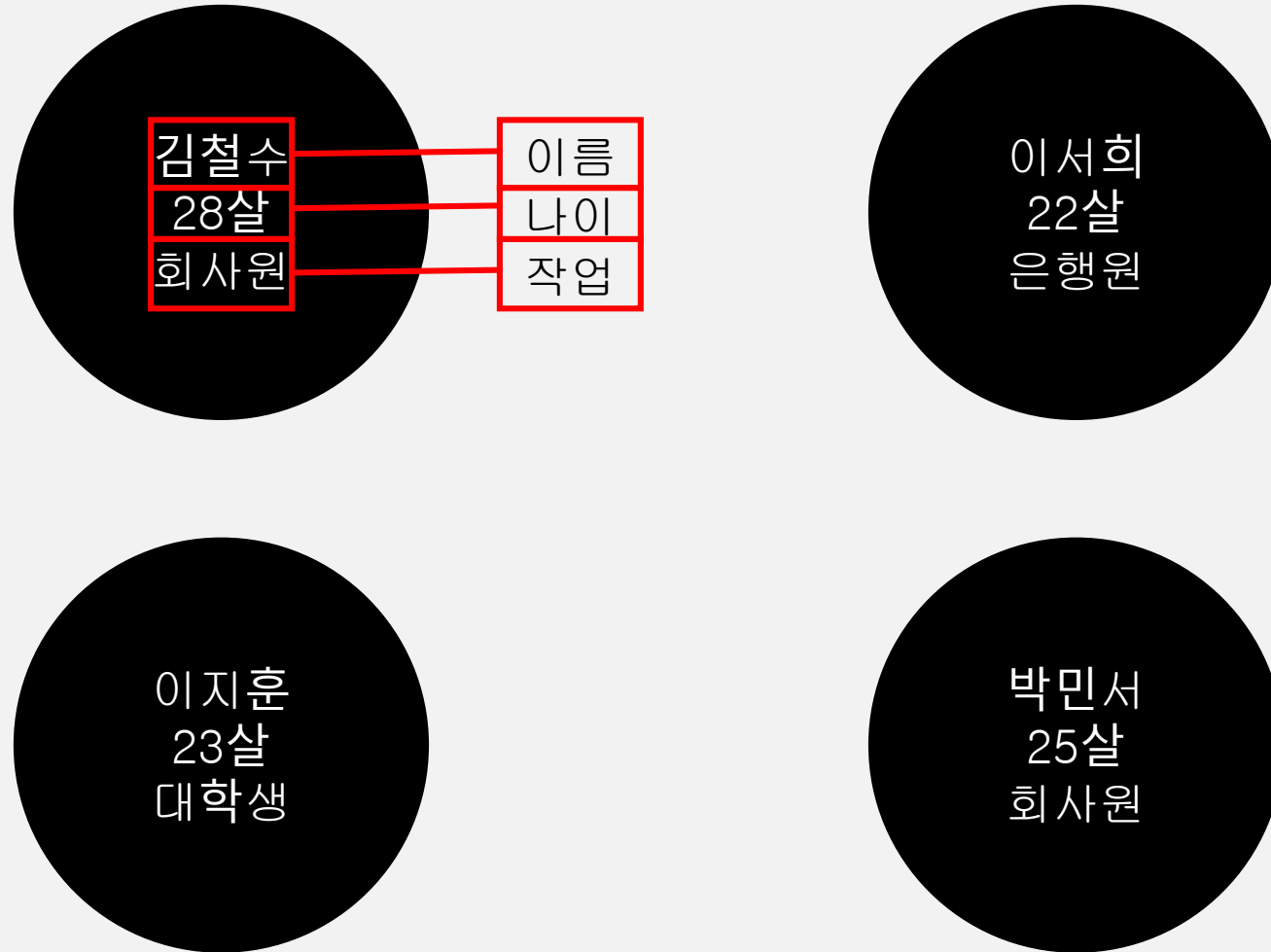
- 사용자와 데이터베이스 사이에 존재

사용자의 명령에 따라 데이터 생성, 삭제, 수정, 조회 등을 처리

Contents 1

홈페이지 구축

Database (DB)



테이블 (Table)

이름	나이	직업
김철수	28	회사원
이서희	22	은행원
이지훈	23	대학생
박민서	25	회사원

행 (Row / Record)

