WEB3.Ajax(Asynchronous JavaScript And XML)

reload없이 최신화

1.수업소개

추천검색어

한자한자 입력할때마다 클라이언트와 웹 서버와 Ajax로 통신함(reload X)

한동안 잠자는 사자로 살고 있었지만 5년 뒤 구글이 Gmail을 운영함으로써 꺠워줌

클라이언트에게는 좋은 경험을, 서버에게는 지출을 줄여줌

2.1 수업의 목적(1/2)

기존에는 페이지를 이동할 때 url을 변경해야 함?

공통된 부분은 안변하고 부분적으로만 변경되면 좋음

2.1 수업의 목적(2/2)

바꿔야 하는 부분과 고정되야 하는 부분을 구분

+바꿔야 하는 부분 유지보수가 쉬움

3.실습환경 준비

폴더 다운

4.동적으로 컨텐츠 변경하기

document.querySelector('article').innerHTML='<h2>HTML</h2>HTML is...'

# 5.0.fetch API-사용법

JS를 이용하여 웹서버에 접속하여 정보를 가져오는 방법

그 중 fetch API를 사용

fetch('html').then(function(response){ // 서버에게 html이라는 파일을 요청 response.text().then(function(text){ alert(text); }) })// 서버가 응답해 준 데이터가 text에 저장된다.

웹으로 접속 안해서 그런가 ? API를 못읽음ㅜㅜ

* <http://localhost/ajax/fetch.html>

모르면 넘어가 (쓰다보면 자유도가 높아짐)

# 5.1.fetch API-요청과 응답

fetch(‘html’): 브라우저야 서버에 html을 요청해줘

then: 그러고 나서 API한테 다음 함수를 실행시켜줘

# 5-2.fetch API-response 객체

function() : 익명함수 (이름이 없는 함수)

한번만 호출될거라 정의 x?

서버가 응답할 때 status 200: ok, 404: not found

response라는 객체로 응답

# 6.1.ajax의 적용

<li><a onclick="

      fetch('css').then(function(response){

        response.text().then(function(text){

          document.querySelector('article').innerHTML=text;

        })

      })

      ">CSS</a></li>

# 6.2.리팩토링 함수화

중복을 줄이자

# 7.1.초기 페이지 (1/2)

hash 기능?(북마크) (시작 페이지부분 설정 가능)

id="three"

로 진행하고

hash.html#three로 가면 북마크로 이동

<a href="#three">bookmark</a>

fragment identifier

location.hash

# 7.2.초기 페이지 (2/2)

#~~: 북마크 표시

보통 #!~ 으로 관습적으로 붙임

단점: 검색엔진 최적화가 잘 안됨

실제 내용이 없고 BE에서 동적으로 가져온거기 때문에…

현실적으로는 hash #를 사용하지 않고 pjax를 사용

# 8.1.글목록

데이터 , 로직, 애플리케이션

데이터를 저장하는 부분을 list로 따로 뺌

fetch('list').then(function(response){

          response.text().then(function(text){

            document.querySelector('#nav').innerHTML=text;

          })

        })

# 8.2.글목록 (2/2)

반복되는 부분을 array로 선언

split

‘나 “를 문자처럼 쓰려면\앞에 써야함

fetch(‘aa’)

이제 list파일만 관리하면 됨

띄어쓰기나 공백떄문에 에러가 날떄

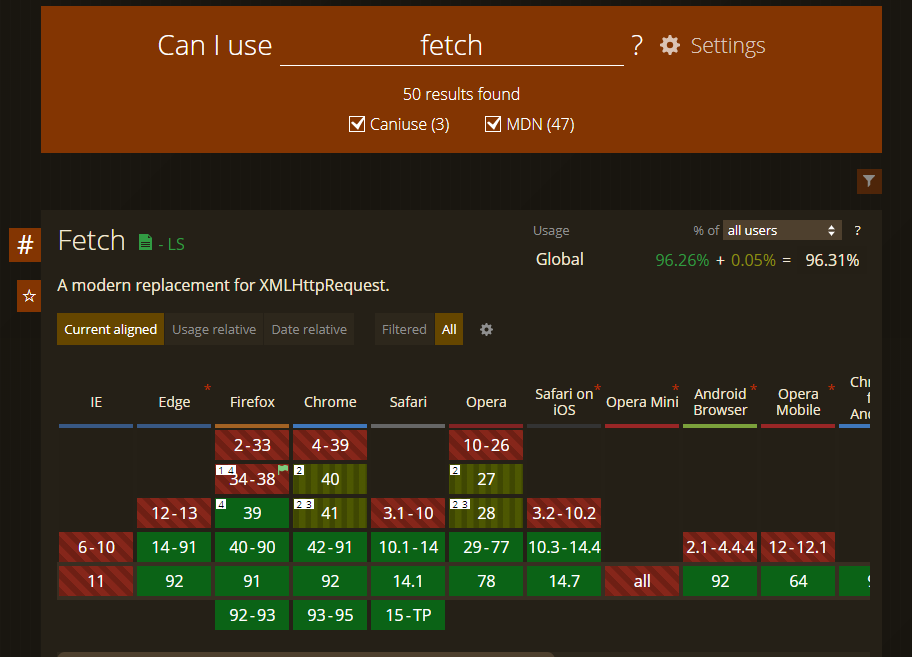
.trim( )

