

네트워크서비스 프로토콜

7주차 1

» 소프트웨어학부

» 김형균 교수



수업에 들어가며

- 복습
 - 퀴즈
- 오늘 학습 내용
 - 글생성 UI 만들기
 - POST 방식으로 전송된 데이터 받기
 - 파일생성과 리다이렉션
 - 글수정(update) - 수정링크생성
 - 글수정(update) - 수정할 정보 전송
 - 글수정(update) - 수정된 내용으로 변경
 - 글수정(update) - fs.rename()
 - 글 삭제 - 삭제버튼 구현
 - 삭제버튼 위치 지정
 - 글 삭제 - 처리 프로세스

글생성 UI 만들기



노드 웹서버 WEB - create

localhost:3000/create

WELCOME WEB - create

- [css](#)
- [html](#)
- [javascript](#)

[create](#)

title

description

제출

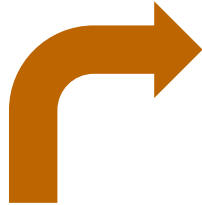
글 생성 UI 만들기



```
function templateHTML(title, list, body){
  return `
    <!doctype html>
    <html>
    <head>
    <title>노드 웹서버 ${title}</title>
    <meta charset="utf-8">
    </head>
    <body>
    <h1><a href="/">WELLCOME ${title}</a></h1>
    ${list}
    <a href="/create">create</a>
    ${body}
    </body>
    </html>
  `;
}
```




글 생성 UI 만들기



```
var app = http.createServer(function(request,response){
  var _url = request.url;
  var queryData = url.parse(_url, true).query;
  var pathname = url.parse(_url, true).pathname;
  if(pathname == '/'){
    if(queryData.id == undefined){...
    } else {...
    }
  } else {
    response.writeHead(404);
    response.end('Not found');
  }
});
app.listen(3000);
```

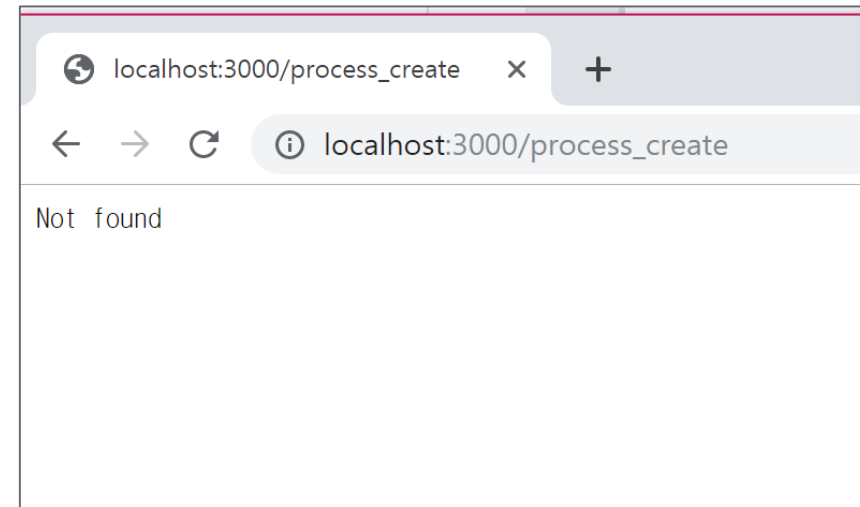
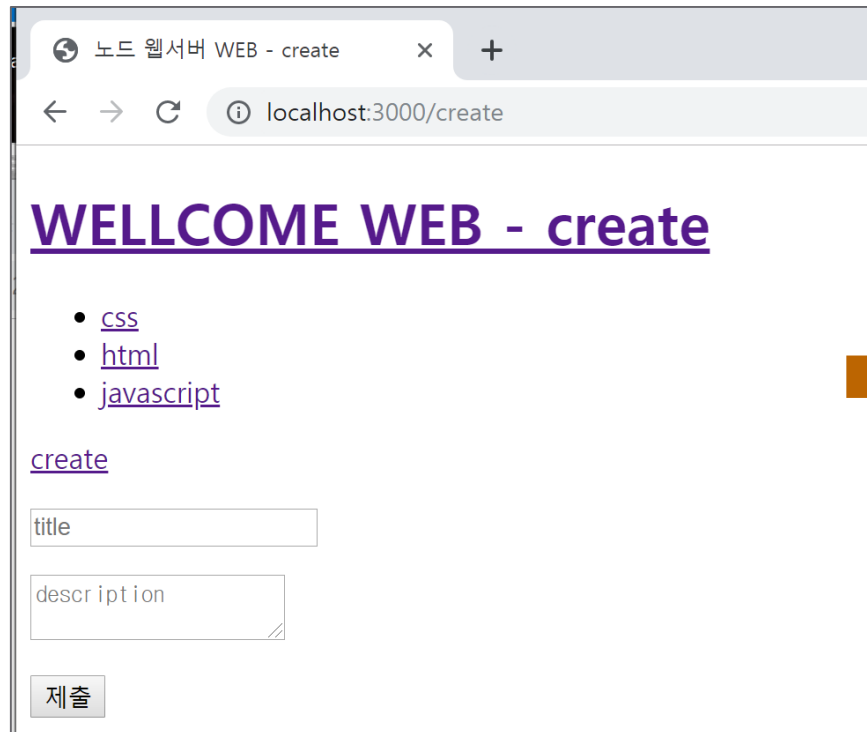
```
var app = http.createServer(function(request,response){
  var _url = request.url;
  var queryData = url.parse(_url, true).query;
  var pathname = url.parse(_url, true).pathname;
  if(pathname == '/'){
    if(queryData.id == undefined){...
    } else {...
    }
  } else if(pathname == '/create'){
    fs.readdir('./data', function(error, filelist){...
    });
  } else {
    response.writeHead(404);
    response.end('Not found');
  }
});
app.listen(3000);
```

```
if(pathname == '/'){  
  if(queryData.id == undefined){...  
  } else {...  
  }  
}  
else if(pathname == '/create'){  
  fs.readdir('./data', function(error, filelist){  
  
  });  
} else {  
  response.writeHead(404);  
  response.end('Not found');  
}
```



```
form action="http://localhost:3000/process_create"
```

