인터넷 프로그래밍

**1. Internet**

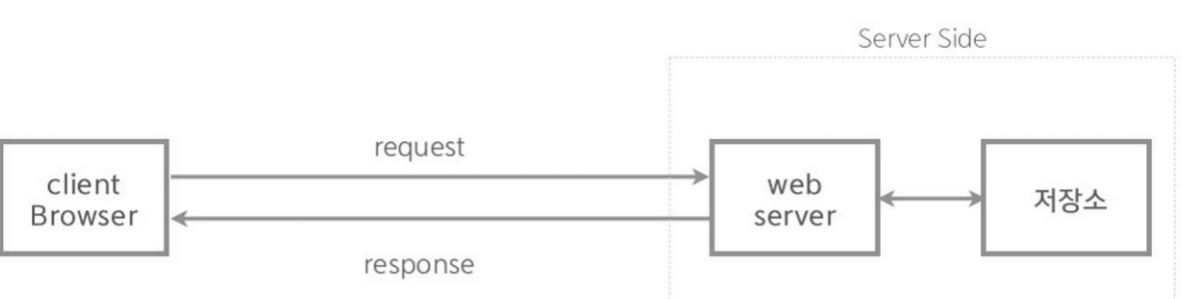
- 전 세계에 걸쳐 원거리 접속이나 파일 전송, 전자 메일 등의 데이터 통신 서비스를 받을 수 있는 컴퓨터 네트워크의 시스템

**2. Web**

- 인터넷에 연결된 컴퓨터들을 통해 사람들이 정보를 공유할 수 있는 전세계적인 정보 공간

- 인터넷에서 HTTP 프로토콜, 하이퍼텍스트, HTML 형식 등을 사용하여 그림과 문자, 멀티미디어를 교환하는 전송 방식

- Web의 흐름



1) 특정 **URI**로 서버에 접근하여 resource를 요청

2) Server에서는 필요한 정보가 있는지 서버의 저장소를 통해 확인

3) 저장소에서 추출한 정보를 Server가 받아서 client에게 보내줌

4) Client는 Server가 보내준 정보를 받음

5) 사용자가 쉽게 알아볼 수 있도록 정해진 UI의 형태에 맞게 정보를 보여줌

**URI**?

- 통합자원식별자(Uniform Resource Identifier)

- 인터넷에 있는 자원을 나타내는 유일한 주소

ex) http://www.naver.com (http프로토콜임을 명시하고 있음)

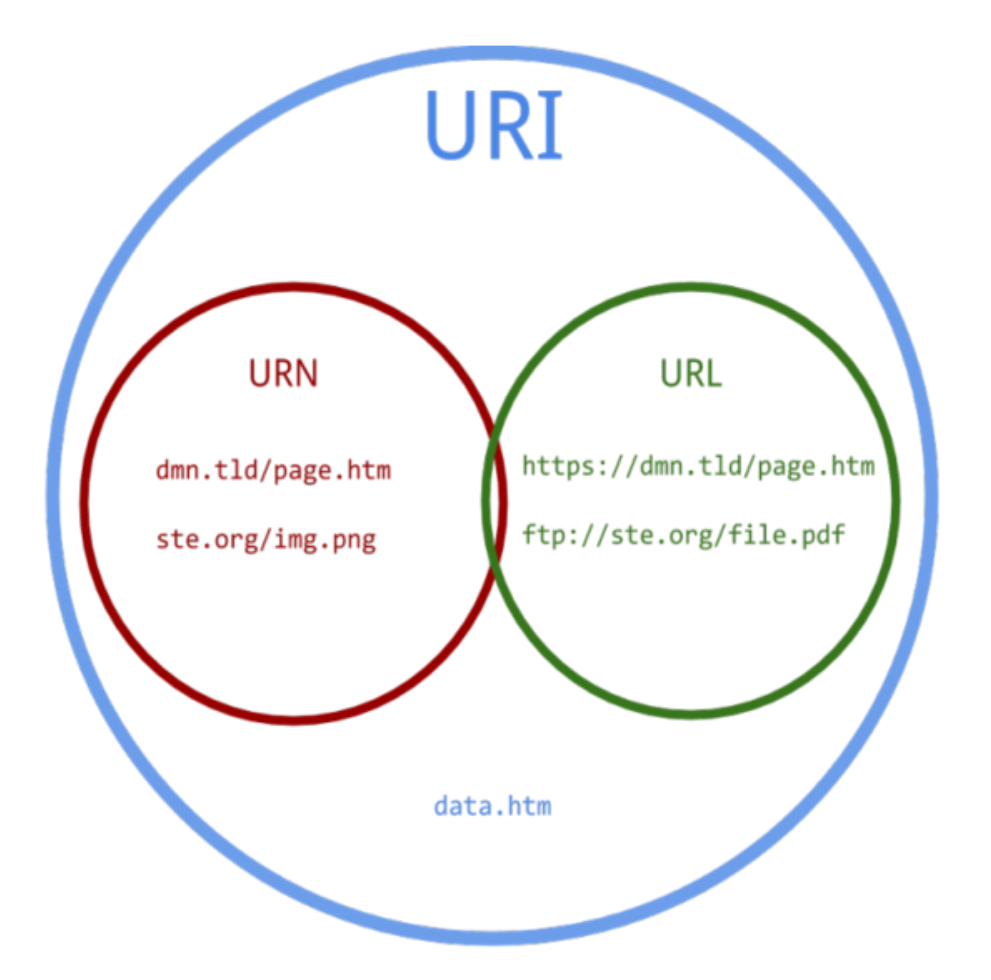
- 하위 개념으로는 **URL과 URN**이 있다.

