JavaScript

JavaScript의 개요

- 자바스크립트(JavaScript)는 웹 브라우저 상에서 실행되는 스크립트 언어
  - HTML만으로 작성되어 정적이기만 한 웹 문서에 JavaScript를 통해 동적인 동작을 추가 할 수 있게 됨
  - 마우스의 클릭, 키보드의 입력 뿐 아니라 사용자가 페이지를 열거나 이동할 때를 알아내어 원하는 작업을 수행함
- 특징
  - 스크립트 언어
  - 이벤트 중심 프로그래밍
  - 객체 기반/지향 언어

기본 구문

- <script> : HTML에서 JavaScript 코드를 작성하기 위한 태그
  - 작성법
    - <script> 태그 내에 JavaScript 코드를 작성한다.
    - 외부의 파일에 JavaScript 코드를 작성 한 뒤 <script> 에 파일의 위치를 지정
  - 속성
    - language
    - src

```
<script language="javascript" src="script/script.js"></script>
<script language="javascript">
  //java script 코드
</script>
```

- 코드 작성위치
  - JavaScript 프로그램은 HTML의 어느 부분에 삽입해도 지장없으나 일반적으로 HTML 문의 헤더 부분에 두는 것이 권장되고 있다.

변수와 데이터 타입

- 변수 : 데이터를 저장할 수 있는 공간
- 변수 선언 : 변수의 이름과 데이터 타입 등을 선언하는 작업
  - 자바스크립트는 데이터 타입 선언 없이 변수의 이름만을 선언한다.
  - 선언구문
    - [var] 변수명 [=값];

EX) var num;

var name = "홍길동";

var age = 10;

address="서울"

- 데이터 타입
  - 숫자 : 정수, 실수
  - 문자열 : 값은 ' ' 또는 " " 로 묶는다.
  - 논리형 : true, false
  - null

함수

- 프로그램 수행의 일련 과정을 하나의 단위로 묶어주는 수행block
- 함수는 객체로 취급된다.
- 반복 호출이 가능 – event 핸들러나 다른 함수에서 호출
- 구문

```
function 함수이름(args...){
        //함수 코드
        [return value]
}
```

```
function abc(){
    alert("abc");
    return 10;
}
```

```
var 함수이름 = function(args.. ){
        //함수코드
        [return value]
}
```

```
var abc = function(){
  alert("abc");
  return 10;
}
```

```
window.함수이름=function(args..){
        //함수코드
        [return value]
}
```

```
window.abc = function(){
        alert("abc");
        return 10;
  }
```

제어문 - 반복문

- while문
  - 구문

```
while(조건){
  반복구문;
  [break;]
}
```

- for문
  - 구문

```
for( 초기식;조건식;증감식){
  반복구문;
  [break;]
}
```

제어문 - 조건문

- switch case 문
  - 구문

```
switch(비교대상값){
  case value:
    코드;
    [break;]
  default:
    코드
    [break;]
}
```

- if 문

```
if(조건){
  코드
}else if(조건){
  코드
}else{
  코드
}
```

Event

- 자바 스크립트는 사용자의 요청처리를 Event 모델을 통해 처리한다.
  - Event
    - 사용자가 특정 동작(사건)을 가해 발생되는 신호(signal)
    - 마우스로 버튼 클릭, Text 입력양식에 글 입력, 페이지 로딩 등..
  - Event Handler
    - Event 발생시 처리하는 코드를 등록하는 것.

예

```
<input type="button" value="확인" onClick="alert('button클릭')"/>
<form action="a.jsp" onSubmit="alert('전송합니다.')">
```

Event

- 종류

Event	Handler	설명
focus	onFocus	입력 양식를 선택해서 포커스가 주어졌을 때
blur	onBlur	포커스가 폼의 입력 양식을 벗어났을 때
select	onSelect	입력 양식에서 한 필드를 선택했을 때 발생
change	onChange	입력 양식에 있던 값이 바뀌었을 때 발생
load	onLoad	해당 페이지가 로딩 되었을 때(처음 읽힐 때) 발생
unload	onUnload	해당 페이지를 빠져 나갈때 발생
mousemove	onMousemove	해당영역에 마우스를 움직였을 때 발생
mouseover	onMouseover	해당 영역에 마우스가 올라갔을 때 발생
mouseout	onMouseout	해당 영역에서 마우스가 나갔을 때 발생
mousedown	onMousedown	해당 영역에서 마우스를 눌렀을 때 발생
mouseup	onMouseup	해당 영역에서 마우스를 눌렀다가 떼었을 때 발생
click	onClick	해당 영역에서 마우스를 클릭 했을 때 발생
keydown	onKeydown	해당 영역에서 키보드를 눌렀을 때 발생
keyup	onKeyup	해당 영역에서 키보드를 눌렀다가 떼었을 때 발생
keypress	onKeypress	해당 영역에서 키보드를 계속 누르고 있을 때 발생
submit	onSubmit	폼의 내용을 전송 할 때 발생
reset	onReset	폼의 내용을 초기화 시킬 때

