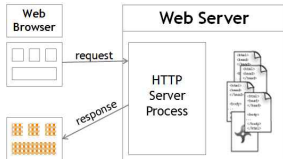


UI/UX - 01

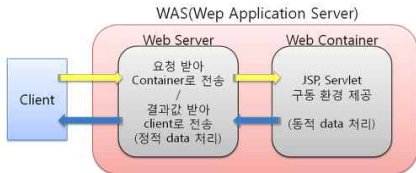
V1.0

웹이 동작하는 과정

- 웹 서버(Web Server)
 - 클라이언트의 요청을 받아 HTML이나 오브젝트를 **HTTP 프로토콜**을 이용해 전송하는 것이다.
 - 클라이언트가 요청을 보내오면 그 요청(request)을 처리하고 그 결과를 **사용자에게 답변(response)으로 보내준다.**
 - 웹 서버만 구축된 서버는 웹 페이지, 이미지 등 정적인 페이지를 생성하지만, servlet 컨테이너가 탑재되어 있는 WAS(Web Application Server)는 JSP 페이지를 구동할 수 있게 함으로써 동적인 페이지를 생성할 수 있다.
 - 웹 서버는 웹 문서를, WAS는 JSP 페이지 등을 양분하여 서버 부담을 줄이는 것이 가능하다.
 - 웹서버 : Apache, IIS (Internet Information Server) 등



웹 애플리케이션 서버 - WAS



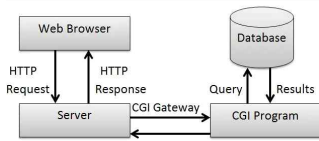
- 웹서버 + 웹컨테이너 로 웹 상에서 사용하는 컴포넌트를 올려놓고 사용하게 되는 서버이다.
 - **미들웨어**(소프트웨어 엔진)
 - 웹컨테이너에서 JSP와 Servlet을 실행시킬 수 있는 SW를 웹 컨테이너 혹은 서블릿 컨테이너라고 한다.
 - 웹 서버에서 JSP를 요청하면 톰캣에서 JSP파일을 서블릿으로 변환하여 컴파일을 수행하고, 서블릿 수행결과를 웹서버에 전달하게 된다.
 - 따라서 동적인 페이지를 생성할 수 있다.
 - 흔히 **apache**는 **web Server** , **tomcat**은 **WAS**라고 말한다.
 - Web서버는 HTML,CSS,IMG 같은 정적 콘텐츠를 처리하고(HTTP 프로토콜을 통해 읽힐 수 있는 문서),
 - WAS서버는 ASP,PHP,JSP 등 개발 언어를 읽고 처리하여 동적 콘텐츠를 처리한다.

WEB SERVER와 WAS의 분리 이유

- **WAS 와 WebServer를 분리하지 않은 경우**
 - 모든 콘텐츠를 한 곳에 집중시켜 웹서버와 WAS의 역할을 동시에 수행하게 된다. 스위치를 통한 로드 밸런싱, 사용자가 적을 경우 효과적이다.
- **분리한 경우**
 - 기능적 분류를 통해 효과적인 분산을 유도한다.
 - 정적인 데이터는 웹서버에서 동적인 데이터는 WAS가 처리하게 된다.
- **WAS를 여러 개로 분리하기도 한다.**
 - 로직별로 구분하여 특정 로직의 부하에 따라 적절한 대응을 할 수 있게 되지만 설계단계, 유지보수가 복잡해질 수 있다.
- **WAS와 WebServer를 분리하는 이유**
 - 기능을 분리하여 서버의 부하방지
 - 물리적으로 분리하여 보안을 강화
 - 여러대의 WAS를 연결가능(로드밸런싱 역할, fail over, fail back 처리에 유리)
 - 여러 웹어플리케이션을 서비스 가능(java,c#,php 서버 등 하나의 웹서비스를 통해 서비스 가능하다)

웹 응용프로그램

- 동적인(dynamic) 페이지 반환을 위한 노력
 - 클라이언트(웹 브라우저) 측
 - JavaScript(interpreter in browser)를 이용
 - AJAX (Asynchronous(비동기) JavaScript and XML)
 - » 웹 서버와 비동기적으로 데이터 교환 (JSON, JSON-RPC)
 - » 전체페이지를 reload 하지 않고 변경된 데이터 부분만 refresh하는 기법
 - 서버 측:
 - CGI(Common Gateway Interface) 프로그램 - 웹서버와 DB 사이
 - » HTTP 요청으로부터 매개변수들을 받아들일 수 있으며, 마치 저장된 페이지가 있는 것처럼 데이터를 반환할 수 있는 프로그램
 - » 웹서버의 기능으로는 필요한 데이터를 가져오기 힘들어 필요한 데이터를 가져올 수 있는 별도의 프로그램을 작성
 - » 1단계 : C, Fortran, Unix Shell Script, Perl, Visual Basic, ...
 - » 2단계(script) : PHP, ASP, JSP, ... 쉬운 언어로 제공, 서버에서 동작하는 스크립트 언어