이름	나이	직업
김철수	28	회사원
이서희	22	은행원
이지훈	23	대학생
박민서	25	회사원

열 (Column / Field)

이름	나이	직업
김철수	28	회사원
이서희	22	은행원
이지훈	23	대학생
박민서	25	회사원

Database

Table

Table

Table

- **C, Python 같은 언어의 일종**
데이터베이스에 접근하는 언어
- **SQL을 기초로 하는 여러 DBMS가 존재**
예) MySQL, SQLite, MSSQL 등
- **관계형 데이터베이스를 다루는 언어**



데이터 정의 언어 (DDL)

문법	기능
CREATE	테이블, 데이터베이스 등을 생성
DROP	테이블, 데이터베이스 등을 삭제
ALTER	기존의 테이블, 데이터베이스 등을 수정

데이터 조작 언어 (DML)

문법	기능
SELECT	데이터 조회
INSERT	데이터 삽입
UPDATE	데이터 수정
DELETE	데이터 삭제

```
C:\>cd C:\Bitnami\wampstack-7.3.6-2\mysql\bin MySQL이 설치된 경로로 이동한다.  
C:\Bitnami\wampstack-7.3.6-2\mysql\bin>mysql -u root -p WAMP 설치 시 입력한 root 계정 정보를 이용하여  
Enter password: ***** user명이 root인 계정을 패스워드를 이용하여 MySQL  
Welcome to the MySQL monitor. Commands end with ; or \g. 에 로그인한다.  
Your MySQL connection id is 8  
Server version: 8.0.16 MySQL Community Server - GPL  
  
Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> _
```

- MySQL에 성공적으로 로그인했다면 이제 계정 정보를 저장할 Database와 Table을 만들어줌

```
mysql> CREATE DATABASE User;
Query OK, 1 row affected (0.03 sec)

mysql> use User;
Database changed
mysql> CREATE TABLE account
-> (
-> id int NOT NULL AUTO_INCREMENT,
-> username char(32),
-> userpass char(32),
-> PRIMARY KEY(id)
-> );
Query OK, 0 rows affected (0.09 sec)

mysql> show tables;
+-----+
| Tables_in_user |
+-----+
| account        |
+-----+
1 row in set (0.06 sec)
```

- User라는 Database를 생성하고 그 아래에 account라는 테이블을 생성.
- 해당 테이블에는 id와 username과 userpass라는 필드가 존재
username은 아이디를 의미하며 userpass는 비밀번호를 의미
- Id는 각 레코드를 구분 짓는 필드로 중복 없이 고유한 값을 갖는다.

User (Database)

User (Database)

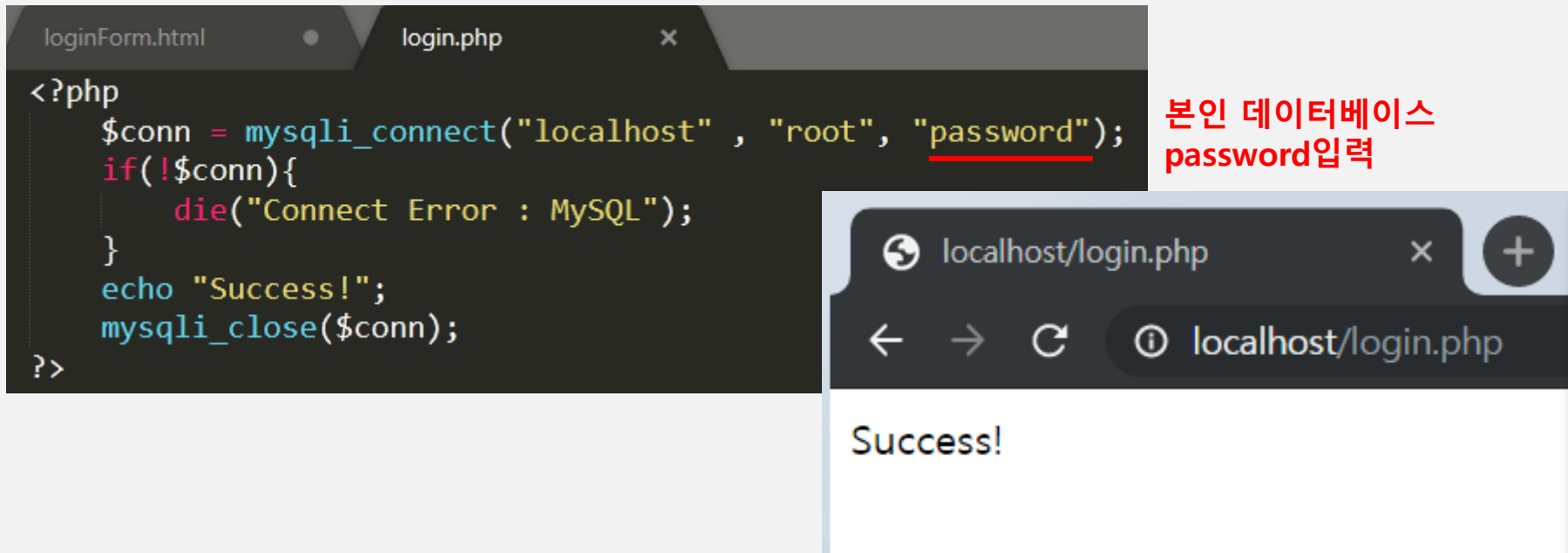
```
mysql> INSERT INTO account(username, userpass) VALUES('user123', 'password123');
```

```
Query OK, 1 row affected (0.04 sec)
```