- **URL(Uniform Resource Locator)** : 웹 상에 서비스를 제공하는 각 서버들에 있는 파일의 위치를 표시하기 위한 것.-> 어떤 자원의 위치에 대한 절대경로값을 지닌 문자열

Ex) http://blong.com/work/test.pdf 는 blog.com서버에서 work폴더안의 test.pdf를 요청

- **URN(Uniform Resource Name)** : 리소스의 위치에 상관없이 리소스의 이름값을 이용해서 접근하는 방식 -> 지속적이면서 위치에 독립적인 자원을 가르키기 위한 unique한 지시자

이미, URL이 대중화되어서 표준으로 사용되지는 않음

**3. HTML(Hyper Text Markup Language)**

- **www**서비스에서 **하이퍼텍스트** 문서를 만들기 위해 사용되는 **마크업** 언어

- www를 통해 서비스하는 모든 내용은 HTML로 표현되어야 한다.

**Hypertext?**

- 노드(node)와 링크(link)로 구성

노드는 문서고, 링크는 문서를 서로 연결하는 통로

->링크를 통해 서로 다른 문서 오가며 정보를 얻고, 필요할 경우 자신이 아는 정보를 문서로 만들어 다른 문서와 연결 가능

- 대표적인 예가 위키백과

**Markup Language?**

- 문서가 화면에 표시되는 형식을 나타내거나 데이터의 논리적인 구조를 명시하기 위한 규칙들을 정의한 언어

- 데이터를 기술한 언어로, 프로그래밍언어와는 차이가 있다.

**www?** World Wide Web,w3 -> 간단히 Web이라고 부름

- 태그(tag) : HTML을 기술하기 위해 사용되는 명령어의 집합

꺾쇠괄호(“<>”)안에 둘러 싸인 키워드

- HTML 요소(element)

: HTML은 세가지 구성 요소를 가진다.

시작 태그(start tag, opening tag)

콘텐츠(contents) : 태그가 적용하는 텍스트

종료 태그(end tag, closing tag) : 종료태그의 이름은 슬래스 문자(“/”)로 시작

+ 종료태그가 없는 태그도 존재 ex) <img>, <br> 등 이들은 빈 태그(empty tag)라 한다.

- HTML 요소 속성

: 대부분의 요소 속성들은 이름-값 형태이고, “=”로 구분하고, 요소의 시작태그 안에서 요소의 이름 다음에 쓰인다.

<**abbr** id="anId" class="aClass" style="color:blue;" title="하이퍼텍스트 마크업 언어">HTML</**abbr**>

Ex) id 속성 : 요소에 대한 문서 전체의 고유 식별자 제공

class 속성 : 유사한 요소를 분류

style 속성 : 다양한 tag의 스타일 지정

- HTML의 기본 구조 : 머리글(Heading)과 본문(Body)로 나뉜다.

**<html>**

**<head> <title>**타이틀**</title> </head>**

**<body>**

문서의 본문내용

**</body>**

**</html>**

<HTML>..</HTML> 태그는 HTML로 작성된 문서라는 것을 알려주는 태그, 문서의 시작과 끝을 알림.

