

웹프로그래밍 기초(7강)



1. 용어 정리

AJAX : Asynchronous Javascript And XML

JavaScript를 사용한 비동기 통신, 클라이언트와 서버간에
XML데이터를 주고 받는 기술

=> 자바스크립트를 통해서 서버에 데이터를 요청하는 것

비동기 : 웹페이지를 리로드하지 않고도 데이터를 불러오는 방식

AJAX 코드 예

```
function loadDoc() {  
  var xhttp = new XMLHttpRequest();  
  xhttp.onreadystatechange = function() {  
    if (this.readyState == 4 && this.status == 200) {  
      document.getElementById("demo").innerHTML =  
        this.responseText;  
    }  
  };  
  xhttp.open("GET", "ajax_info.txt", true);  
  xhttp.send();  
}
```

JSON 표현 예시

```
{  
  "userId": 1,  
  "id": 1,  
  "title": "delectus aut autem",  
  "completed": false  
}
```

JSON은 인간과 컴퓨터 모두 읽기 쉽고 사용하기 편리하고
용량도 적어서 현재 데이터 교환 방식으로 가장 많이 사용하고 있습니다

화면에 JSON 내용 뿌리기

```
let resultDiv = document.getElementById('result');

fetch('https://jsonplaceholder.typicode.com/posts/1')
  .then(response => response.json())
  .then((json) => {
    resultDiv.textContent = JSON.stringify(json);
  });

console.log('test');
```

JSON.stringify() :
json 데이터를 string 으로 변환해 주는
내장 함수

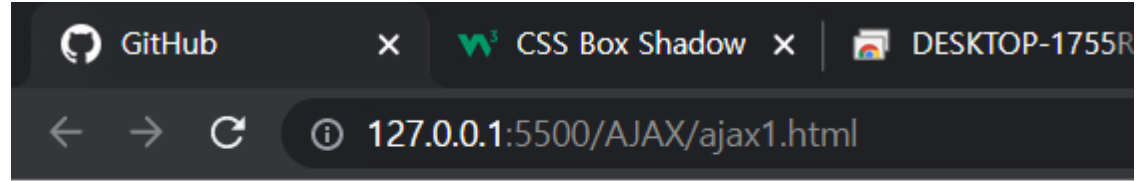
JSON 에서 원하는 항목에 . 으로 간단히 접근 가능

```
fetch('https://jsonplaceholder.typicode.com/posts/1')
  .then(response => response.json())
  .then((json) => {
    resultDiv.textContent = json.title;
  });
```

오전수업 연습

```
10
11 <body>
12   <script>
13     function loadDoc() {
14       //XMLHttpRequest 생성
15       let xhttp = new XMLHttpRequest();
16
17       //응답 왔을 때의 콜백 함수 정의
18       xhttp.onreadystatechange = function () {
19
20         //this.readyState == 4 : 정상 완료 상태 (잘못 응답도 ok)
21         // https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/readyState
22
23
24         //this.status == 200 : 서버 입장에서 정상 동작
25         //this.status == 404 : Not Found
26         //https://developer.mozilla.org/en-US/docs/Web/HTTP/Status
27
28         if (this.readyState == 4 && this.status == 200) {
29           document.getElementById("demo").innerHTML = this.responseText;
30         }
31       };
32
33       // 통신방식이나 대상을 설정
34       // 방식 : GET(읽기) / POST(쓰기) / PATCH / DELETE etc
35       xhttp.open("GET", "ajax_info.txt", true);
36
37       // 실행
38       xhttp.send();
39     }
40   </script>
41
42   <div id="demo"></div>
43   <button onclick="loadDoc()">get data</button>
44 </body>
45
46 </html>
```

```
AJAX > ajax_info.txt
1 hello world
```



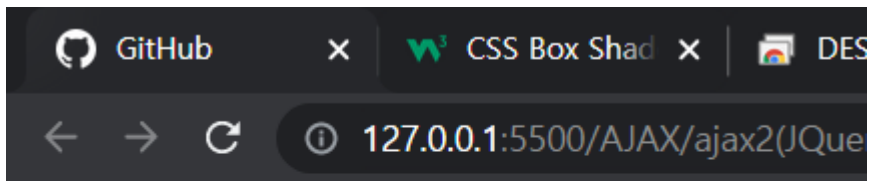
hello world

get data

오전수업 연습

```
AJAX > ajax2(jQuery).html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>ajax-jQuery 형식으로</title>
9   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
10 </head>
11
12 <body>
13   <script>
14     $(function () {
15       $('#get_data_button').click(function () {
16         //GET 방식이 곧 JQUERY에서는 LOAD를 사용한다.
17         $('#demo').load("ajax_info.txt");
18       });
19     });
20   </script>
21
22   <div id="demo"></div>
23   <button id="get_data_button">get data</button>
24 </body>
25
26 </html>
```

```
AJAX > ajax_info.txt
1 hello world
```