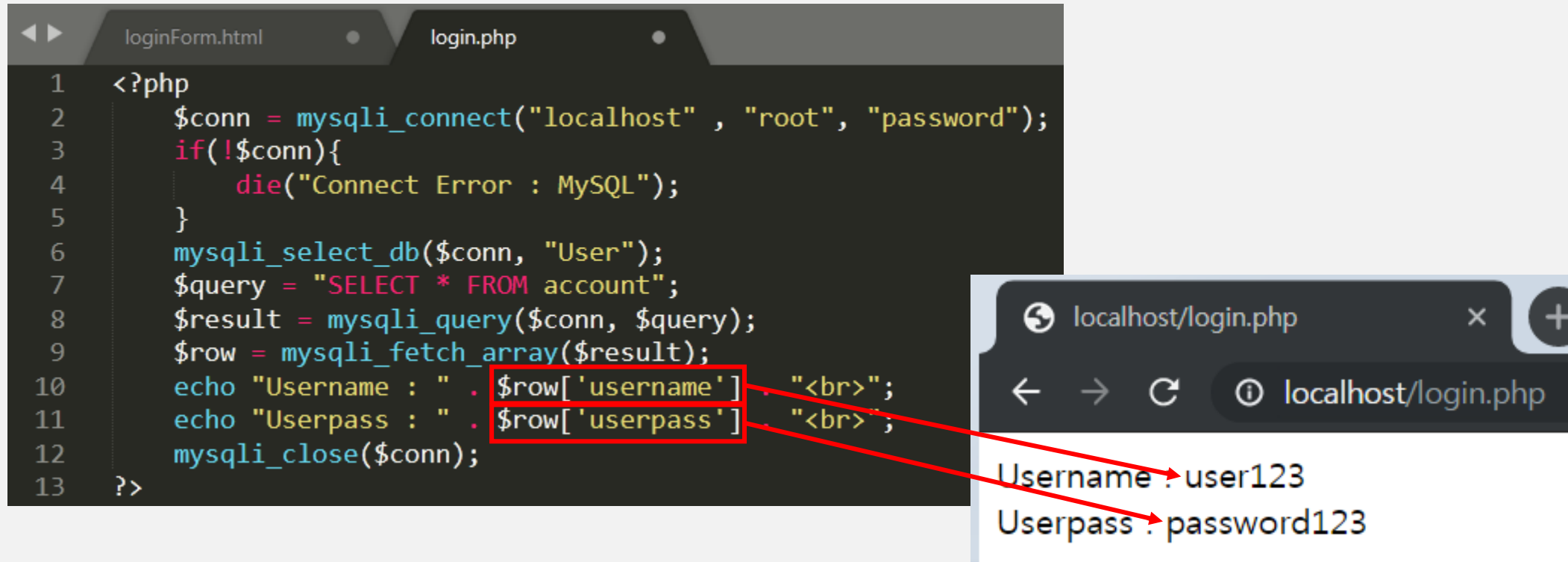
```
mysql> SELECT * FROM account;
```

```
+-----+-----+-----+
| id | username | userpass |
+-----+-----+-----+
| 1 | user123 | password123 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

- 로그인 구현 확인을 위하여 user123 / password123 이라는 계정을 하나 생성
- 잘 생성이 되었는지 확인하기 위하여 SELECT로 확인



- mysqli_connect함수로 PHP에서 MySQL과 연결하여 조건문으로 연결에 성공했는지 확인
- 작업을 모두 마친 뒤에는 연결을 종료하는 mysqli_close함수를 써줌



The image shows a code editor with a file named `login.php` open. The code is a PHP script that connects to a MySQL database, selects a database, runs a query to fetch all data from an 'account' table, and then outputs the results. The output is displayed in a web browser at `localhost/login.php`. The browser shows the output: `Username : user123` and `Userpass : password123`. Red arrows point from the `$row['username']` and `$row['userpass']` in the code to the corresponding output in the browser.