<HEAD>..</HEAD> 태그는 머리글로, 문서의 기본정보를 포함하고 있는 부분. 문서 서문의 시작과 끝을 알림

그리고 <TITLE>..</TITLE>란 태그는 <HEAD>..</HEAD> 태그안에 포함되어 있으며, 문서의 제목을 나타내주는 태그  
  
<BODY>..</BODY>란 태그는 문서에서 실제적으로 보여지는 내용 입력. 문서 본문의 시작과 끝을 알림. 대부분의 태그가 이곳에 들어감.

**4. HTML5**

- HTML의 완전한 5번째 버전, 차세대 웹 표준으로 확정됨

문서 작성시 최상단에 아래 선언문을 적어야 HTML5을 사용할 수 있다.

<!DOCTYPE html>

특별한 경우가 아닌 이상 DOCTYPE 부분은 대문자로 적는 게 좋다.

- 기존 텍스트와 하이퍼링크만 표시하던 HTML이 멀티미디어 등 다양한 애플리케이션까지 표현, 제공하도록 진화된 언어

- HTML5에서는 **시맨틱 웹(semantic web)**을 중요시해서 **시멘틱 태그**라는 것을 여러가지 새로 만들었다.

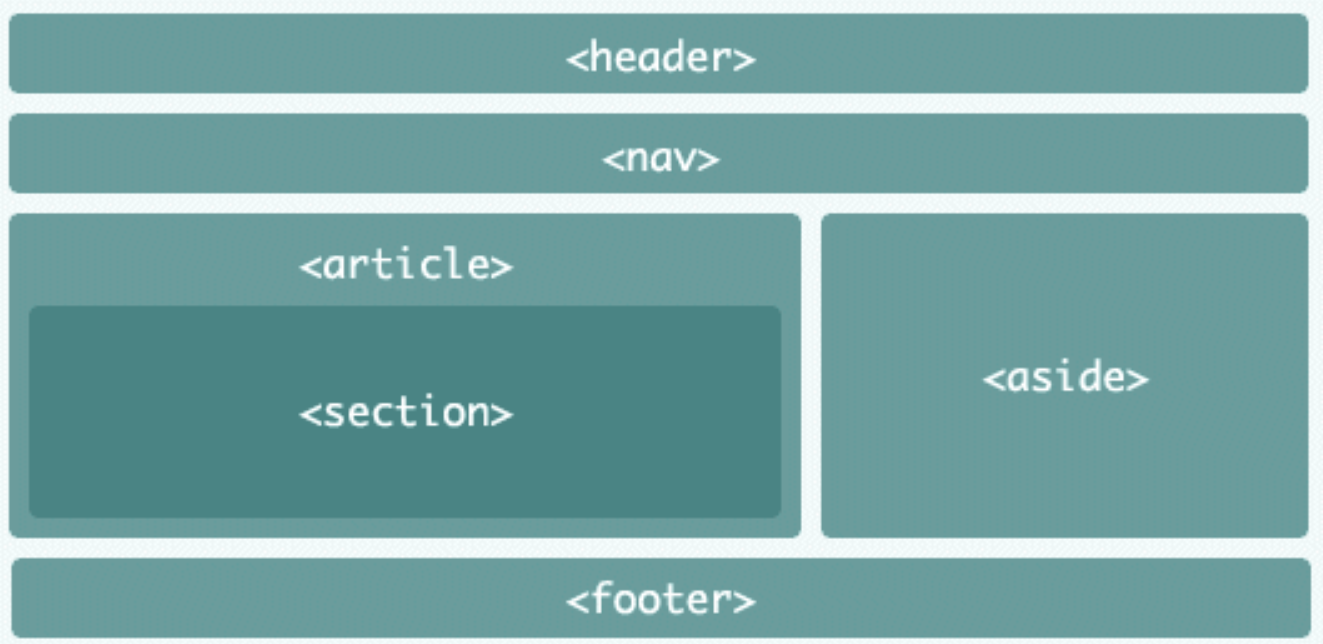
**시멘틱 웹(semantic web)**

- 기계가 사람을 대신해서 웹페이지의 정보를 이해하고, 우리에게 필요한 정보만 보여주거나 정보를 가공해서 우리가 필요로 하는 형태로 가공해 주는 것

