



UI/UX



05



웹 서버, 웹 애플리케이션 서버

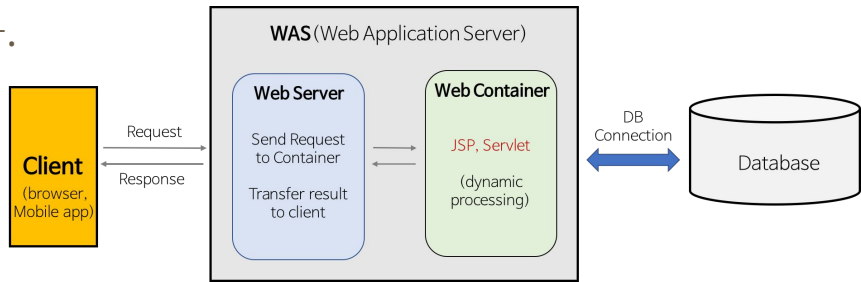
클라이언트가 데이터나 서비스 요청을 보내오면 서버가 그 요청을 처리하고 응답을 보내준다.

웹 서버 (WebServer) 의 특징

- 웹 서버는 정적인 콘텐츠를 생성한다.

웹 애플리케이션 서버(Web Application Server) 특징

- WAS라고 불린다.
- 웹서버를 포함한다.
- 웹 컨테이너가 포함되어있으며, 동적 데이터를 처리할 수 있다.
- ASP, PHP, JSP 등 개발 언어를 읽고 처리할 수 있다.



window.onload

- 페이지가 완전히 로드 되기 전에 요소에 접근하거나 변경하려 하면 자바스크립트 에러가 발생하거나 여러분이 작성한 코드가 작동하지 않는다.
- 페이지가 완전히 로드 된 후에 JavaScript가 실행을 시작하게 하는 것이 의미가 있으며, **브라우저가 페이지 로드를 완료한 뒤 함수를 실행하려면 window.onload 속성에 해당 함수를 할당 한다.**

할당방법

- **window.onload = init; function init(){}**

JS- 01

indexOf - 특정문자 위치 찾기, index값

"Hello, welcome!".indexOf("Hello")

"Hello, welcome!".indexOf("wel")

"Hello, welcome!".indexOf("bye")

```
> "Hello, welcome!".indexOf("Hello")
```

```
< 0
```

```
> "Hello, welcome!".indexOf("wel")
```

```
< 7
```

```
> "Hello, welcome!".indexOf("bye")
```

```
< -1
```

JS- 02

document.getElementById("demo")

- phrase이 id값인 요소

document.getElementById("demo").innerHTML

- 그 요소 안에 html 요소 등록가능

```
1  <!doctype html>
2  <html lang="ko">
3
4  <head>
5      <title>강아지</title>
6      <meta charset="utf-8">
7  </head>
8
9  <body>
10     <p id="soundslike">
11     </p>
12     <script>
13
14         var walksLike = "duck";
15         //var walksLike = "dog";
16         //var walksLike = "sleepdog";
17
18         var soundsLike = document.getElementById("soundslike");
19         //document : 해당 페이지
20         //getElementById(id명) : 태그에 있는 id 속성을 사용하여 해당 태그에 접근
21         // 반환값 : 주어진 id와 일치하는 dom 요소를 나타내는 Element 객체를 반환
22         //          만약 주어진 id와 일치하는 dom 요소가 없으면 null을 반환
23
24         if (walksLike == "dog") {
25             soundsLike.innerHTML = "멍! 멍!";
26             //해당 요소의 HTML 값을 변경할 수 있음.
27         } else if (walksLike == "duck") {
28             soundsLike.innerHTML = "꽹, 꽹";
29         } else {
30             soundsLike.innerHTML = "조용~";
31         }
32     </script>
33 </body>
34
35 </html>
```

JS - 03

- `rand1 = Math.floor(Math.random() * words1.length);`
`Math.random()` : 0~1 사이의 수 [1은 포함하지 않음]
- `Math.floor()` : 소수점 버림
- `words1.length` : `words1`의 요소 개수
- 즉, `rand1`의 값은 0 ~ (`words1`의 요소개수-1) 사이의 값