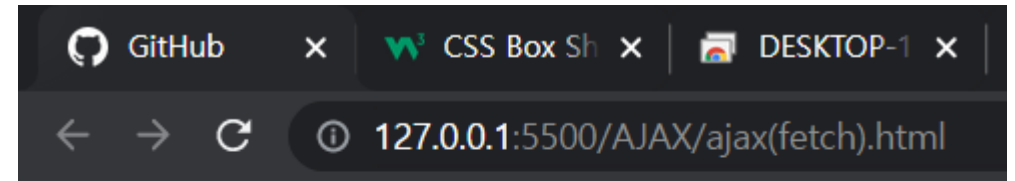
hello world

get data

오후수업 연습

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>ajax(fetch함수로)</title>
9 </head>
10
11 <body>
12   <script>
13     // y = f(x) -> 수학적 표현
14     // function f(x) { // 내용 } - > 일반적인 표현(JS)
15     // x => 내용 -> 람다 표현 방식(js 화살표 표기법, lambda(람다 표기법 형태))
16
17     function loadDoc() {
18       fetch('people.json')
19         .then(response => response.json()) // then : promise 사용 가능
20         // .then(function(response){
21         //   return response.json();          /* 동일한 함수 표현 방식을 람다 방식으로 표기 */
22         // }])
23         .then(json => {
24           document.getElementById("demo").innerHTML = JSON.stringify(json);
25           console.log(json);
26           document.getElementById("result").innerHTML = json.name;
27           document.getElementById("result2").textContent = json.skill;
28
29         });
30   }
31 </script>
32 <div id="demo"></div>
33 <div id="result"></div>
34 <div id="result2"></div>
35 <button onclick="loadDoc()">get data</button>
36
37
38 </body>
39
40 </html>
```

```
AJAX > {} people.json > ...
1  {
2    "name": "손영식",
3    "skill": "java"
4  }
```



{"name":"손영식","skill":"java"}

손영식

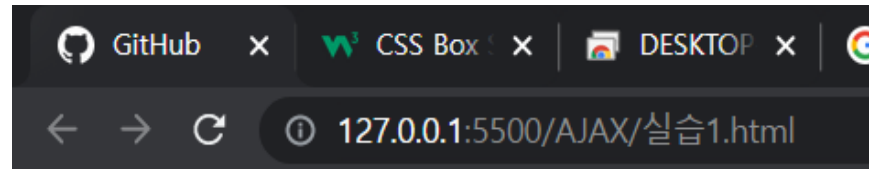
java

get data

실습 - 1

AJAX > 실습1.html > html > body > button

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
9   <title>실습1</title>
10 </head>
11
12 <body>
13   <script>
14     function button1() {
15       fetch('https://jsonplaceholder.typicode.com/todos/1')
16         .then(response => response.json())
17         .then(json => {
18           json.title;
19           $('#list').append(`<li><b>${json.title}</b><b>`);
20         });
21     }
22   </script>
23   <button onclick="button1()">데이터로드</button>
24   <ul id="list">
25     <li id="text"><b>여기</b></li>
26   </ul>
27 </body>
28
29
30 </html>
```