**시멘틱 태그(semantic tag)**

- 웹페이지의 구조를 쉽게 이해할 수 있도록 정의된 태그

- 예시



예를 들면 이런 방식.

|  |  |  |
| --- | --- | --- |
| 1 |  | [cs](http://colorscripter.com/info#e)<div id="header"></div> |

과거에는 로 나타냈던 태그를 아래처럼 바꿈

|  |  |  |
| --- | --- | --- |
| 1 | <header></header> | [cs](http://colorscripter.com/info#e) |

로봇은 id가 header라고 적혀 있다고 해서 그 부분이 웹 페이지의 헤더에 해당하는지 알 수 없다. 게다가 사람에 따라서는 'pageHeader'과 같이 다른 형태로 header id를 작성하는 사람도 있을 수 있다.

**하지만 <header>라는 태그를 사용해서 웹 사이트의 헤더를 정의해주면 기계도 이 부분이 웹 페이지의 헤더라는 것을 이해할 수 있다.**

**5. CSS(Cascading Style Sheets)**

- HTML 등의 마크업 언어로 작성된 문서가 실제로 웹사이트에 표현되는 방식을 정해주는 언어

- HTML에 디자인 요소를 욱여넣다 보니, HTML의 본연의 목적인 구조화된 문서가 아닌 디자인 적인 문서가 되어 버림. 그래서 디자인적 요소를 따로 전담하기 위해 만들어진 언어

- 장점 : 작성이 쉬움

- 문법 : 선택자(selector)와 선언부(declaratives)로 구성



선택자는 CSS를 적용하고자 하는 HTML 요소

선언부는 하나 이상의 선언들을 세미클론(;)으로 구분하여 포함할 수 있으며, 중괄호({})를 사용해서 전체 둘러쌈

각 선언은 CSS 속성명과 속성값을 가지며, 그 둘은 콜론(:)으로 연결됨, 마지막은 세미클론(;)

- CSS를 HTML에 적용하는 방법 3가지

인라인 스타일(inline style) : HTML 요소 내부에 style 속성에 CSS코드 넣어 CSS스타일 적용 하는 방법

<body>

    <h2 style="color:green; text-decoration:underline">

        인라인 스타일을 이용하여 스타일을 적용하였습니다.

    </h2>

</body>

-> 선택자 h2의 스타일을 꾸며줌. 단, 인라인 방식은 재사용이 불가능

내부스타일 시트(internal style sheet) : HTML 문서 내의 <head>태그에 <style>태그 안에 CSS코드 넣어 CSS 스타일 적용

<head>

    <style>

        body { background-color: lightyellow; }

        h2 { color: red; text-decoration: underline; }

    </style>

</head>

-> 선택자 h2의 스타일을 꾸며줌.

인라인과 다른 점은, 문서안의 모든 h2요소들의 스타일이 바뀐다는 것이다.

문서안의 여러 요소를 한번에 꾸밀 수는 있으나, 다른 HTML문서에는 적용할 수 없다.

외부스타일 시트(external style sheet) : 별도의 CSS파일을 만들어 HTML 문서와 연결하는 방법

외부에서 작성된 스타일 시트 파일은 .css확장자를 사용해 저장한뒤

스타일을 적용할 웹페이지의 <head>태그에 <link>태그를 사용하여

외부 스타일 시트를 포함해 스타일 적용

<head>

    <link rel="stylesheet" href="/examples/media/expand\_style.css">

</head>

-> 여러 HTML 문서에서 사용할 수 있다.

- 스타일 적용 우선순위: 인라인 스타일 > 내부/외부 스타일 > 웹브라우저 기본 스타일

**스크립트 언어**? 프로그래밍 언어의 한 종류, 기존에 이미 존재하는 소프트웨어를 제어하기 위한 용도

웹의 전반적인 구조를 알고 있다면, 클라이언트가 서버에게 요청한 페이지를 서버에서 잘 가공하여 다시 클라이언트에게 응답하는 간단한 구조는 알고 있을 것이다.(맨 위에서 이미 설명)

여기서 어느 측에서 요청을 처리하느냐에 따라서 클라이언트 사이드 스크립트 언어(Client Side Script Language)와 서버 사이드 스크립트 언어(Server Side Script Language)로 나뉘게 되는데, 말 그대로 클라이언트에서 실행되는 스크립트 언어를 클라이언트 사이드 스크립트 언어라고 하고, 서버에서 실행되는 스크립트 언어를 서버 사이드 스크립트 언어라고 한다.