```
1 <?php
2 $conn = mysqli_connect("localhost" , "root", "password");
3 if(!$conn){
4     die("Connect Error : MySQL");
5 }
6 mysqli_select_db($conn, "User");
7 $query = "SELECT * FROM account";
8 $result = mysqli_query($conn, $query);
9 $row = mysqli_fetch_array($result);
10 echo "Username : " . $row['username'] . "<br>";
11 echo "Userpass : " . $row['userpass'] . "<br>";
12 mysqli_close($conn);
13 ?>
```

localhost/login.php

Username : user123
Userpass : password123

- `mysqli_select_db`함수로 사용할 database를 선택
- `mysqli_query`함수는 작성한 쿼리를 실행하여 결과를 반환 받는 함수
- `mysql_fetch_array`함수는 쿼리의 결과를 배열로 바꿔주며 필드 명을 이용하여 값을 사용할 수 있음

Login – DB connect

```
loginForm.html x login.php
1 <?php
2 $conn = mysqli_connect("localhost" , "root", "password");
3 if(!$conn){
4     die("Connect Error : MySQL");
5 }
6 mysqli_select_db($conn, "User");
7 $query = "SELECT * FROM account WHERE username = '{$_POST['id']}' AND
8         userpass = '{$_POST['password']}'";
9 $result = mysqli_query($conn, $query);
10 $row = mysqli_fetch_array($result);
11
12 if(!$row){
13     echo '<script>alert("아이디 혹은 비밀번호가 잘못되었습니다.");';
14     echo 'location.href="/loginForm.html";</script>';
15 }
16
17 echo "안녕하세요 " . $row['username'] . "님";
18
19 mysqli_close($conn);
20 ?>
```

- POST로 가져온 username과 userpass를 이용하여 인증을 진행


```
loginForm.html x index.php ● login.php x
<?php
$conn = mysqli_connect("localhost" , "root", "asdfasdf");
if(!$conn){
    die("Connect Error : MySQL");
}
mysqli_select_db($conn, "User");
$query = "SELECT * FROM account WHERE username = '{$_POST['id']}' AND userpass = '{$_POST['password']}'";
$result = mysqli_query($conn, $query);
$row = mysqli_fetch_array($result);

if(!$row){
    echo '<script>alert("아이디 혹은 비밀번호가 잘못되었습니다.");';
    echo 'location.href="/loginForm.html";</script>';
}
setcookie("user", $_POST['id'], time() + 3600);
echo '<script>location.href="/"</script>';

mysqli_close($conn);
?>
```

- 인증 유지를 위하여 cookie를 생성
- user라는 이름의 쿠키를 생성하고 그 값은 POST로 전달 받은 id로 설정하며
time() + 3600은 쿠키의 만료 시간을 의미함

```
loginForm.html x index.php login.php x
<?php
    if(isset($_COOKIE['user'])){
        echo "Hello " . $_COOKIE['user'];
    }