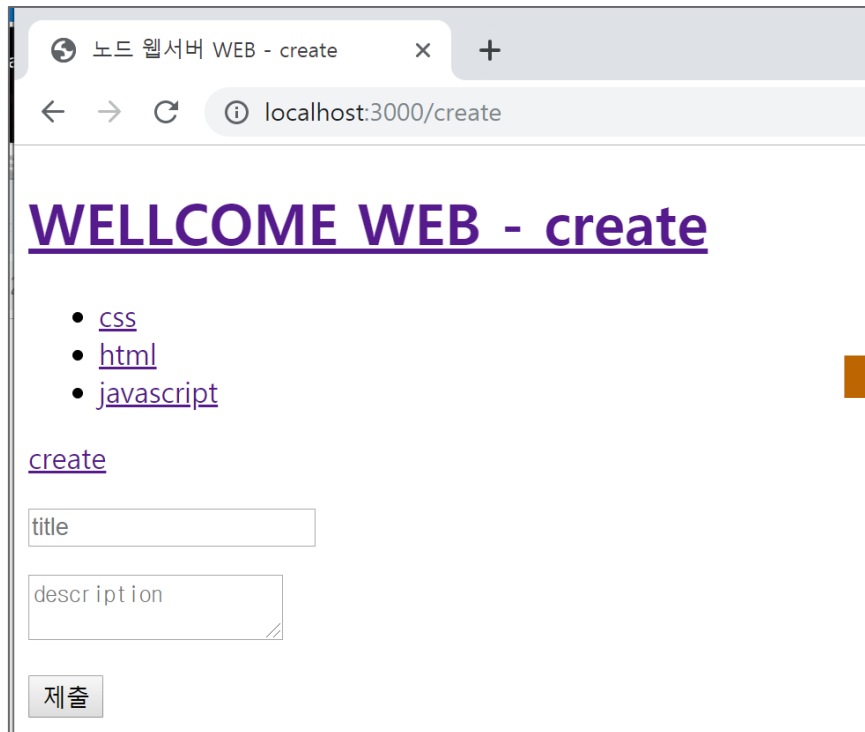
» process_create path에 대한 처리가 아직 되어있지 않음