JavaScript와 객체

- 자바스크립트는 객체지향언어(Object Oriented Language)
  - Java Script가 지원하는 객체만 사용할 수 있었으나 1.2 버전부터는 개발자가 직접 객체를 만들 수 있다.
- 객체의 종류
  - 내장 객체 : 웹 브라우저에서 제공하는 객체
    - 자바스크립트 내장객체 : JavaScript 코드 작성시 유용한 기능을 제공해 주는 객체들
    - DOM 객체 (브라우저 내장객체) : 웹 브라우저의 각 요소들을 객체로 제공
  - 사용자 정의 객체 : 사용자가 만드는 객체

JavaScript 내장객체 - Array

- 배열 처리를 위한 객체
  - 생성 구문
    - 변수 = new Array([length]);
    - 변수 = new Array(value, value…);
    - 변수 = [value, value, value…]
  - 값 접근
    - 변수[index]
  - 속성
    - length : 배열의 크기

```
var arr = new Array();
arr[0] = 10;
arr[1] = "홍길동";
for(i = 0; i<arr.length; i++){
    alert(arr[i]);
}
```

JavaScript 내장객체 - Date

- 날짜와 시간을 다루는 객체
  - 생성 구문
    - 변수 = new Date();
    - 변수 = new Date(**[**년, 월, 일, 시, 분, 초, 1/1000**]**);
  - 주요 메소드

메 소 드	설 명
getFullYear()	1900년대 4자리
getYear()	1900년대 2자리(2000년대는 4자리로나옴)
getMonth()	월(0~11) 예) 0:1월, 11:12월 …
getDate()	날짜
getDay()	요일(0~6) 예) 0:일요일, 6:토요일
getHours()	시(0~23)
getMinutes()	분(0~59)
getSeconds()	초(0~59)
getTime()	1970년 1월 1일 이후 시간을 1/1000초 단위로 나타낸 값

```
today = new Date();
alert(today.getFullYear()+"."+(today.getMonth()+1)+"."+today.getDate);
```

JavaScript 내장객체 - String

- 문자열을 관리하는 객체
  - 값은 " "로 표현
  - 생성구문
    - 변수 = "값";
    - 변수 = new String("값");
  - 주요메소드

메 소 드	설 명
charAt(index)	지정된 위치에서 문자 찾기
indexOf(string)	지정된 문자의 위치를 왼쪽부터 찾기
lastIndexOf(string)	지정된 문자의 위치를 오른쪽부터 찾기
substring(index1, index2)	문자열의 일부를 추출하기 index1 ~ index2-1
toLowerCase()	소문자로 변환하기
toUpperCase()	대문자로 변환하기
concat(string)	두 문자열을 합치기
slice(start_index, end_index)	문자열의 일부를 추출하기
substr(start_index, length)	문자열을 length만큼 잘라내기