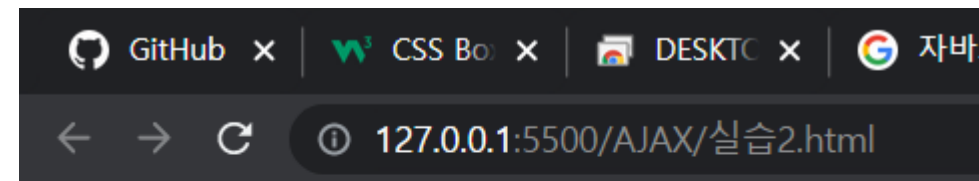


데이터로드

- 여기
- delectus aut autem

실습 - 2

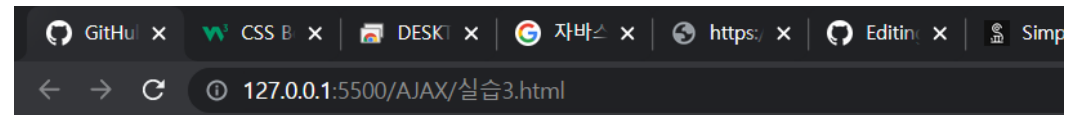
```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
9   <title>실습 2</title>
10 </head>
11
12 <body>
13   <script>
14     $(function () {
15       fetch(['https://jsonplaceholder.typicode.com/todos/1'])
16         .then(response => response.json())
17         .then(json => {
18           json.title;
19           $('#list').append(`<li><b>${json.title}</b><b>`);
20         });
21     });
22   </script>
23
24   <ul id="list">
25     <li id="text"><b>여기</b></li>
26   </ul>
27
28 </body>
29
30 </html>
```



- 여기
- delectus aut autem

실습 - 3

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
9   <title>실습3</title>
10 </head>
11
12 <body>
13   <script>
14     fetch('https://jsonplaceholder.typicode.com/todos/')
15       .then(response => response.json())
16       .then(jsonArray => {
17         //   for (var i = 0; i < json.length; i++) {
18         //     var item = json[i].title;
19         //   }
20
21         let list = $('#list')
22         jsonArray.forEach((data, i) => {
23           console.log(data.title)
24           list.append('<li><b>${data.title}</b></li>')
25         })
26       })
27   </script>
28
29   <ul id="list">
30   </ul>
31 </body>
32
33 </html>
```



- delectus aut autem
- quis ut nam facilis et officia qui
- fugiat veniam minus
- et porro tempora
- laboriosam mollitia et enim quasi adipisci quia provident illum
- qui ullam ratione quibusdam voluptatem quia omnis
- illo expedita consequatur quia in
- quo adipisci enim quam ut ab
- molestiae perspiciatis ipsa
- illo est ratione doloremque quia maiores aut
- vero rerum temporibus dolor
- ipsa repellendus fugit nisi
- et doloremque nulla
- repellendus sunt dolores architecto voluptatum
- ab voluptatum amet voluptas
- accusamus eos facilis sint et aut voluptatem
- quo laboriosam deleniti aut qui
- dolorum est consequatur ea mollitia in culpa
- molestiae ipsa aut voluptatibus pariatur dolor nihil
- ullam nobis libero sapiente ad optio sint
- suscipit repellat esse quibusdam voluptatem incidunt
- distinctio vitae autem nihil ut molestias quo
- et itaque necessitatibus maxime molestiae qui quas velit
- adipisci non ad dicta qui amet quaerat doloribus ea
- voluptas quo tenetur perspiciatis explicabo natus
- aliquam aut quasi
- veritatis pariatur delectus

실습 - 4

```
/* On mouse-over, add a deeper shadow */
.card:hover {
  box-shadow: 0 8px 16px 0 rgba(0, 0, 0, 0.2);
}

/* Add some padding inside the card container */
.container {
  padding: 2px 16px;
}
</style>
<script>

function commentPage(postId) {
  window.location.href = `실습4-comments.html?id=${postId}`;
  // window.location.href = `https://jsonplaceholder.typicode.com/comments/${postId}`;
}

fetch('https://jsonplaceholder.typicode.com/posts')
  .then(response => response.json())
  .then(jsonArray => {
    // for (var i = 0; i < json.length; i++) {
    //   var item = json[i].title;
    // }

    let items = $('#items')
    jsonArray.forEach((data, i) => {
      // console.log(data.id)
      items.append(`<div class="card" onclick = commentPage(${data.id})">
<div class="container">
  <h4><b>${data.id}</b></h4>
  <p>${data.title}</p>
  <p>${data.body}</p>
</div>
</div>`)
    })
  })
</script>