POST 방식으로 전송된 데이터 받기

» process_create path에 대한 처리가 필요



노드 웹서버 WEB - create

localhost:3000/create

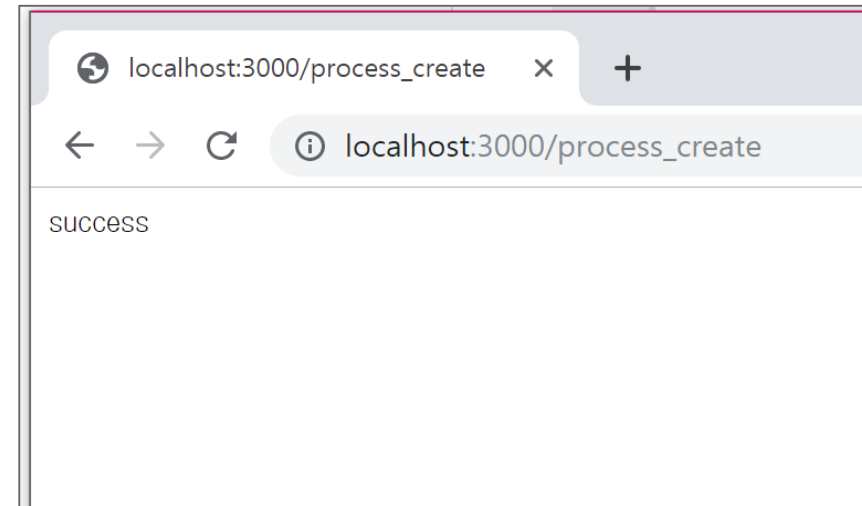
WELLCOME WEB - create

- [css](#)
- [html](#)
- [javascript](#)

[create](#)

title

description





POST 방식으로 전송된 데이터 받기

```
var http = require('http');
var fs = require('fs');
var url = require('url');
var qs = require('querystring');
```

```
var app = http.createServer(function(request, response){
  var _url = request.url;
  var queryData = url.parse(_url, true).query;
  var pathname = url.parse(_url, true).pathname;
  if(pathname == '/'){
    if(queryData.id == undefined){ ...
    } else { ...
    }
  } else if(pathname == '/create'){
    fs.readdir('./data', function(error, filelist){ ...
    });
  } else if(pathname == '/process_create'){ ...
  } else {
    response.writeHead(404);
    response.end('Not found');
  }
});
app.listen(3000);
```



```
var app = http.createServer(function(request, response){
  var _url = request.url;
  var queryData = url.parse(_url, true).query;
  var pathname = url.parse(_url, true).pathname;
  if(pathname == '/'){
    if(queryData.id == undefined){ ...
    } else { ...
    }
  } else if(pathname == '/create'){
    fs.readdir('./data', function(error, filelist){ ...
    });
  } else {
    response.writeHead(404);
    response.end('Not found');
  }
});
app.listen(3000);
```



POST 방식으로 전송된 데이터 받기

» process_create path에 대한 처리

- http.createServer 메서드의 콜백함수 인자인 request의 on메서드를 실행함
- 이에 process_create path에 대한 요청이 올때 data와 end이벤트를 발생해 처리함
- `request.on('data', function(data){ });`
- `request.on('end', function(){ });`



POST 방식으로 전송된 데이터 받기

```
} else if(pathname == '/process_create'){
```



```
response.writeHead(200);  
response.end('success');
```

