

O p e n Y e a r R o u n d 3 조

jQuery Json Ajax

발표자

박현우
김현지

CONTENTS

1. jQuery
2. json
3. Ajax
4. 과제 코드 리뷰

“

jQuery

”

- 오픈소스 기반의 자바스크립트 라이브러리로, 사용하기 간편하다는 장점이 있습니다.
- 직접 다운받는 방식, CDN을 이용하는 방식이 있습니다.
- 요즘은 많이 사용하지 않는 추세입니다.

jQuery

1. jQuery는 주요 웹 브라우저의 구버전을 포함한 대부분의 브라우저에서 지원됩니다.
2. HTML DOM을 손쉽게 조작할 수 있으며, CSS 스타일도 간단히 적용할 수 있습니다.
3. 애니메이션 효과나 대화형 처리를 간단하게 적용해 줍니다.
4. 같은 동작을 하는 프로그램을 더욱 짧은 코드로 구현할 수 있습니다.
5. 다양한 플러그인과 참고할 수 있는 문서가 많이 존재합니다.
6. 오픈 라이선스를 적용하여 누구나 자유롭게 사용할 수 있습니다.

jQuery

1. 직접 파일을 다운받아 로드하는 방법

- `<script type= " text/javascript " src= " jquery.js " ></script>`

2. CDN을 이용하여 로드하는 방법 (가장 가까운 서버에서 다운받는 기술)

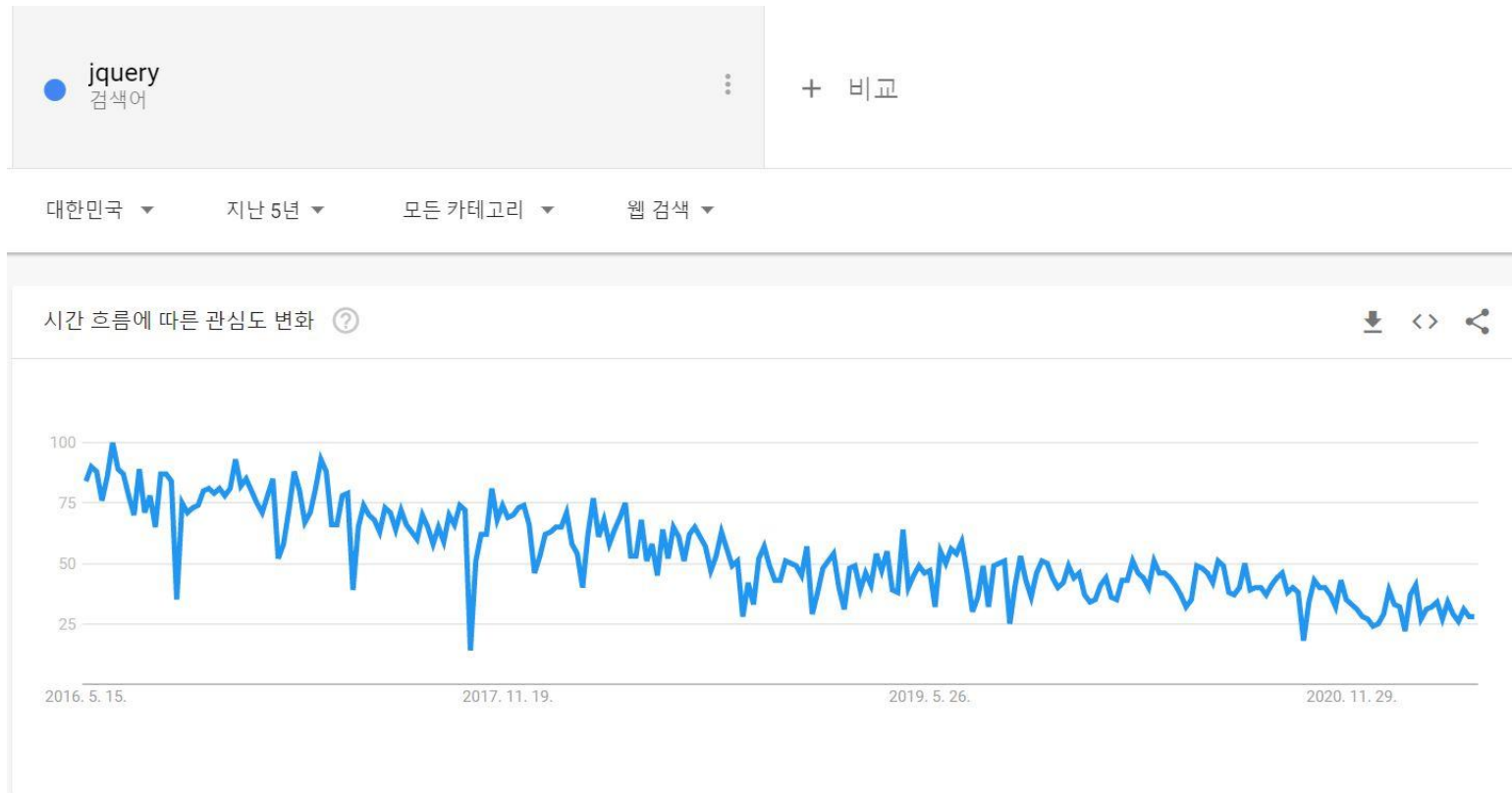
- `<script src= " https://code.jquery.com/jquery-3.6.0.js
crossorigin="anonymous"></script>`