HTML5

- **Markup + JavaScript APIs + CSS =>HTML5**
- **HTML5 JavaScript APIs**
 - Canvas
 - Video
 - Audio
 - **Geolocation**
 - Local Storage
 - Drag & Drop
 - Sockets
 - Forms
 - Web Workers

ex01.html

- 기본적인 dom 구조
- Document
- .getElementById("id")
- innerHTML

```
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>
```

ex02.html

```
1  <!doctype html>
2  <html lang="kr">
3
4  <head>
5      <title>행성</title>
6      <meta charset="utf-8">
7      <script>
8          var planet = document.getElementById("greenplanet");
9          planet.innerHTML = "적색경보: 광선총에 맞았음!";
10     </script>
11 </head>
12
13 <body>
14     <h1>녹색 행성</h1>
15     <p id="greenplanet">모든 것이 순조로움</p>
16     <h1>붉은 행성</h1>
17     <p id="redplanet">이상 무</p>
18     <h1>푸른 행성</h1>
19     <p id="blueplanet">전 시스템 완벽함</p>
20 </body>
21
22 </html>
```

녹색 행성

모든 것이 순조로움

붉은 행성

이상 무

푸른 행성

전 시스템 완벽함

왜 변화가 없을까?

- 페이지가 완전히 로드되지 않았을 때, JavaScript 코드가 실행된다면 아직 DOM 역시 완전하게 생성되지 않았을 것이다.
- DOM이 생성되지 않았다면 `<p id="greenplanet">` 요소 역시 아직 **존재하지 않는다**.
- 따라서, 페이지가 완전히 로드 되기 전에 요소에 접근하거나 변경하려 하면 자바스크립트 에러가 발생하거나 여러분이 작성한 코드가 작동하지 않는다.
- 따라서 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.**

ex02-1.html

```
1 <!doctype html>
2 <html lang="kr">
3
4 <head>
5   <title>행성</title>
6   <meta charset="utf-8">
7   <script>
8     window.onload = function(){
9       var planet = document.getElementById("greenplanet");
10      planet.innerHTML = "적색경보: 광선총에 맞았음!!";
11    }
12  </script>
13 </head>
14
15 <body>
16   <h1>녹색 행성</h1>
17   <p id="greenplanet">모든 것이 순조로움</p>
18   <h1>붉은 행성</h1>
19   <p id="redplanet">이상 무</p>
20   <h1>푸른 행성</h1>
21   <p id="blueplanet">전 시스템 완벽함</p>
22 </body>
23
24 </html>
```

녹색 행성

적색경보: 광선총에 맞았음!!

붉은 행성

이상 무

푸른 행성

전 시스템 완벽함

ex03.html

배열을 사용하여
구절 자동 생성

- 배열
- Math.floor()
- Math.random()

```
1  <!doctype html>
2  <html>
3  <head>
4      <title>Phrase-o-matic (구절 자동 생성기)</title>
5      <meta charset="utf-8">
6      <style>
7          body {
8              font-family: Verdana, Helvetica, sans-serif;
9          }
10     </style>
11     <script>
12         function makePhrases() {
13             var words1 = ["노랑고", "유난히", "그래서", "어제 받은", "군침 도는"];
14             var words2 = ["신선한", "맛이 간", "시원한", "커다란", "의문의"];
15             var words3 = ["우유", "오렌지 주스", "바나나", "마늘빵", "치킨과 맥주"];
16             //Math.random()
17             var rand1 = Math.floor(Math.random() * words1.length);
18             var rand2 = Math.floor(Math.random() * words2.length);
19             var rand3 = Math.floor(Math.random() * words3.length);
20             var phrase = words1[rand1] + " " + words2[rand2]
21             + " " + words3[rand3];
22             var phraseElement = document.getElementById("phrase");
23             //id 가 phrase인 <p> 요소를 가져옵니다.
24             phraseElement.innerHTML = phrase;
25             //<p>요소의 내용에 이 구문을 할당합니다.
26         }
27         window.onload = makePhrases;
28     </script>
29 </head>
30 <body>
31     <h1>구절 생성기 :</h1>
32     <p id="phrase"></p>
33 </body>
34 </html>
```

Math.floor(), Math.random()

- `rand1 = Math.floor(Math.random() * words1.length);`
Math.random() : 0~1사이의 [1은 포함하지 않음]
- Math.floor() : 소수점 버림
- 즉, 수의 범위는 0 ~ (곱해지는 수-1) 가 된다.