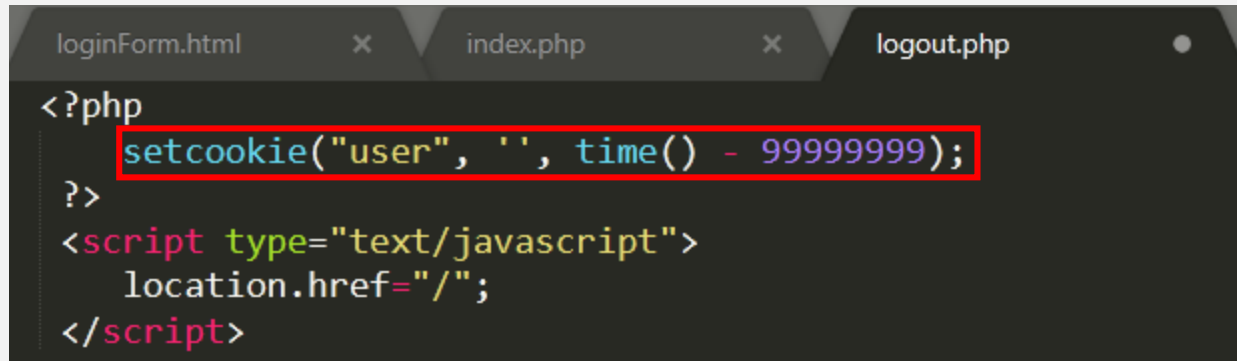
    else{
        echo '<script>alert("로그인이 필요합니다.");';
        echo 'location.href="/loginForm.html";</script>';
    }
?>
```

- index.php에서는 cookie를 이용하여 로그인 여부를 확인
- isset함수는 해당 변수가 존재하는지 확인하는 함수.
즉, user라는 이름의 쿠키가 존재하는지 확인하여 로그인 여부를 확인
- 조건문을 통해 로그인을 확인하는 코드는 권한이 필요한 페이지마다 모두 존재해야 함
- 현재는 로그아웃 기능이 없으므로 로그아웃 기능을 추가해야 함

```
loginForm.html x index.php x logout.php ● login.php x
<?php
    if(isset($_COOKIE['user'])){
        echo "Hello " . $_COOKIE['user'] . "<br>";
        echo '<a href="logout.php">로그아웃</a>';
    }

    else{
        echo '<script>alert("로그인이 필요합니다.");';
        echo 'location.href="/loginForm.html";</script>';
    }
?>
```

- 로그인한 경우 Logout을 진행하는 logout.php로 하이퍼링크가 걸려 있는 로그아웃이라는 글자를 보여줌



```
loginForm.html x index.php x logout.php
<?php
    setcookie("user", '', time() - 99999999);
?>
<script type="text/javascript">
    location.href="/";
</script>
```

- user라는 쿠키의 값을 비어 있도록 설정하였고 만료 기간을 과거로 두어 해당 쿠키의 효력을 무효화 시킴
- 이후 다시 /index.php로 Redirection 하여 로그인을 하도록 유도
- 현재는 회원 가입 기능이 없으므로 SQL의 Insert를 이용하여 회원 가입을 구현

```
loginForm.html x index.php x logout.php login.php
<!DOCTYPE html>
<html>
<head>
  <title>로그인 페이지</title>
  <meta charset="utf-8">
</head>
<body>
  <form action="login.php" method="POST">
    <input type="text" name="id" placeholder="ID"><br>
    <input type="password" name="password" placeholder="PW"><br>
    <a href="signup.html">Sign Up</a>
    <button type="submit">LOGIN</button>
  </form>
</body>
</html>
```

- 로그인 버튼 왼쪽에 회원 가입을 진행하는 signup.html로 Redirect하는 링크를 만듦

```
loginForm.html x signup.html index.php x logout.php
<!DOCTYPE html>
<html>
<head>
  <title>회원 가입</title>
  <meta charset="utf-8">
</head>
<body>
  <form action="signup.php" method="POST">
    <input type="text" name="id" placeholder="ID"><br>
    <input type="password" name="password" placeholder="PW"><br>
    <button type="submit">회원 가입</button>
  </form>
</body>
</html>
```

- 회원 가입을 위해 username과 userpass를 입력 받아야 한다. form을 이용하여 입력 받은 뒤 signup.php로 인자를 전달하여 회원 가입을 진행