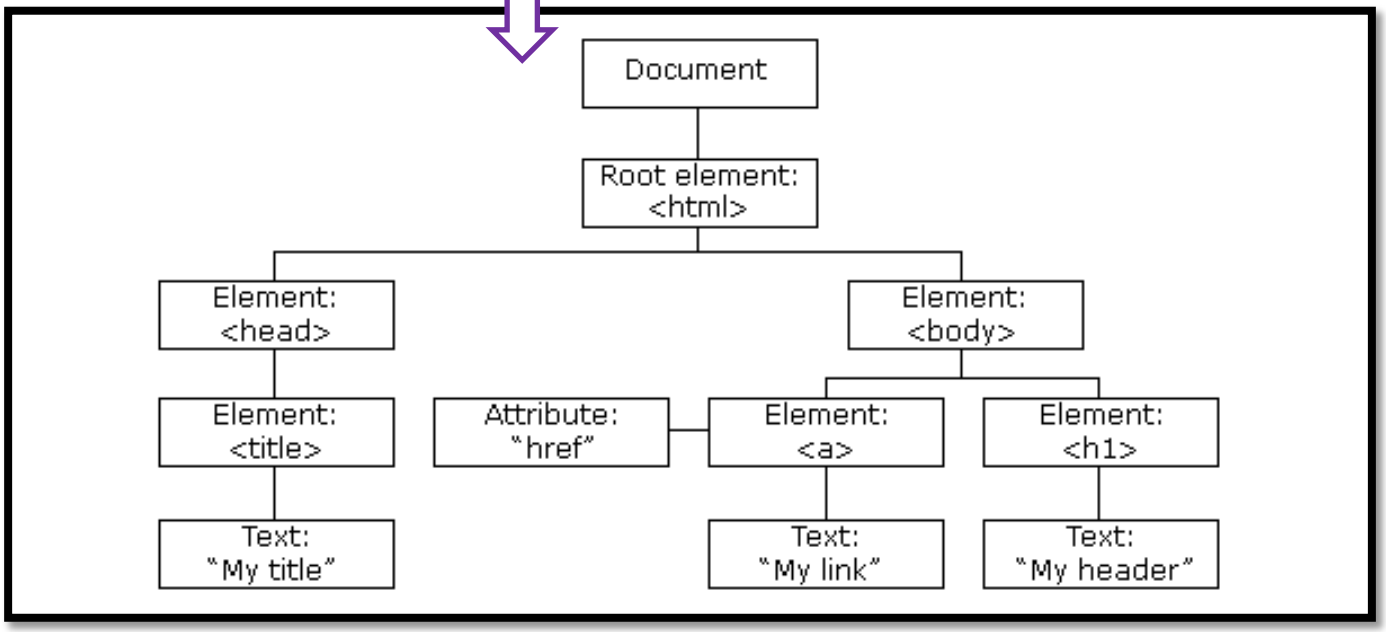
DOM(Document Object Model)

- 객체지향 모델로 구조화된 문서를 표현하는 형식
- 역할 : 동적으로 문서의 내용, 구조, 스타일에 접근하여 변경, 생성
- 구성
  - Element(요소) 노드
  - Text 노드
  - Attribute(속성) 노드
- 웹 문서가 로딩 시 Web Brower에 의해 생성/구성 됨

DOM(Document Object Model)

- Tree 구조

```
<html>
   <head>
         <title>My Title</title>
   </head>
   <body>
         <a href="#">My Link</a>
         <h1>My header</h1>
   <body>
</html>
```



DOM 객체 (브라우저 내장객체)

- window 객체
  - 브라우저 내장객체 중 최상위 트리에 위치한 객체
  - 브라우저에 대한 정보를 가지고 있는 객체
  - 접근방법
    - window.속성, window.메소드
    - window 는 생략 가능
  - 주요 메소드

메소드	설 명
open()	새로운 브라우저(윈도우) 열기
close()	열려있는 브라우저(윈도우) 닫기
alert()	간단한 메시지를 보여주기 위한 dialog box
confirm()	사용자로부터 어떠한 작업을 확인 받기 위한 dialog box
prompt()	사용자로부터 문자열을 입력 받기 위한 dialog box
back()	이전 페이지로 이동
parseInt(String) parseFloat(String)	인수의 String을 정수, 실수 형태로 변환
isNaN(String)	인수의 String이 숫자 형태가 아니면 true, 숫자형태이면 true리턴
eval(String)	String을 실제 형태로 변환하여 주는 함수.

DOM 객체 (브라우저 내장객체)

- window 객체

```
<script>
    //Here
</script>
```

  - 위 here 부분에 선언된 변수, 함수는 모두 window객체의 property로 등록된다.
  - 예

```
<script>
    var flag = false;
    function abc(){
        alert("a");
    }
</script>
```

```
<script>
    window.flag = false;
    window.abc = function(){
        alert("a");
    }
</script>
```

DOM 객체 (브라우저 내장객체)

- document객체 :
  - 브라우저에 보여지는 웹 문서를 가리키는 객체
  - window의 하위 객체
  - 접근 방법
    - window.document.속성, document.메소드
  - 주요 메소드