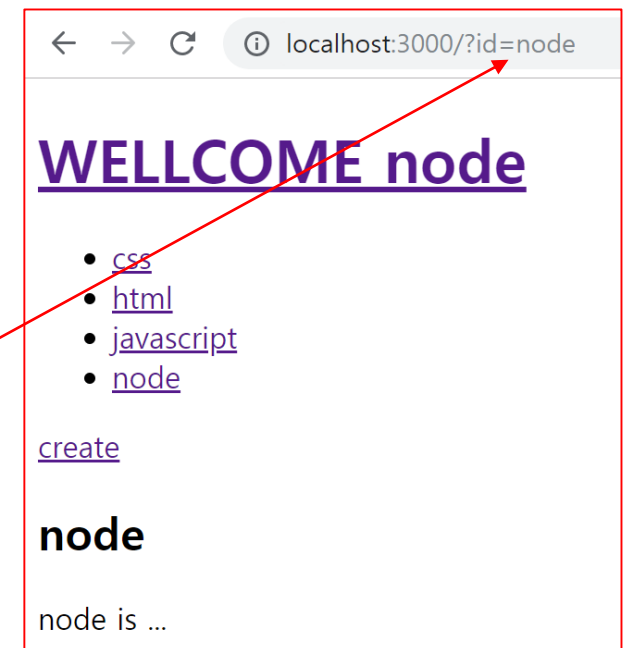
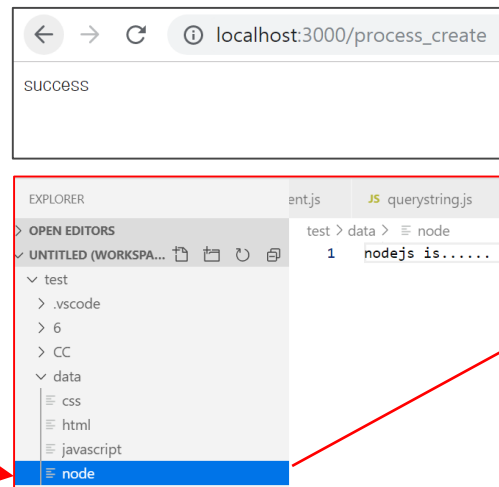
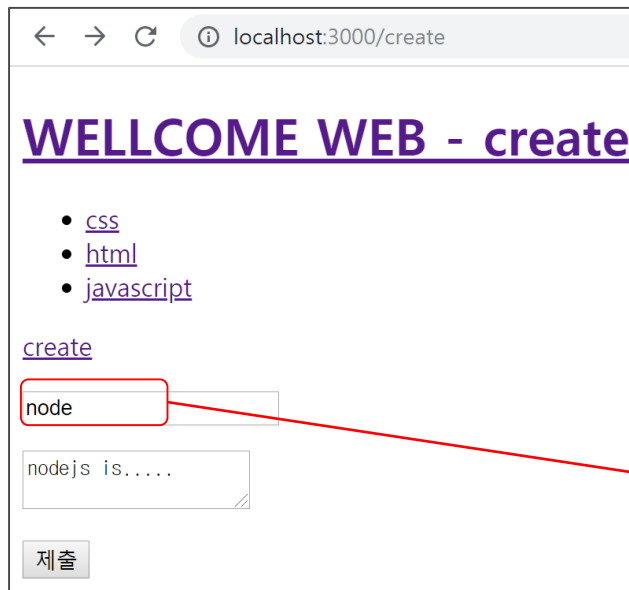
파일생성과 리다이렉션

» 전송된 POST 데이터를 받아서 파일에 저장

- fs.writeFile()

» 그 결과 페이지로 리다이렉션하기

- 특정 페이지로 강제로 보내고자 할 때 응답 헤더의 Location 속성을 사용



파일생성과 리다이렉션

```
app.listen(3000);
```



글수정(update) - 수정링크생성

The diagram illustrates the state of a web application across three browser windows:

- Home Page (localhost:3000):** Displays the title "WELCOME Home", a list of links (111, css, eee, html, javascript, node, sss), a "create" link, and the text "Home" and "Hello, Node.js".
- WELCOME html Page (localhost:3000/?id=html):** Displays the title "WELCOME html", the same list of links, "create" and "update" links, the text "html", and the message "이곳은 HTML문서입니다." (This is an HTML document).
- Not found Page (localhost:3000/update?id=html):** Displays the message "Not found".



```
function templateHTML(title, list, body){  
  return `  
    <!doctype html>  
    <html>  
    <head>  
    <title>노드 웹서버 ${title}</title>  
    <meta charset="utf-8">  
    </head>  
    <body>  
    <h1><a href="/">WELLCOME ${title}</a></h1>  
    ${list}  
    <a href="/create">create</a>  
    ${body}  
    </body>  
    </html>  
  `;  
}
```

```
function templateHTML(title, list, body, control){  
  return `  
    <!doctype html>  
    <html>  
    <head>  
    <title>노드 웹서버 ${title}</title>  
    <meta charset="utf-8">  
    </head>  
    <body>  
      
    </body>  
    </html>  
  `;  
}
```



```
if(pathname == '/'){
  if(queryData.id == undefined){
    fs.readdir('./data', function(error, filelist){
      var title = 'Home';
      var description = 'Hello, Node.js';
      var list = templateList(filelist);

      response.writeHead(200);
      response.end(template);
    })
  }
}
```

```
if(pathname == '/'){
  if(queryData.id == undefined){
    fs.readdir('./data', function(error, filelist){
      var title = 'Home';
      var description = 'Hello, Node.js';
      var list = templateList(filelist);
      var template = templateHTML(title, list, `

## ${title}</h2>${description}`); response.writeHead(200); response.end(template); }) } }