```
loginForm.html x signup.html signup.php login.ph
<?php
$conn = mysqli_connect("localhost", "root", "password");
if(!$conn){
    die('Connect Error : MySQL');
}

mysqli_select_db($conn, "User");
$query = "
    INSERT INTO account(username, userpass)
    VALUES('{$_POST['id']}', '$_POST['password'])
";
$result = mysqli_query($conn, $query);
if(!$result){
    echo '<script>alert("회원 가입에 실패하였습니다.);';
    echo "history.back();</script>";
}
?>
<script type="text/javascript">
    alert("회원 가입에 성공하였습니다. 로그인 페이지로 이동합니다.");
    location.href="/loginForm.html";
</script>
```

- SQL의 INSERT를 이용하여 회원 가입을 진행. POST 방식으로 전달 받은 인자를 이용하여 INSERT문의 VALUES 부분을 채워 줌
- 회원 가입의 성공 여부에 따라서 다른 메시지를 띄워주며 만약 성공했다면 로그인 페이지로 이동하고 실패 했다면 다시 회원 가입 페이지로 돌아감
- 현재는 중복된 아이디를 체크하는 기능이 없으므로 DB 조회를 통해 이미 아이디가 존재하는지 확인하는 코드가 필요

```
$query = "SELECT * FROM account WHERE username = '{$_POST['id']}'";
$result = mysqli_query($conn, $query);
if(mysqli_num_rows($result) > 0){
    echo '<script>alert("이미 존재하는 아이디 입니다.");';
    echo 'history.back();</script>';
    exit;
}

$query = "
    INSERT INTO account(username, userpass)
    VALUES('{$_POST['id']}', '{$_POST['password']}')
";
```

- SELECT를 통해 회원 가입 하려는 아이디가 이미 존재하는지 확인하는 코드
- SELECT로 DB를 조회하고 몇 개의 레코드가 반환되었는지 mysqli_num_rows함수로 검사

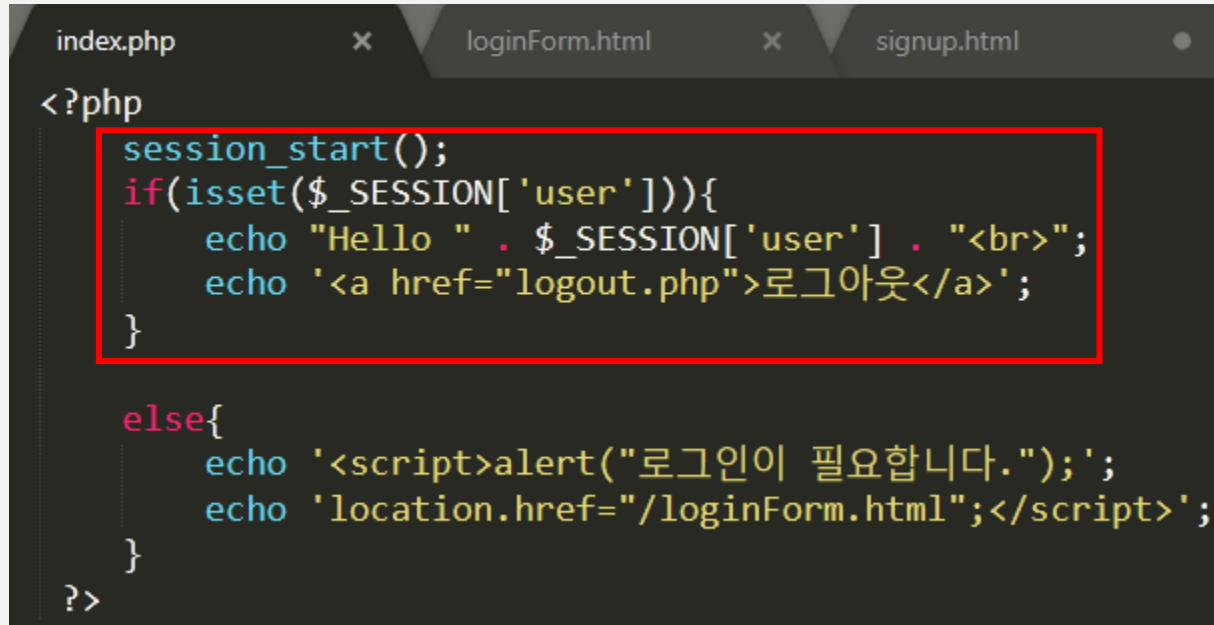

```
loginForm.html x signup.html x signup.php x login.php x index.php x logout.php
<?php
    session_start();
    $conn = mysqli_connect("localhost" , "root", "asdfasdf");
    if(!$conn){
        die("Connect Error : MySQL");
    }
    mysqli_select_db($conn, "User");
    $query = "SELECT * FROM account WHERE username = '{$_POST['id']}' AND userpass = '{$_POST['password']}'";
    $result = mysqli_query($conn, $query);
    $row = mysqli_fetch_array($result);

    if(!$row){
        echo '<script>alert("아이디 혹은 비밀번호가 잘못되었습니다.");';
        echo 'location.href="/loginForm.html";</script>';
    }