메소드	설 명
write("string")	문서에 데이터를 출력함
writeIn("string")	문서에 데이터를 출력하고 줄바꿈 실행

- history 객체
  - 브라우저가 방문한 URL의 정보를 저장하고 있는 객체
  - window의 하위객체
  - 접근 방법
    - window.history.속성, history.메소드
  - 주요메소드

메소드	설 명
back()	히스토리 리스트에 등록되어 있는 URL 중 가장 최근의 주소로 이동. (일반적으로 현재 페이지의 바로 이전 페이지로 이동한다.)
forward()	back() 메소드를 이용하여 이동 하였을 경우 다시 전 페이지로 이동.
go()	원하는 주소로 이동. go(-2) : back() 메소드를 2번 호출 한 것과 같다. go(2) : forward() 메소드를 2번 호출 한 것과 같다.

DOM 객체 (브라우저 내장객체)를 통한 Form 처리

- form 객체 :
  - <form> 객체
  - document의 하위 객체
  - 접근 방법
    - window.document.form_name.속성, window.document.form_name.메소드
  - 주요 속성

속성	설 명
action	Form태그 내의 ACTION 속성에 기술된 문자열 Form 의 정보를 이용하여 수행되는 프로그램 혹은 목적 페이지
method	전송 HTTP 메소드 설정. (get, post 등)
encoding	Form 을 통해 서버로 전송되는 정보의 MIME 형태를 기술하는 문자열 Form 태그에서의 ENCTYPE속성을 반영
name	Form 태그에서의 NAME 속성의 값
target	Form 전송의 목적(Target)윈도우 /프레임 이름을 나타내는 속성

  - 주요 메소드

메소드	설 명
reset()	입력양식을 초기화 시킨다. RESET 버튼과 동일한 기능.
submit()	입력된 값을 서버로 전송한다. SUBMIT 버튼과 동일한 기능.

DOM 객체 (브라우저 내장객체)를 통한 Form 처리

- checkbox 객체 :
  - <input type="checkbox"> 객체
  - form의 하위 객체
  - 접근 방법
    - window.document.form_name.checkbox_name.속성,
      window.document.form_name.checkbox_name.메소드
  - 주요 속성

속성	설 명
name	Checkbox의 이름.
value	Checkbox의 값.
checked	Checkbox의 현재 체크 상태 – true/false

  - 주요 메소드

메소드	설 명
focus()	객체에 포커스를 준다.

DOM 객체 (브라우저 내장객체)를 통한 Form 처리

- radio 객체 :
  - <input type="radio"> 객체
  - form의 하위 객체
  - 접근 방법
    - window.document.form_name.radio_name.속성,
      window.document.form_name.radio_name.메소드
  - 주요 속성

속성	설 명
name	Radio의 이름.
value	Radio의 값.
checked	Radio의 현재 체크 상태 – true/false
length	한 그룹내의 Radio 개수

  - 주요 메소드

메소드	설 명
focus()	객체에 포커스를 준다.

DOM 객체 (브라우저 내장객체)를 통한 Form 처리

- select 객체 :
  - <select> 객체
  - form의 하위 객체
  - Select 태그의 값은 <option> 태그에 있다. 값에 접근하기 위해서는 select 를 통해 option 객체로 접근한다.
  - 접근 방법
    - window.document.form_name.select_name.속성,
      window.document.form_name.select_name.메소드
      값에 접근 : window.document.form.select.**options[0].text**
  - 주요 속성

속성	설 명
name	Select의 이름.
selectedIndex	선택된 Option의 인덱스. 0부터 시작
length	한 그룹내의 Select 개수. <OPTION> 태그의 개수.
options	Option객체 배열로 return.

  - 주요 메소드

메소드	설 명
focus()	객체에 포커스를 준다.

DOM 객체 (브라우저 내장객체)를 통한 Form 처리

- option 객체 :
  - select 내의 <option> 태그 객체
  - select 객체의 options 속성을 통해 배열로 받아 접근한다.
  - 주요 속성

속성	설 명
text	태그내 content
value	value 속성의 값
selected	선택 여부. true/false

DOM 객체 (브라우저 내장객체)를 통한 Form 처리

- text객체 :
  - <input type="text"> 객체
  - form의 하위 객체
  - 접근 방법
    - window.document.form_name.text_name.속성,
      window.document.form_name.text_name.메소드
  - 주요 속성

속성	설 명
name	Text의 이름.
value	Text의 값.