<!-- <button onclick="commentPage(1)">페이지 이동</button> -->
<div id="items"></div>

</body>
```

GitH x W CSS x DESK x 자바 x https x
← → ↻ ⓘ 127.0.0.1:5500/AJAX/실습4.html

1

sunt aut facere repellat provident occaecati
excepturi optio reprehenderit

quia et suscipit suscipit recusandae consequuntur
expedita et cum reprehenderit molestiae ut ut
quas totam nostrum rerum est autem sunt rem
eveniet architecto

2

qui est esse

est rerum tempore vitae sequi sint nihil
reprehenderit dolor beatae ea dolores neque
fugiat blanditiis voluptate porro vel nihil
molestiae ut reiciendis qui aperiam non debitis
possimus qui neque nisi nulla

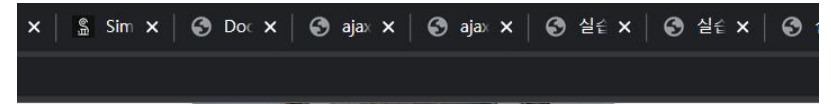
3

ea molestias quasi exercitationem repellat qui
ipsa sit aut

et iusto sed quo iure voluptatem occaecati omnis
eligendi aut ad voluptatem doloribus vel
accusantium quis pariatur molestiae porro eius
odio et labore et velit aut

실습 - 5

```
51 }
52 </style>
53
54 <script>
55   function moveCommentsPage(ArrayIndex) {
56     window.location.href = `실습5-movies.html?ArrayIndex=${ArrayIndex}`;
57   };
58
59   $(function () {
60
61     fetch('https://api.themoviedb.org/3/movie/upcoming?api_key=a64533e7ece6c72731da47c9c8bc691f&language=ko-KR&page=1')
62       .then(response => response.json())
63       .then(jsonArray => {
64
65         for (var ArrayIndex = 0; ArrayIndex < jsonArray.results.length; ArrayIndex++) {
66           let items = $("#items");
67           items.append(`<div class="card" onclick="moveCommentsPage(${ArrayIndex})">
68             <span class="container">
69               <br>
70               <h3 style="text-align:center; margin : 0 auto; padding: 3px;">${jsonArray.results[ArrayIndex].title}</h3>
71             </span>
72             </div><br><hr align="center"><br>`);
73         }
74       });
75
76   });
77
78   let currentHeight = 0;
79
80   setInterval(function () {
81     var pageBottom = document.documentElement.scrollHeight;
82
83     if (currentHeight < pageBottom) {
84       document.documentElement.scrollTop += 300;
85     }
86   }, 2000);
87
88 </script>
89
90 <audio src="../../assets/bgm.mp3" autoplay></audio>
91
92 <div id="items"></div>
93
94
95
96
```



실습 - 6

```
74
75   p #contents {
76     display: inline-block;
77     position: relative;
78     overflow: visible;
79     font-size: large;
80     padding: 3px;
81   }
82 </style>
83 <script>
84
85   function comeBack() {
86     window.location.href = "실습5.html";
87   }
88
89   $(function () {
90     const queryString = window.location.search;
91     const parameters = new URLSearchParams(queryString);
92     const ArrayIndex = parameters.get('ArrayIndex');
93
94     fetch('https://api.themoviedb.org/3/movie/upcoming?api_key=a64533e7ece6c72731da47c9c8bc691f&language=ko-KR&page=1')
95       .then(response => response.json())
96       .then(json => {
97         let comment = $('.comments')
98         comment.append(`<div class="card">
99           <div class="container">
100             <h3 style="text-align: left; font-size: 30px; margin: 10px; color: blue"><b>${json.results[ArrayIndex].title}</b></h3>
101             
102             <div id="openDay"><b>개봉일 : ${json.results[ArrayIndex].release_date}</b></div>
103             <div id="voteAverage"><b>✓ ${json.results[ArrayIndex].vote_average}</b></div>
104             <div id="voteCount"><b>☆ ${json.results[ArrayIndex].vote_count}</b></div>
105             <p id="contents" style="margin: 0 auto">${json.results[ArrayIndex].overview}</p>
106           </div>
107         </div>`)
108       })
109   });
110
111 </script>
112
113 <div class="comments"></div>
114 <br><button id="back" style="width: 70px; margin: 0 auto; display: block;" onclick="comeBack()">첫 화면</button>
115
116 </body>
117
118 </html>
```

미니언즈 2



개봉일 : 2022-05-30

✓ 0

☆ 0

1960년대 '그루'를 만나기 전 미니언들의 이야기를 그린
전편에 이어 1970년대를 배경으로 어린 '그루'를 만난 미
니언들의 한층 새롭고 흥미로운 모험을 그릴 예정

첫화면