    $_SESSION['user'] = $_POST['id'];
    echo '<script>location.href="/"</script>';

    mysqli_close($conn);
?>
```

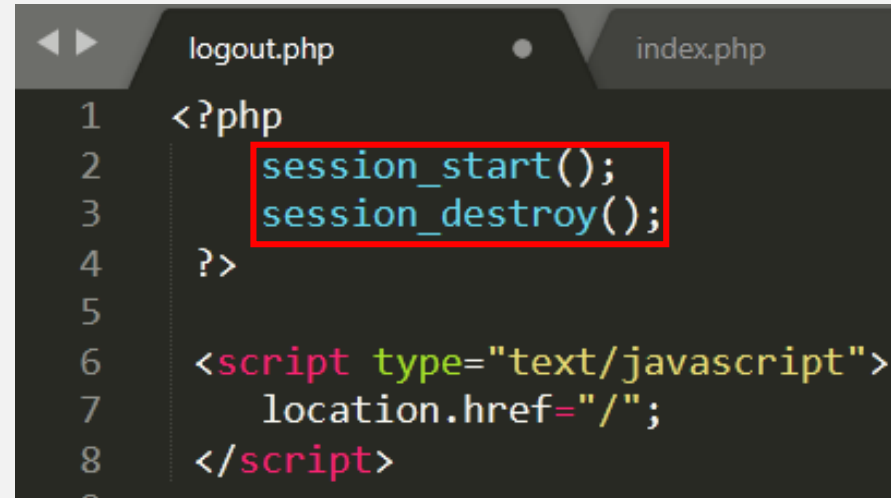
- Session으로 사용하기 위해서 session_start()라는 함수가 반드시 필요
- 로그인에 성공하면 setcookie 함수로 쿠키를 생성한 것처럼 session을 만들어 주어야 함



```
<?php
session_start();
if(isset($_SESSION['user'])) {
    echo "Hello " . $_SESSION['user'] . "<br>";
    echo '<a href="logout.php">로그아웃</a>';
}