```




```
} else {
  fs.readdir('./data', function(error, filelist){
    fs.readFile(`data/${queryData.id}`, 'utf8', function(err, description){
      var title = queryData.id;
      var list = templateList(filelist);
      response.writeHead(200);
      response.end(template);
    });
  });
}
```

```
} else {
  fs.readdir('./data', function(error, filelist){
    fs.readFile(`data/${queryData.id}`, 'utf8', function(err, description){
      var title = queryData.id;
      var list = templateList(filelist);
      var template = templateHTML(title, list, `

## ${title}</h2>${description}`); response.writeHead(200); response.end(template); }); }); }


```



글수정(update) - 수정할 정보 전송

노드 웹서버 html x +

localhost:3000/update?id=html

WELLCOME html

- [111](#)
- [css](#)
- [eee](#)
- [html](#)
- [javascript](#)
- [node](#)
- [sss](#)

[create update](#)

이곳은 HTML문서입니다.

노드 웹서버 html x +

localhost:3000/update?id=html

WELLCOME html

- [111](#)
- [css](#)
- [eee](#)
- [html](#)
- [javascript](#)
- [node](#)
- [sss](#)

[create update](#)

이곳은 HTML2로
수정된 문서입니다.

localhost:3000/update_process x +

localhost:3000/update_process

Not found

노드 웹서버 html2 x +

localhost:3000/?id=html2

WELLCOME html2

- [111](#)
- [css](#)
- [eee](#)
- [html2](#)
- [javascript](#)
- [node](#)
- [sss](#)

[create update](#)

html2

이곳은 HTML2 수정된 문서입니다.



글수정(update) - 수정할 정보 전송



```
} else if(pathname == '/process_create'){
  var body = '';
  request.on('data', function(data){
    body = body + data;
  });
  request.on('end', function(){
    var post = qs.parse(body);
    var title = post.title;
    var description = post.description;
    fs.writeFile(`data/${title}`, description, 'utf8', function(err){
      response.writeHead(302, {Location: `/?id=${title}`});
      response.end();
    });
  });
} else {
  response.writeHead(404);
  response.end('Not found');
}
});
app.listen(3000);
```

```
} else if(pathname == '/process_create'){
  var body = '';
  request.on('data', function(data){
    body = body + data;
  });
  request.on('end', function(){
    var post = qs.parse(body);
    var title = post.title;
    var description = post.description;
    fs.writeFile(`data/${title}`, description, 'utf8', function(err){
      response.writeHead(302, {Location: `/?id=${title}`});
      response.end();
    });
  });
} else if(pathname == '/update'){ ...
} else {
  response.writeHead(404);
  response.end('Not found');
}
});
app.listen(3000);
```



글수정(update) - 수정할 정보 전송

» 수정한 문서의 쿼리스트링 id 정보

```
<form action="/update_process" method="post">
```

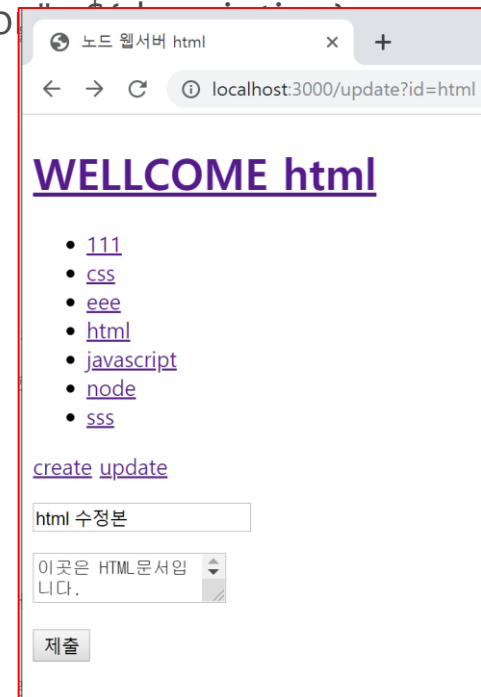
```
  <input type="hidden" name="id" value="{title}" >
```