JS - 03

```
<script>
  let guess = prompt("무슨색인지 맞춰봅시다.(빨,노,파)");
  let answer= checkGuess(guess);
  alert(answer);
  function checkGuess(guess)
  {
    let answers=["빨","노","파"];
    let index=Math.floor(Math.random()*answers.length);
    if(guess == answers[index])
    {
      return "맞췄습니다."
    }else
    {
      return "아쉽게도 정답은 "+answers[index]+"입니다.";
    }
  }
}
```

JS - 04

버튼 클릭 이벤트
등록

```
//페이지 로딩후 실행될 수 있도록  
window.onload = init;
```

```
function init() {  
    //곡 추가버튼  
    let button = document.getElementById("addButton");  
    button.onclick = handleAddButtonClick;  
  
    //곡 삭제버튼  
    let button1 = document.getElementById("deleteButton");  
    button1.onclick = function (e){  
        removeAll();  
    };  
  
    //저장되어있는 곡이 있다면 불러옴  
    loadPlaylist();  
}
```

```
function handleAddButtonClick(e) {
```


JS - 05

생성자

사용법

```
<title>생성자 TEST</title>
<script>
  function stud(name,score,grade)
  {
    this.name=name;
    this.score=score;
    this.grade=grade;
    this.comment=function(){
      alert(`${this.name} 학생의 점수는 ${this.score}이고 등급은 ${this.grade} 입니다.`);
    };
  }
  let stud1=new stud("김하나",98,"A");
  let stud2=new stud("이하나",35,"F");
  let stud3=new stud("박하나",87,"B");
  let stud4=new stud("최하나",77,"C");

  stud1.comment();
  stud2.comment();
  stud3.comment();
  stud4.comment();
</script>
```

JS - 06

indexOf와

length

```
function searchForSong(word) {  
    word = word.toUpperCase();  
    let song;  
    for (let i = 0; i < songs.length; i++) {  
        song = songs[i].toUpperCase();  
        if (song.indexOf(word) >= 0) {  
            return song;  
        }  
    }  
    return "없는 곡입니다.";  
}
```

window 객체

- window 객체는 global object 이다.
 - 생략 가능하다. => window.alert("사용가능") , alert("사용가능")
 - 프로퍼티 : location, status, onload, document
 - 메서드 : alert, prompt, open, close, setTimeout, setInterval
-
- document 객체는 DOM을 접근(access)하는데 사용된다.
 - document 객체는 window 객체의 프로퍼티이다.
 - window.document 와 document 는 같다.
 - 프로퍼티 : domain, title, URL
 - 메서드 : getElementById, getElementsByTagName, getElementsByClassName, createElement

```
> window.document === document  
< true
```

로컬스토리지

- 데이터를 저장하는 장소 중 하나
- 데이터의 값은 항상 **문자**로 저장
- **localStorage.setItem("name","김윤지")** : 키,값 쌍으로 보관
- **localStorage.getItem("name")** : 해당하는 키의 값을 받아옴
- **localStorage.removeItem("name")** : 키 값을 삭제한다.
- **localStorage["name"]** = "김윤지"
- **localStorage["name"]**
- **localStorage.clear()** : LocalStorage에 저장된 모든 것을 삭제한다.

JSON

`JSON.stringify(playlistArray)`

- JavaScript 객체를 JSON 문자열로 변환

`JSON.parse(playlistArray)`

- JSON 문자열을 JavaScript 객체로 변환

```

function handleAddButtonClick(e) {
  let songName = document.getElementById("songTextInput").value;
  //alert("Adding " + songName);

  if(songName==""){
    alert("곡을 입력하세요.");
  }else{
    //localStorage에 넣기
    let playlistArray = getStoreArray("playlist");
    playlistArray.push(songName);
    localStorage.setItem("playlist", JSON.stringify(playlistArray));

    //addSongList([songName]); //함수변경
    let ul=document.getElementById("playlist")
    let li = document.createElement("li");
    li.innerHTML = songName;
    ul.appendChild(li);
  }
}