# 9.fetch API polyfill(호환성 확보)

fetch api는 네트워크 통신을 포함한 리소스 획득을 위한 인터페이스가 정의되어 있습니다.

<https://caniuse.com/>

내가 이 기능을 특정브라우저에서 쓸 수 있는지



옛날 브라우저에는 지원 x ->polyfill을 사용하여 지원하게 해줌

polyfill은 기본적으로 지원하지 않는 이전 브라우저에서 최신 기능을 제공하는 데 필요한 코드 (일반적으로 웹의 JavaScript)입니다.

브라우저에서는 지원안해줘서 그 브라우저에 돌아가게끔 함(유저가?)

내가 만든 html에서 src=fetch.js를 선언해줘서

익스프롤러11 유저들도 쓸 수 있게끔 수정 가능(호환성 확보)

# 10.수업을 마치며

페이지 전환 없이도 서버에서 일부데이터를 가져와서 동적으로 변환 가능

웹은 다른 플랫폼과 달리 검색이 되면서 어플리케이션으로 동작한다

공부할 만한 것들

https://www.youtube.com/watch?v=55FrHTNjTCc

1. XML(eXtensible Markup Language)은 [W3C](https://ko.wikipedia.org/wiki/W3C)에서 개발된, 다른 특수한 목적을 갖는 [마크업 언어](https://ko.wikipedia.org/wiki/%EB%A7%88%ED%81%AC%EC%97%85_%EC%96%B8%EC%96%B4)를 만드는데 사용하도록 권장하는 다목적 [마크업 언어](https://ko.wikipedia.org/wiki/%EB%A7%88%ED%81%AC%EC%97%85_%EC%96%B8%EC%96%B4)이다.)

,만으로는 구분하기 힘들 때

XML과의 경쟁구도 JSON : but 완전 커버는 안됨?

정보를 주고받을때의 형식 구조를 약속한 것?

컴퓨터들은 테이블같은거 없이 한줄짜리 코드로 쭉 있기 떄문에 주고받으며 알아보기 어려움

xml은 태그<>를 사용

<https://www.webtoolkitonline.com/xml-minifier.html> 로 한줄화

<https://www.freeformatter.com/xml-formatter.html> 로 다줄화

or beautifier

XSD문서 형식이라는 스키마를 통해 문법 검증화 가능

XSD<->XML 서로 호환 가능

2. json은 {},[] , 로 구분

but 문법에 취약화, 문법틀리면 동작 X,

주석 사용이 불가능?

### XSD 개요

#### XSD란?

XSD는 XML 스키마 정의(XML Schema Definition)를 의미합니다.

XSD는 XML 문서의 구조 및 해당 문서가 포함할 수 있는 적법한 요소와 속성을 명시합니다.

안정성 중시:xml

가벼움 중시:json

3. YAML:사람이 보기 좋게 작성

JSON -> YAML

(SPA)Single Page Application: **싱글 페이지 애플리케이션**(single-page application, **SPA**, 스파)은 서버로부터 완전한 새로운 페이지를 불러오지 않고 현재의 페이지를 동적으로 다시 작성함으로써 사용자와 소통하는 [웹 애플리케이션](https://ko.wikipedia.org/wiki/%EC%9B%B9_%EC%95%A0%ED%94%8C%EB%A6%AC%EC%BC%80%EC%9D%B4%EC%85%98)이나 [웹사이트](https://ko.wikipedia.org/wiki/%EC%9B%B9%EC%82%AC%EC%9D%B4%ED%8A%B8)를 말한다.

PJAX

PWA(Progressive Web Apps)=SPA+오프라인에서 동작가능

online+offline