	A	B	C	D
1	Math.random	words.length	Math.random*words.length	Math.floor
2	0	5	0	0
3	0.3	5	1.5	1
4	0.6	5	3	3
5	0.9999	5	4.9995	4

```

1  <!doctype html>
2  <html lang="kr">
3  <head>
4      <title>기온</title>
5      <meta charset="utf-8">
6      <script>
7          window.onload = showTemps;
8          function showTemps() {
9              var tempByHour = [59.2, 60.1, 63, 65, 62];
10             for (var i = 0; i < tempByHour.length; i++) {
11                 var theTemp = tempByHour[i];
12                 var id = "temp" + i; //id=temp0 ~ id=temp4
13                 var li = document.getElementById(id);
14                 if (i == 0) {
15                     li.innerHTML = "정오의 기온은 " + theTemp +
16                         "도였습니다.";
17                 } else {
18                     li.innerHTML = "오후 " + i + "시의 기온은 " +
19                         theTemp + "도였습니다.";
20                 }
21             }
22         }
23     </script>
24 </head>
25 <body>
26     <h1>기온</h1>
27     <ul>
28         <li id="temp0"></li> <li id="temp1"></li>
29         <li id="temp2"></li> <li id="temp3"></li>
30         <li id="temp4"></li>
31     </ul>
32 </body>
33 </html>

```

ex04.html

배열과 반복문 사용

기온

- 정오의 기온은 59.2도였습니다.
- 오후 1시의 기온은 60.1도였습니다.
- 오후 2시의 기온은 63도였습니다.
- 오후 3시의 기온은 65도였습니다.
- 오후 4시의 기온은 62도였습니다.

해보기

ex04-1.html

- 평균기온 추가하기

기온

- 정오의 기온은 59.2도였습니다.
- 오후 1시의 기온은 60.1도였습니다.
- 오후 2시의 기온은 63도였습니다.
- 오후 3시의 기온은 65도였습니다.
- 오후 4시의 기온은 62도였습니다.

평균 기온은 61.86입니다.

```

1  <!doctype html>
2  <html lang="kr">
3  <head>
4      <title>기온</title>
5      <meta charset="utf-8">
6      <script>
7          window.onload = showTemps;
8          function showTemps() {
9              var tempByHour = [59.2, 60.1, 63, 65, 62];
10             sumTemp = 0;
11             for (var i = 0; i < tempByHour.length; i++) {
12                 var theTemp = tempByHour[i];
13                 var id = "temp" + i; //id=temp0 ~ id=temp4
14                 var li = document.getElementById(id);
15                 if (i == 0) {
16                     li.innerHTML = "점오의 기온은 " + theTemp +
17                         "도 였습니다.";
18                 } else {
19                     li.innerHTML = "오후 " + i + "시의 기온은 " +
20                         theTemp + "도 였습니다.";
21                 }
22                 sumTemp += theTemp;
23             }
24             var avg = document.getElementById("avg");
25             avg.innerHTML = "평균 기온은 " + sumTemp/tempByHour.length + "입니다.";
26         }
27     </script>
28 </head>
29 <body>
30     <h1>기온</h1>
31     <ul>
32         <li id="temp0"></li> <li id="temp1"></li>
33         <li id="temp2"></li> <li id="temp3"></li>
34         <li id="temp4"></li>
35     </ul>
36     <p id="avg"></p>
37 </body>
38 </html>

```

자판기 만들어 보기

이 페이지에는

생성된 코인 수 : 2

확인

이 페이지에는

생성된 캔 수 : 5

확인

자판기

5 캔 남은 사이다

하나를 뽑아서 나눠주면,

자판기에 사이다가 4캔 있습니다.

4 캔 남은 사이다

하나를 뽑아서 나눠주면,

자판기에 사이다가 3캔 있습니다.

남은 사이다개수 : 3, 코인 개수 : 0

alert 를 이용하여 생성된 코인수와 캔의 수를 나타내기.

생성되는 코인과 캔의 수는 랜덤으로 0~9의 수가 된다.

자판기의 캔의 수와 코인이 있다면 둘 중 하나가 없어질 때까지

자판기에서 사이다를 뺀다.

최종 사이다와 코인의 수는 몇 개가 남았는지 마지막에 알려준다.