jQuery

1. 기본 문법 : `$(선택자).동작함수();`
2. 태그 : `$("div")`
3. 클래스 : `$(".클래스명")`
4. 아이디 : `$("#id")` -> `document.getElementById('id')` 보다 훨씬 짧다.
5. 함수에 관한 더 많은 자료

https://github.com/demun/FrontEndStudy/blob/master/document/Jquery/docs/01_00_jquery_%EA%B8%B0%EC%B4%88%EB%AC%B8%EC%84%9C.md

jQuery



- 5년 전과 비교하여 관심도가 많이 줄어 들었습니다.

- JavaScript Object Notation의 약자입니다.
- 서버와 클라이언트가 데이터를 주고 받을 때 사용합니다.
- Key와 value로 이루어져 있습니다.
- 프로그래밍 언어와 플랫폼에 상관없이 사용할 수 있습니다.
- 텍스트로 이루어져 있어 사람과 기계 모두 읽고 쓰기 쉽습니다.
- XML에 비하여 경량적이며 빠르다는 장점이 있습니다.

기본 자료형

- 수(Number)
- 문자열(String)
- 참/거짓(Boolean)
- 배열(Array)
- 객체(Object)
- Null

```
let json = JSON.stringify(1);  
console.log(json);
```

```
let json = JSON.stringify('문자열');  
console.log(json);
```

```
let json = JSON.stringify(true);  
console.log(json);
```

```
let json = JSON.stringify(['배열1', '배열2']);  
console.log(json);
```

```
json = JSON.stringify(null);  
console.log(json);
```

json

Stringify()

- Object를 JSON으로

```
const bear={
  name: 'teddy',
  color: 'brown',
  size: null,
  birthDate: new Date(),
  jump: ()=>{
    console.log(`${name} can jump!`);
  }
} // data만 JSON으로 변경!! 함수는 제외됨

json = JSON.stringify(bear);
console.log(json);
```

```
{"name":"teddy","color":"brown","size":null, "json.js:40"
"birthDate":"2021-05-15T21:18:19.764Z"}
```

json

Stringify()

```
json = JSON.stringify(bear, ["name"]);
console.log(json);
// 원하는 property만 고르면 이 값만 JSON으로 변경
```

```
{"name": "teddy"}
```

```
json = JSON.stringify(bear, (key, value)=>{
  console.log(`key: ${key}, value: ${value}`)
  return value;
});
console.log(json);
```

```
key: , value: [object Object]
key: name, value: teddy
key: color, value: brown
key: size, value: null
key: birthDate, value: 2021-05-15T21:18:19.764Z
key: jump, value: ()=>{
  console.log(`${name} can jump!`);
}
```

```
json = JSON.stringify(bear, (key, value)=>{
  console.log(`key: ${key}, value: ${value}`)
  return key === 'name' ? 'ted' : value;
});
// key가 name인 경우 ted로 바꾸고 아닌경우 원래 값
console.log(json);
```

```
key: , value: [object Object]
key: name, value: teddy
key: color, value: brown
key: size, value: null
key: birthDate, value: 2021-05-15T21:18:19.764Z
key: jump, value: ()=>{
  console.log(`${name} can jump!`);
}
{"name": "ted", "color": "brown", "size": null, "birthDate": "2021-05-15T21:18:19.764Z"}
```