```

ul.appendChild(li);

createElement("li") : li 태그 요소 생성

li.innerHTML : 생성된 요소안에 내용 넣기

ul.appendChild(li) : ul요소 안에 만든 li요소
마지막에 넣기(+=)

```
function getStoreArray(key) {  
    let playlistArray = localStorage.getItem(key);  
    if (playlistArray == null || playlistArray == "") {  
        playlistArray = new Array();  
    }  
    else {  
        playlistArray = JSON.parse(playlistArray);  
    }  
    return playlistArray;  
}
```

```
function removeAll(){  
    if(confirm('모두 지울까요?')){  
        localStorage.clear();  
        let ul = document.getElementById("playlist");  
        ul.innerHTML = "";  
    }  
}
```

부트스트랩

버튼 색 지정 :

```
<button type="button" class="btn btn-primary">Primary</button>  
<button type="button" class="btn btn-secondary">Secondary</button>  
<button type="button" class="btn btn-success">Success</button>  
<button type="button" class="btn btn-danger">Danger</button>  
<button type="button" class="btn btn-warning">Warning</button>  
<button type="button" class="btn btn-info">Info</button>  
<button type="button" class="btn btn-light">Light</button>  
<button type="button" class="btn btn-dark">Dark</button>
```

버튼 -> btn

버튼 색 지정 -> btn-*

부트스트랩

container

row , col

large:lg, small:sm 등

row당 : col의 합계 12

```
<div class="container text-center">
  <!-- Stack the columns on mobile by making one full-width and the other half-width -->
  <div class="row">
    <div class="col-md-8">.col-md-8</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
  </div>

  <!-- Columns start at 50% wide on mobile and bump up to 33.3% wide on desktop -->
  <div class="row">
    <div class="col-6 col-md-4 col-sm-3">abc</div>
    <div class="col-6 col-md-4 col-sm-3">abc</div>
    <div class="col-6 col-md-4 col-sm-6">abc</div>
  </div>

  <!-- Columns are always 50% wide, on mobile and desktop -->
  <div class="row">
    <div class="col-6">.col-6</div>
    <div class="col-6">.col-6</div>
  </div>
</div>
```

부트스트랩

탭 추가 예제보기

id 와 data-bs-target 의 관계 알아두기

카카오 맵 - 01

- api키 발급받아야 함
- 지도를 담기 위한 영역을 만들 때 크기 지정
- src에 // 라는 상대 프로토콜을 사용하여 환경에 따라 자동으로 해당 프로토콜을 따라가게 할 수 있다.
- 플랫폼 등록과 애플리케이션등록 이후에 사용이 가능하다.
- API키를 발급하여 자신의 키로 사용하여야 한다.
- **options** 의 속성 중 **center** 값은 지도를 생성하는 데 반드시 필요하다
- 그 center 값은 LatLng 타입이다.
- **new kakao.maps.Map()**, 등을 통하여 맵을 생성할 수 있다.
- **new kakao.maps.LatLng(37.54699, 127.09598)** : (위도,경도) 순서

```
// 지도가 이동, 확대, 축소로 인해 중심좌표가 변경되면 마지막 파라미터로 넘어온 함수를 호출하도록 이벤트를 등록함
kakao.maps.event.addListener(map, 'tilesloaded', function () {
    marker.setPosition(map.getCenter());
});

// 지도에 클릭 이벤트를 등록합니다
// 지도를 클릭하면 마지막 파라미터로 넘어온 함수를 호출합니다
kakao.maps.event.addListener(map, 'click', function (mouseEvent) {
    marker.setPosition(mouseEvent.latLng);
});

</script>
```

```
window.onload = getMyLocation;
function getMyLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(
            displayLocation,
            displayError);
    }
    else {
        alert("이런, 지오로케이션이 제공되지 않네요");
    }
}

function displayLocation(position) {
    var latitude = position.coords.latitude;
    var longitude = position.coords.longitude;

    var div = document.getElementById("location");
    div.innerHTML = "당신은 위도: " + latitude + ", 경도: " + longitude + "에 있습니다";
}
```