else {
    echo '<script>alert("로그인이 필요합니다.");';
    echo 'location.href="/loginForm.html";</script>';
}
?>
```

- 항상 Session을 이용하고자 하면 session_start()함수가 필요
- 이전에는 쿠키 값이 존재하는지 확인하였다면 이번에는 Session이 존재하는지 확인하여 로그인 여부를 검증



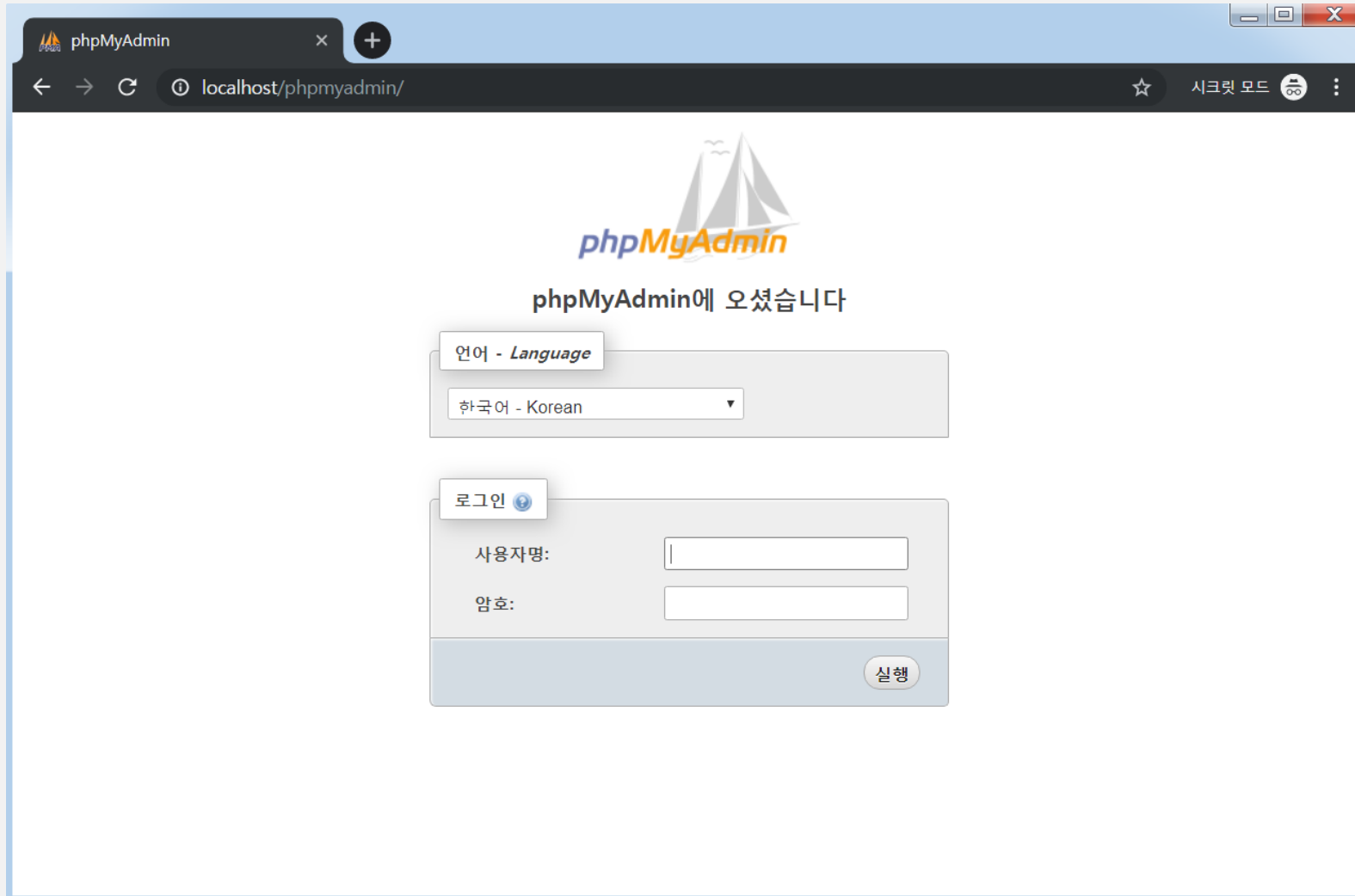
```
1 <?php
2     session_start();
3     session_destroy();
4 ?>
5
6 <script type="text/javascript">
7     location.href="/";
8 </script>
```

- 로그아웃으로 세션을 파기할 때는 session_destroy()함수를 이용
- 역시 세션을 이용하므로 session_start()함수를 먼저 사용

Contents 1

홈페이지 구축

phpMyAdmin



- MySQL을 웹에서 더 쉽고 간편하게 관리할 수 있는 기능
- 로그인 시 실습 때 사용한 정보를 그대로 사용하면 됨
- GUI 기반으로 작동하여 편리함

Contents 1

홈페이지 구축

phpMyAdmin

The screenshot shows the phpMyAdmin web interface in a browser window. The address bar displays the URL: `localhost/phpmyadmin/sql.php?db=user&table=account&pos=0`. The left sidebar shows the database structure, with the 'user' database selected and the 'account' table highlighted. The main panel displays the table structure and data for 'account'.

Database: user, Table: account

0 - 1행 표시 중 (합계 2, 질의 실행시간 0.0010 초.)

`SELECT * FROM `account``

☐ 프로파일링 [\[인라인 편집\]](#) [\[수정\]](#) [\[SQL 해석\]](#) [\[PHP 코드 보기\]](#) [\[다시 보기\]](#)

☐ 모두 보기 | 행 갯수: 25 | 행 필터링: 현재 테이블 검색 | 키로 정렬: 없음

	id	username	userpass
<input type="checkbox"/>	1	user123	password123
<input type="checkbox"/>	3	signup	signup123

☐ 모두 체크 | 선택한 것을: [수정](#) [복사](#) [삭제](#) [내보내기](#)

☐ 모두 보기 | 행 갯수: 25 | 행 필터링: 현재 테이블 검색 | 키로 정렬: 없음

쿼리 결과 처리방법

[인쇄](#) [클립보드에 복사하기](#) [내보내기](#) [차트 표시](#) [뷰 생성](#)

콘솔

감사합니다