» 수정한 문서의 정보

```
<p><input type="text" name="title" placeholder="title" value="{title}"></p>
```

```
<p><textarea name="description" placeholder="description" >
```

```
</textarea></p>
```





글수정(update) - 수정할 정보 전송

```
} else if(pathname == '/update'){
```

```
} else {
```

글수정(update) – 수정된 내용으로 변경 파일명, 본문내용 수정



```
var app = http.createServer(function(request,response){
  var _url = request.url;
  var queryData = url.parse(_url, true).query;
  var pathname = url.parse(_url, true).pathname;
  if(pathname == '/'){...
} else if(pathname == '/create'){...
} else if(pathname == '/process_create'){...
} else if(pathname == '/update'){...
} else if(pathname === '/update_process'){...
} else {
  response.writeHead(404);
  response.end('Not found');
}
});
app.listen(3000);
```



fs.rename()

» fs.rename(기존 경로, 새 경로, 콜백)

- 파일의 이름을 바꾸는 메서드입니다.
- 기존 파일 위치와 새로운 파일 위치를 적어주면 됩니다.
- 반드시 같은 폴더를 지정할 필요는 없으므로 잘라내기 같은 기능을 할 수도 있습니다.

노드 웹서버 html

localhost:3000/update?id=html

WELCOME html

- [111](#)
- [css](#)
- [eee](#)
- [html](#)
- [javascript](#)
- [node](#)
- [sss](#)

[create update](#)

이곳은 HTML 문서입니다.

제출

노드 웹서버 html

localhost:3000/update?id=html

WELCOME html

- [111](#)
- [css](#)
- [eee](#)
- [html](#)
- [javascript](#)
- [node](#)
- [sss](#)

[create update](#)

이곳은 HTML2로 수정된 문서입니다.

제출

노드 웹서버 html2

localhost:3000/?id=html2

WELCOME html2

- [111](#)
- [css](#)
- [eee](#)
- [html2](#)
- [javascript](#)
- [node](#)
- [sss](#)

[create update](#)