  - 주요 메소드

메소드	설 명
focus()	객체에 포커스를 준다.

DOM 객체 (브라우저 내장객체)를 통한 Form 처리

- textarea객체 :
  - <textarea> 객체
  - form의 하위 객체
  - 접근 방법
    - window.document.form_name.textarea_name.속성,
      window.document.form_name.textarea_name.메소드
  - 주요 속성

속성	설 명
name	Text의 이름.
value	Text의 값.

  - 주요 메소드

메소드	설 명
focus()	객체에 포커스를 준다.

DOM 다루기

- Javascript에서 DOM객체를 동적으로 생성, 삭제, 추가 등 관리를 할 수가 있다.
- Dom 객체 접근
- Dom 객체 생성, 추가, 삭제

DOM 다루기 – 노드 접근

- 장거리 접근
  - document.getElementById(“태그 id”)
    - 태그의 ID 속성으로 접근

```
<p id='text'></p>

var x = document.getElementById("text");
```

  - document.getElementsByTagName(“태그이름”)
    - 태그 명으로 접근
    - 노드 리스트 Return : 배열 개념

```
//모든 <p> 태그 조회
var par = document.getElementsByTagName("p");
for(i=0;i<par.length;i++){
  //par[i] 로 작업
}
```

DOM 다루기 – 노드 접근

- 장거리 접근
  - document.getElementsByName("태그 name속성값")
    - 태그의 name 속성의 값으로 접근
    - Node 리스트 리턴 : 배열개념

```
<input type="text" name="age">

var x = document.getElementsByName("age");
```

DOM 다루기 – 노드 접근

- 단거리 접근 – 특정 노드 객체를 중심으로 그 주위 노드 접근
  - parentNode
    - 부모 노드 return
  - firstChild
    - 첫 자식 노드 return
  - lastChild
    - 마지막 자식 노드 return
  - previousSibling
    - 앞 쪽 형제 노드 return
  - nextSibling
    - 뒤 쪽 형제 노드 return
  - children/childNodes
    - 모든 자식노드를 노드 리스트로 return
    - 부모노드.children.length : 모든 자식 노드의 개수
    - 부모노드. children[0] : 첫 번째 자식 노드

DOM 다루기-노드의 생성,삽입

- document.createElement(태그명)
  - 태그 생성
  - var x = document.createElement('p'); // <p> 태그 생성
- document.createTextNode("문자열")
  - 텍스트 요소 생성
  - var txt = document.createTextNode("hello");
- appendChild() : 자식 노드 추가
  - 하위 노드 추가. 마지막 자식 노드로 추가한다.
  - 부모노드.appendChild(추가할 자식노드)
- insertBefore() : 자식 노드 추가
  - 특정 하위 노드 앞에 새 하위 노드를 추가한다.
  - 부모노드.insertBefore(추가할 자식노드, 기존 자식노드)

DOM 다루기-노드의 생성(document객체 메소드)

- hasChildNodes() : 자식노드 유무
  - true/false 리턴
- removeChild() : 자식노드 삭제
  - 자식 노드를 제거한다.
  - 부모노드.removeChild(삭제할 자식노드)
- replaceChild()
  - 자식 노드를 다른 노드로 교체 한다.
  - 부모 노드.replaceChild(새 노드, 교체할 자식노드)
- setAttribute()
  - 노드의 속성 추가
  - Node.setAttribute("속성명", "속성값");
- getAttribute()
  - 노드의 속성 값 조회
  - var attrValue = Node.getAttribibute("속성명")

<DIV> 또는 <SPAN>을 통한 값 출력

- <DIV> <SPAN> - 범위 지정 태그
- **<DIV>**
  - 자동 줄 바꿈 기능을 가지고 있다.
  - CLASS 속성이나 ID속성과 함께 사용하여 임의의 범위에서 스타일을 설정, 값 변경 등 다양한 작업이 가능
- **<SPAN>**
  - <DIV>와 같은 목적으로 사용
  - 태그는 줄 바꾸기 기능이 없다.
- 태그객체.innerHTML;
  - java script에서 특정 태그 영역에 접근 할 때 사용
  - var value = 태그 객체.innerHTML; 태그 내의 값 return
  - 태그객체.innerHTML = value; 태그 내에 값 삽입

```
function exam(){
    document.getElementById("result").innerHTML = "결과값";
}

<div id="result"></div>
```