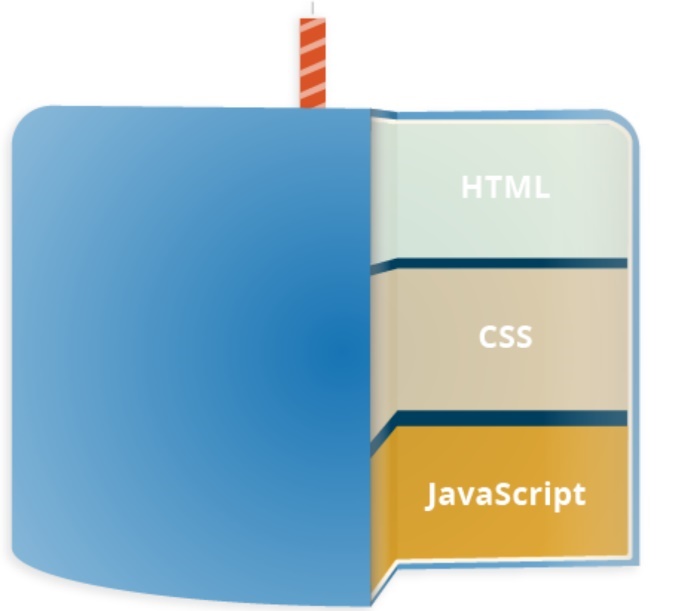
클라이언트 사이드 스크립트 언어 : 자바스크립트, VB스크립트

서버 사이드 스크립트 언어 : JSP, ASP, PHP, Python

**6. Javascript**

- .js확장자

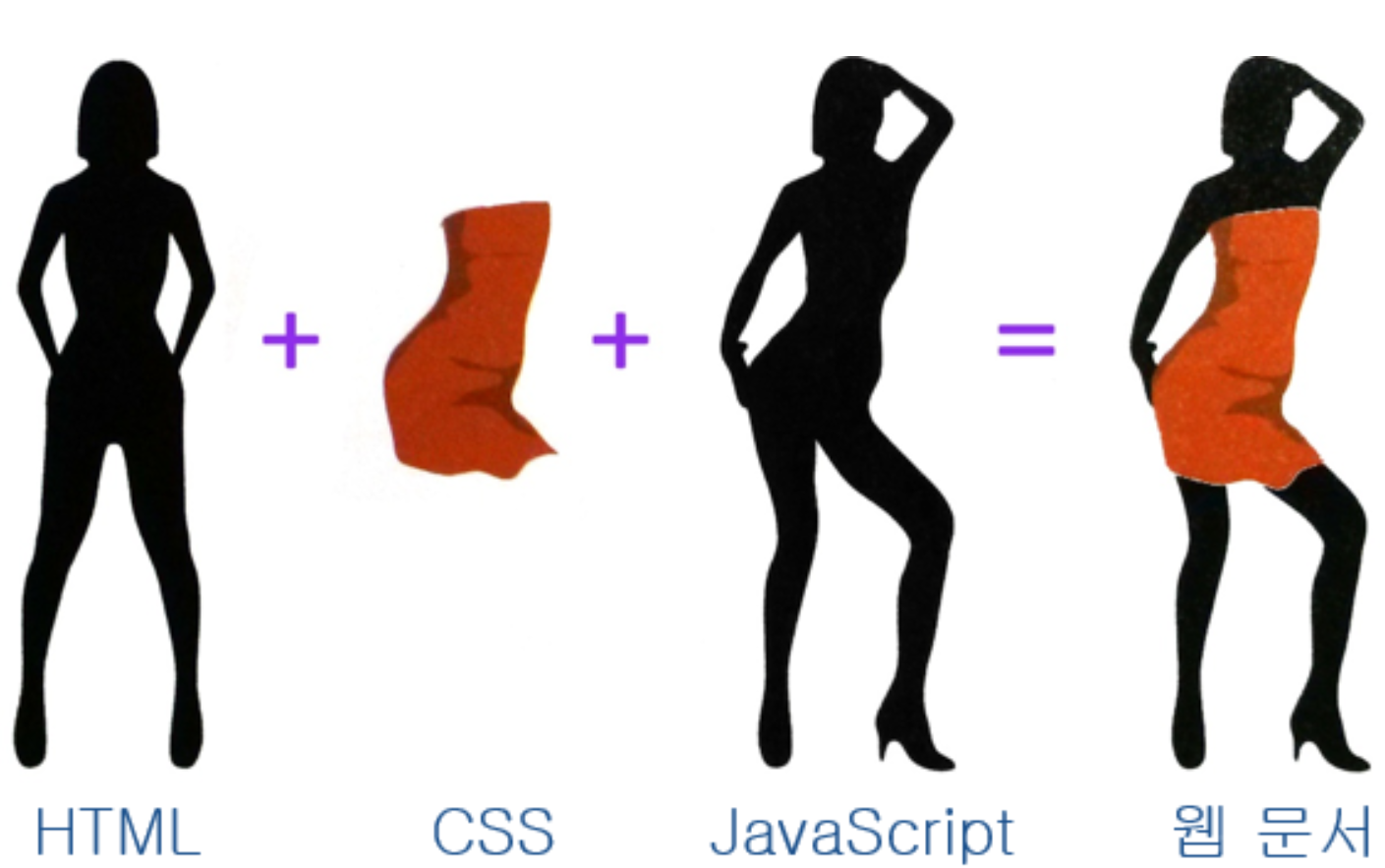
- 동적으로 컨텐츠를 바꾸고, 멀티미디어를 다루는 등을 할 수 있는 스크립트 언어, 웹 브라우저에서 주로 사용