json

Parse()

```
json = JSON.stringify(bear);

let obj = JSON.parse(json);
console.log(obj);
```

json.js:70
 ▶ {name: "teddy", color: "brown", size: null, birthDate: "2021-05-15T21:26:17.987Z"}

- JSON을 object로

```
console.log(bear.birthDate.getDate());
console.log(obj.birthDate.getDate()); //는 오류

obj = JSON.parse(json, (key, value)=>{
  console.log(`key: ${key}, value: ${value}`)
  return key === 'birthDate' ? new Date(value) : value;
});
console.log(obj);

console.log(obj.birthDate.getDate());
// 사용가능
```

Ajax

“

Ajax

”

- Asynchronous JavaScript And XML의 약자입니다.

자바스크립트를 통해서 서버에 XML, json의 형태로 데이터를 요청합니다.

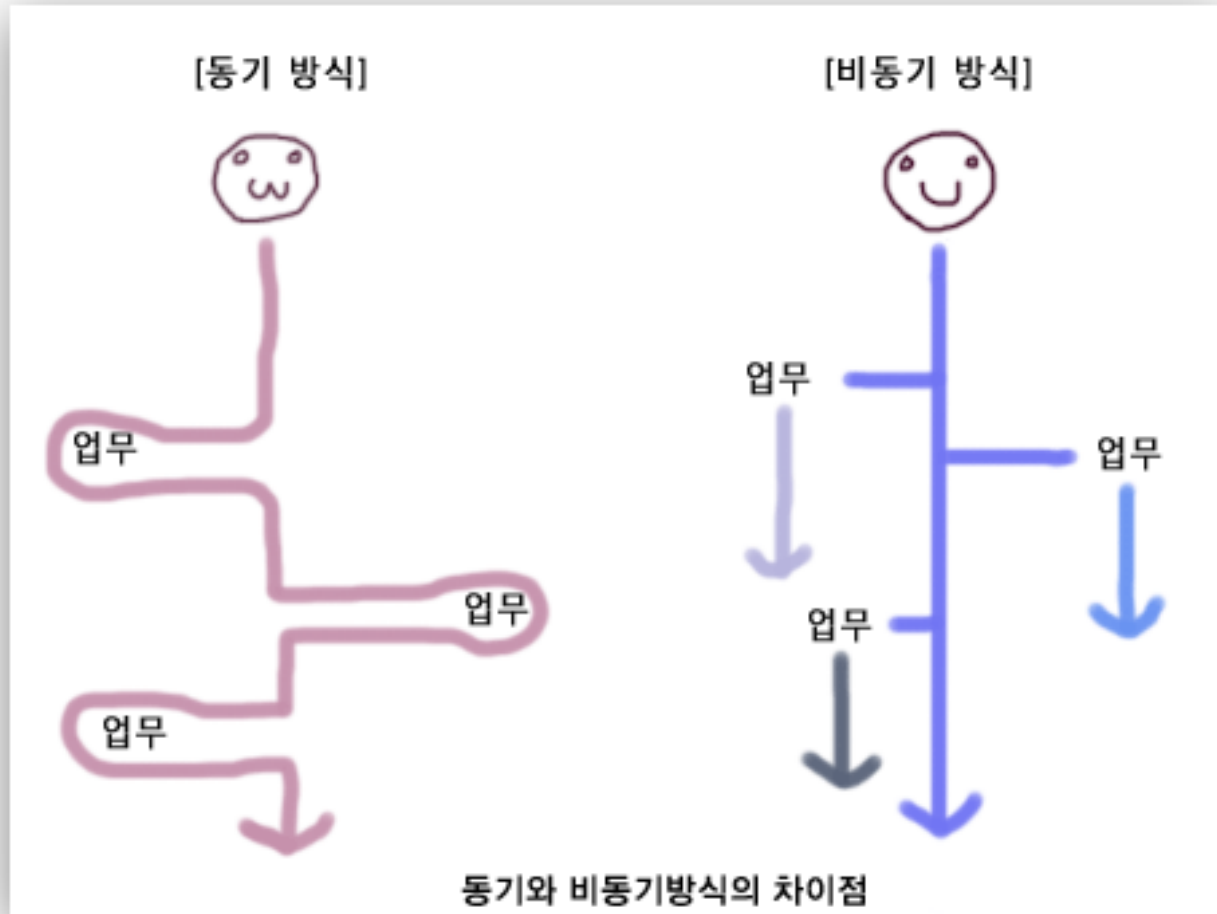
- 비동기 방식: 웹페이지를 리로드하지 않고 필요한 데이터를 불러오는 방식

