**1. 웹 프로그래밍 기초**

**1-2 HTTP 프로토콜 이해**

- HTTP ? 서버와 클라이언트가 인터넷상에서 데이터를 주고 받기 위한 프로토콜

- 텍스트, 이미지, 동영상 등 어떠한 형태의 데이터도 전달 가능

- 작동방식 : 클라이언트 -> 서버 (요청)

서버 -> 클라이언트 (응답)

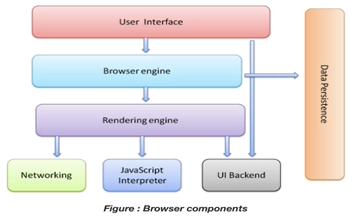
- Stateless 방식 : 클라이언트가 요청을 하면 서버와 연결되고 서버는 응답 후 즉시 연결을 끊게 되는데, 다음에 다시 그 클라이언트가 요청을 하면 서버는 그 클라이언트가 아까 클라이언트인지 모른다 -> stateless (무상태) -> 불특정 다수 대상 서비스에 적합

이렇게 정보유지가 어려우므로 ‘쿠키’를 사용

- URL ( uniform resource locator )

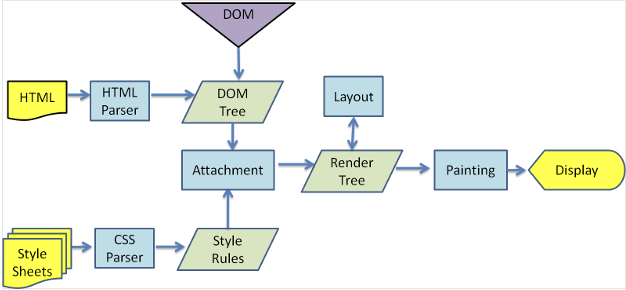


**1-4 브라우저 동작**



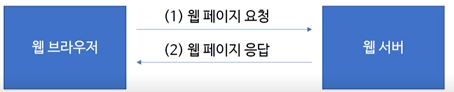
* 브라우저 엔진 : 브라우저 SW를 동작 시켜주는 핵심 엔진
* 네트워크 : HTTP 를 통해서 통신
* 렌더링 엔진 : requested content를 보여주는 역할.

Ex) HTML이 요청되면 렌더링 엔진이 HTML, CSS를 parsing하여 화면에 보여주도록 하는 것



**1-6 웹서버**

- 웹 서버의 가장 중요한 기능은 client가 요청하는 HTML문서나 각종 resouce를 전달하는 것

- 

Client 와 web sever 간에 HTTP를 통해 통신

**1-7 WAS**

* 서버 : 서비스를 제공 ex) 웹 서버
* 클라이언트 : 서버에게 서비스를 요청해서 결과를 보여주는 것 ex) 웹 브라우저

미들웨어 : 클라이언트와 DBMS가 직접적으로 연결되면 여러가지 문제점 발생

* 클라이언트와 DBMS사이에 또 다른 서버를 둠

- WAS : 브라우저와 DBMS사이에서 동작하는 미들웨어, 동적 컨텐츠 처리

1. 프로그램 실행환경과 데이터베이스 접속 기능을 제공

2. 여러 개의 트랜잭션을 관리

3. 업무 처리하는 비즈니스 로직을 수행

- WAS만 있어도 정적/동적 컨텐츠 모두 처리 가능한데, 웹 서버를 사용하는 이유?

웹 서버는 상대적으로 간단한 구조이고, 만일 WAS에 문제가 발생하게 된다면, WAS 앞 단에 위치한 웹서버가 동작함으로써 장애 극복 기능을 하기 위함 -> failover

**3.CSS**

**3-1 CSS 선언방법**

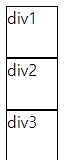
- HTML 페이지에 style 적용하는 방법 3가지

- inline : HTML 태그 안에다가 넣는 방법, 최우선 적용

- internal : head안에 선언 <style> ~~ </style>

- external : 외부 .css 파일 지정

**3-5 Element가 배치되는 방법**

div가 3개 있으면 이렇게 상하로 나타나지는데, display : inline 속성을 통해

 이렇게 가로로 나타낼 수 있다.

- margin : 다른 영역들과의 간격

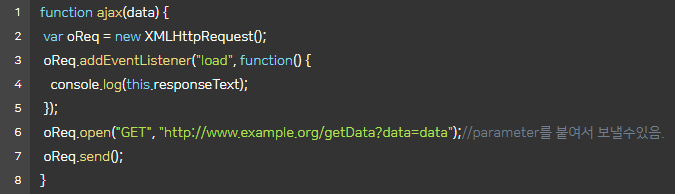
- padding : 같은 영역안에서의 간격

**2. DB연결 웹 앱**

**2-4 Ajax통신 ( XMLHTTPRequest 통신)**

-비동기로 서버로부터 데이터를 가져오는 것,,, 페이지를 리로드할 때 전체 페이지를 리로드 하지 않고, 필요한 부분만 리로드하여 자원과 시간측면에서 장점

-html 페이지 전체가 아닌 일부분만 갱신할 수 있도록 XMLHTTPRequest객체를 통해 서버에 request하고, Json 이나 xml형태로 데이터를 받아 갱신한다.



-ajax를 보내기 위해 XMLHTTPRequest객체 생성

-open()을 통해 요청을 준비하고 send()를 통해 서버로 보낸다 -> ajax 함수 종료

-> 콜백함수 실행

**3-1 JSP**

- 서블릿에서 out.print() 와 같은 기능을 JSP에서 <%= %> 이 한다..

- jsp는 서블릿 기술을 활용하고 서버에서 구동될 때 항상 서블릿으로 바뀐다..

바뀐 서블릿으로 실행이 된다

JSP 실행순서

1. 브라우저가 웹 서버에 JSP에 대한 요청 정보 전달

- 최초로 요청했을 경우

1. JSP코드 -> 서블릿 코드로 변환(java파일 생성)

2. 서블릿 코드를 컴파일 해서 bytecode로 변환 (class 파일 생성)

3. 서블릿 클래스를 로딩하고 인스턴스 생성

2. 서블릿이 실행되어 요청 처리하고 응답정보 생성

4-3 JSP와 서블릿 연동

* Servlet은 로직 수행에 유리(자바 파일이므로)
* JSP는 결과 출력이 유리 (html파일 이므로)
* Servlet에서 로직 수행 결과를 JSP에 포워딩

- 포워딩 할 때 request.setAttribute(“ ”, 변수명); 사용

- RequestDispatcher객체 request.getRequestDispatch(“/~~.jsp") 생성 하여 보내기

- jsp 에서 받을 때는 request.getAttribute() 통해서 받아오기