- HTML로는 웹 내용을 작성하고,

CSS로는 웹을 디자인하며,

자바스크립트로는 웹의 동작을 구현



- 자바와 자바스크립트는 서로 관련이 없다.

-자바스크립트 변수는 var

var a = 1;

var test = “hello”;

var x = true;

- 자바스크립트의 특징

객체 기반의 언어이다. 하지만 상속과 클래스라는 개념은 없다.

객체(object): 프로그래밍 대상을 속성과 동작으로 묶어서 정의한 자료형

인터프리터 언어로서 클라이언트의 웹 브라우저에 의해 해석되고 실행된다.

HTML 문서 내에 기술되고, HTML문서와 함께 수행된다.

HTML에 연산 제어 등 프로그래밍 적인 요소를 추가하고 클라이언트의 자원을 활용할 수 잇게 한다.

- 자바스크립트의 장점 : 컴파일 과정이 없어서 빠른 시간 내에 코드 작성 가능. 웹에 특화되어서 운영체제나 플랫폼 상관없이 잘 작동

- 자바스크립트의 단점: 성능이나 보안. -> 내부에서 제공되는 기능이 제한적이고 개발도구도 적음. 또 HTML소스코드와 함께 작성되면서 소스코드가 외부로 공개되어 보안취약점 발생 가능

- 사용용도 예시

사용자의 입력을 제어하고 싶을 때

웹 브라우저 내의 여러 가지 기능을 조절하고 싶을 때

사용자에게 메시지를 보내고 싶을 때(경고나 확인)

쿠키를 이용한 페이지 간의 data 유지

**쿠키(cookie)** : 서버가 사용자의 웹브라우저에 전송하는 작은 데이터 조각

동적인 웹사이트 제작(HTML 요소나 스타일 동적 변경)

**자바스크립트의 라이브러리는 prototype, jQuery YUI 등등이 있지만 제일 많이 사용하는 건 jQuery**

**7. jQuery**

- 자바스크립트 라이브러리

**라이브러리** : 자주 사용되는 로직들을 재활용 가능하도록 만든 로직들의 묶음

- 기본 문법

$(선택자).동작함수();

달러($) 기호는 제이쿼리를 의미하고, 제이쿼리에 접근할 수 있게 해주는 식별자

선택자를 이용하여 원하는 html 요소를 선택하고, 동작함수를 정의하여 선택된 요소에 원하는 동작설정

예제) 문서 내의 모든 <li>요소를 선택하여 변수에 저장한 후, 해당 변수를 사 용하는 예제.