- 페이지 전체를 다시 불러오지 않고, 필요한 부분만 별도로 요청하고 응답받습니다.

- 모든 자원을 다시 받을 필요가 없으므로 웹페이지의 속도향상에 도움이 됩니다.

- 불필요한 대역폭 감소 -> 비용 절감의 효과가 있습니다.

Ajax



Ajax

```
$.ajax({  
    url: '/study/tmp/test.php', //request 보낼 서버  
    의 경로  
    type: 'post', // 메소드(get, post, put 등)  
    data: {'id': 'admin'}, //보낼 데이터  
    success: function(data) {  
        //서버로부터 정상적으로 응답이 왔을 때 실행  
    },  
    error: function(err) {  
        //서버로부터 응답이 정상적으로 처리되지 못했  
    을 때 실행  
    }  
});
```

Ajax

```
var xmlHttp = new XMLHttpRequest();    // XMLHttpRequest 객체를 생성함.  
xmlHttp.onreadystatechange = function() { // onreadystatechange 이벤트 핸  
    들러를 작성함.  
        // 서버상에 문서가 존재하고 요청한 데이터의 처리가 완료되어 응답할  
        준비가 완료되었을 때  
        if(this.status == 200 && this.readyState == this.DONE) {  
            // 요청한 데이터를 문자열로 반환함.  
            document.getElementById("text").innerHTML = xmlHttp.responseText;  
        }  
    };  
xmlHttp.open("GET", "/examples/media/xml_httpxmlrequest_data.txt", true);  
xmlHttp.send();
```


과제 코드 리뷰 (클라이언트)

```
<form class="outer" action="/feed/" method="post">
  <input id="number" type="text" placeholder="전화번호, 사용자 이름 또는 이메일">
  <input id="password" type="password" placeholder="비밀번호">
  <button type="submit" class="btn" id="join"><b>로그인</b></button>
  <div style="color: ■ #8E8E8E; font-size:15px; margin-top:15px;">----- 또는 -----
</div>
<div style="color:■ #385185; font-size:15px; margin-top:15px;"><b>Facebook으로 로그인</b></div>
<div style="font-size:12px; margin-top:15px; margin-bottom:20px;">비밀번호를 잊으셨나요?</div>
</form>
```

과제 코드 리뷰(서버)

```
router.get('/', function(req, res, next) {
  res.render('login.html');
});

router.get('/feed/', function(req, res, next) {
  /*쿠키가 존재하지 않는다면, 다시 로그인 페이지로 보내야 함*/

  if(!cookieLoginObj){
    res.redirect('/');
  }

  res.render('feed.html');
});

router.post('/feed/', function(req, res, next) {

  /*post로 들어온 경우, 쿠키를 설정해 줌 */

  res.cookie('login', '박현우', {maxAge:300000, httpOnly:true});

  cookieLoginObj = req.cookies.login;
  console.log(cookieLoginObj);
  if(cookieLoginObj){
    console.log('로그인 됨');
  }

  else{
    console.log('로그인 안 됨');
  }

  res.render('feed.html');
});
```

```
app.use(cookieParser());
```

```
var cookieParser = require('cookie-parser');
```

출처

- TCP school
- <https://velog.io/@surim014/AJAX란-무엇인가>
- <https://dororongju.tistory.com/96>
- <https://deeplify.dev/front-end/js/xml-http-request>

**THANK
YOU**