html2

이곳은 HTML2 수정된 문서입니다.



```
} else if(pathname === '/update_process'){
```

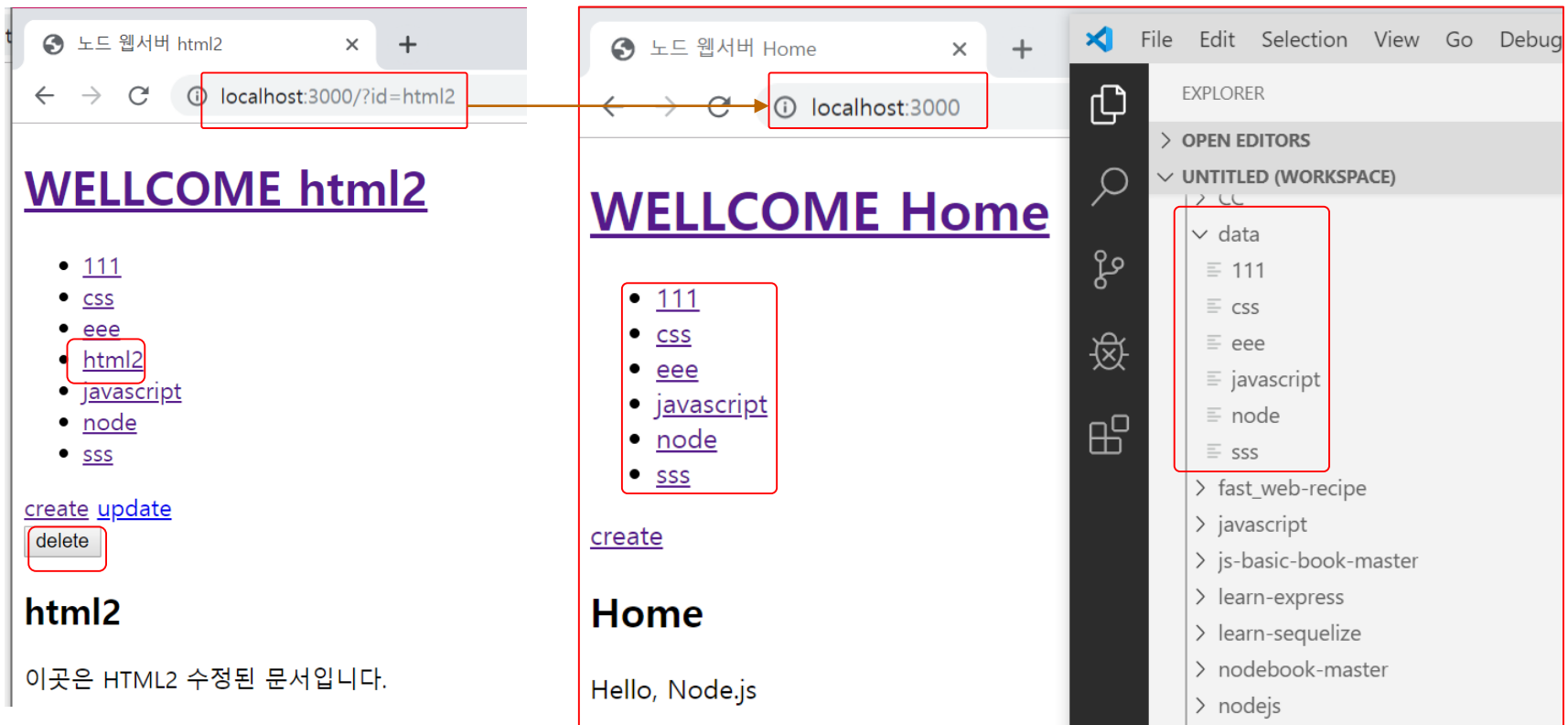
```
} else {
```


글삭제 - 삭제버튼 구현

» fs.unlink(경로, 콜백)

- 파일을 지울 수 있습니다.
- 파일이 없다면 에러가 발생하므로 먼저 파일이 있는지를 꼭 확인해야 합니다.

» 삭제는 하이퍼링크로 구현하지 않는 것이 좋다



The image shows a comparison of two web pages and a code editor. The left browser window displays 'WELLCOME html2' with a sidebar menu containing links like '111', 'css', 'eee', 'html2' (highlighted), 'javascript', 'node', and 'sss'. Below the menu are buttons for 'create', 'update', and 'delete' (highlighted). The right browser window displays 'WELLCOME Home' with a similar menu and a 'create' button. The VS Code Explorer on the right shows a file tree with a 'data' folder containing files '111', 'css', 'eee', 'javascript', 'node', and 'sss' (all highlighted). An arrow indicates a link from 'html2' in the left browser to 'localhost:3000' in the right browser.



삭제버튼 위치 지정

```
var app = http.createServer(function(request, response){
  var _url = request.url;
  var queryData = url.parse(_url, true).query;
  var pathname = url.parse(_url, true).pathname;
  if(pathname == '/'){
    if(queryData.id == undefined){...
    } else {
      fs.readdir('./data', function(error, filelist){
        fs.readFile(`data/${queryData.id}`, 'utf8', function(err, description){
          var title = queryData.id;
          var list = templateList(filelist);
          var template = templateHTML(title, list,
            `

## ${title}</h2>${description}`, `


```

```
var app = http.createServer(function(request,response){
  var _url = request.url;
  var queryData = url.parse(_url, true).query;
  var pathname = url.parse(_url, true).pathname;
  if(pathname == '/'){
    if(queryData.id == undefined){...
    } else {
      fs.readdir('./data', function(error, filelist){
        fs.readFile(`data/${queryData.id}`, 'utf8', function(err, description){
          var title = queryData.id;
          var list = templateList(filelist);
          var template = templateHTML(title, list,
            `

## ${title}</h2>${description}`, ); response.writeHead(200); response.end(template); }); }); }); }); }


```



글 삭제 - 처리 프로세스

```
    } else if(pathname === '/delete_process'){ ...  
    } else {  
      response.writeHead(404);  
      response.end('Not found');  
    }  
  });  
app.listen(3000);
```



```
} else if(pathname === '/delete_process'){  
      
} else {  
    response.writeHead(404);  
    response.end('Not found');  
}  
});  
app.listen(3000);
```