##### 예제

$(*function*() {

*var* items = $("li"); *// <li>요소를 모두 선택하여 변수 items에 저장함.*

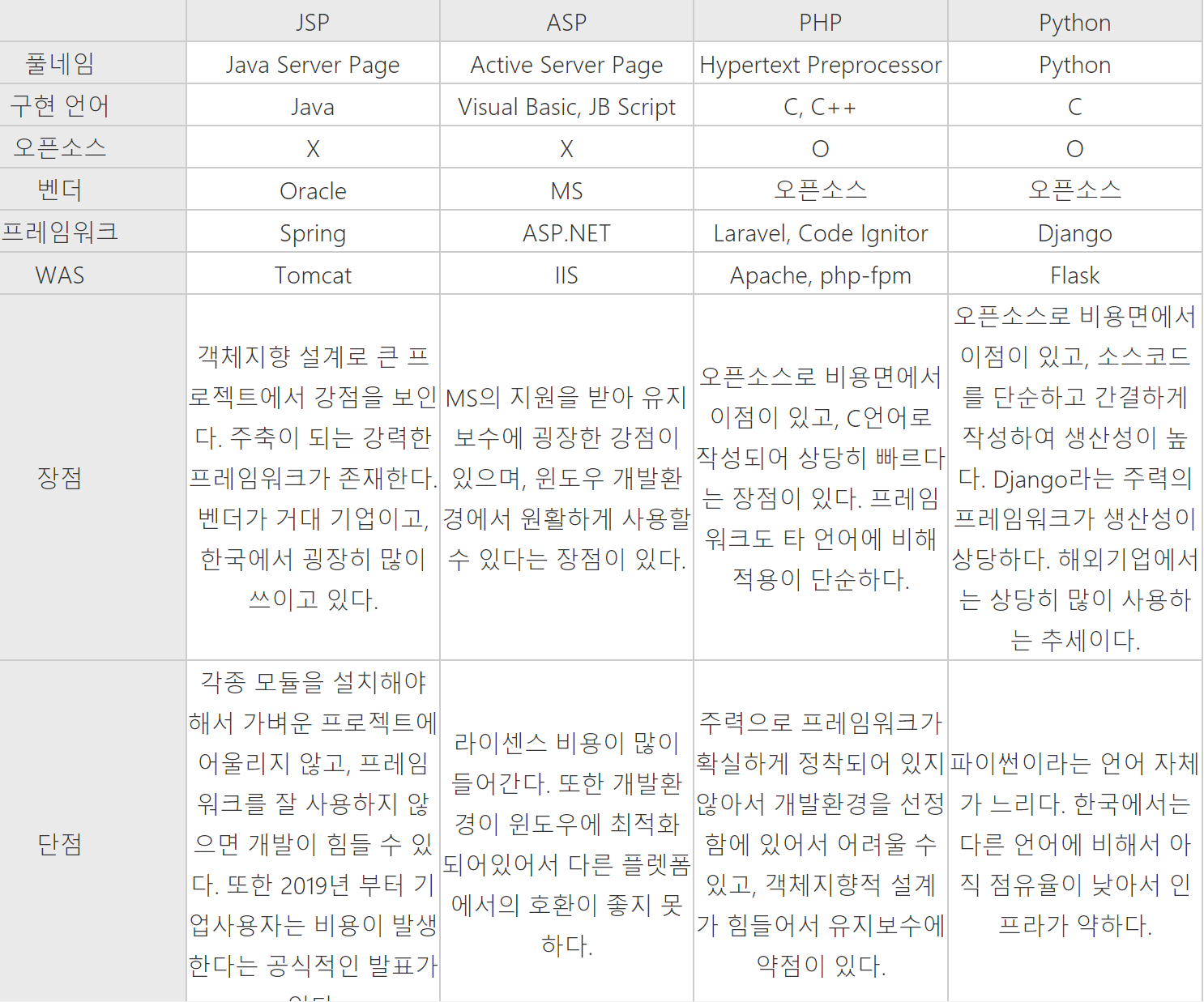
$("button").on("click", *function*() {

$("#len").text("저장된 <li>요소의 총 개수는 " + items.length + "개 입니다.");

});

});

그 다음 서버 사이드스크립트 언어



**8. JSP(java server page)**

- .jsp확장자

- 자바(java)를 서버환경에서 사용하는 스크립트 방식의 언어

- 자바스크립트가 **클라이언트**의 웹 브라우저 화면에 실행되는 방법이라면, JSP는 **서버**에서 실행되어 사용자에게 보여주는 방식 즉 jsp에서 작성된 내용은 서버에서 실행되고 그 결과가 웹 브라우저로 전송

**9. PHP(Hypertext Preprocessor)**

- .php 확장자

- c언어를 기반으로 서버환경에서 사용하는 스크립트 방식의 언어