HTML/CSS/Java Script

yskim@suned.co.kr
http://cafe.naver.com/sunschool

HTML

yskim@suned.co.kr
http://cafe.naver.com/sunschool

1. HTML 문서의 기본적인 형식

- **HTML 문서는 태그를 이용하여 문서의 모양을 지정하는 텍스트 파일이다.**
- **HTML 문서 안에서는 대소문자나 공백 문자는 무시된다.**
- **파일이름은 .html 또는 .htm 확장자를 사용한다.**
- **HTML 문서는 헤더부분과 바디 부분으로 나뉜다.**

```
<html>
    <head>
        <title>브라우저제목</title>
        헤더정보
    </head>
    <body>
        본문내용
    </body>
</html>
```

2. <meta> </meta> 태그

- Meta 태그는 웹브라우저의 행동을 제어하거나 검색엔진이 HTML문서의 키워드를 저장하도록 하는 기능을 제공한다.
- Meta 태그는 반드시 HTML 문서의 head 태그 내에 기술하여야 한다.
- http-equiv 와 name 속성에는 HTTP헤더이름이나 다른 이름을 기술하고, content에는 값을 기술한다.
- Meta 태그는 다음과 같은 속성을 가진다.
  - http-equiv : HTTP 헤더의 이름을 기술하여 웹브라우저의 행동을 제어할수 있다. 다음과 같은 값을 가질수 있다.
    - content-type : MIME 타입을 기술한다.
    - content-disposition : 응용프로그램의 핸들러를 기술한다.
    - pragma : HTTP/1.0에서 Caching을 제어한다. 'no-cache'와 같은 값을 사용한다.
    - cache-control : HTTP/1.1에서 cache agent 행동을 기술한다. 'no-cache'와 같은 값을 사용한다.
    - refresh : HTML문서를 리로드하기전에 걸리는 시간을 기술한다. URL을 추가하여 다른 문서를 로드시킬수 있다.
    - set-cookie : 쿠키를 설정한다.
  - name : HTTP헤더에 포함되지 않은 다른 타입을 기술하기 위해 사용된다.
    - keywords : 현재 HTML페이지를 검색엔진이 indexing할수 있도록 키워드를 기술한다.
    - author : HTML페이지 작성자를 기술한다.
    - generator : HTML 페이지 작성 도구를 기술한다.

<meta> 태그 사용예

- content-type 설정예
  - <meta http-equiv="content-type" content="text/html;charset=euc-kr">
- refresh를 사용하여 페이지 이동
  - <meta http-equiv="refresh" content="2;url=test.html">
    - 2초후에 test.html 페이지로 이동한다.
- refresh를 사용하여 파일 다운로드
  - <meta http-equiv="refresh" content="2;url=test.zip">
    - 2초후에 test.zip 파일을 다운로드 하는 창이 화면에 나타 난다.
- 키워드 설정예
  - <meta name="keywords" conetent="servlet,jsp">

3. HTML로 문서 모양 만들기

- **글자크기 지정**
  <hn> …</hn> (n은 1에서 6까지 숫자)

- **문단지정**
   …

- **줄바꾸기**

 또는

- **수평선 그리기**
  <hr>
  <hr size="n" width="n" align="center" noshade> (noshade는 음영효과 없애기)

- **가운데 정렬**
   . . .

- **주석**
  <!-- 주석 -->

4. HTML 문자 모양 만들기

- 문자의 논리적 스타일

태그	이름	용도
<dfn> … </dfn>	Definition	용어 정의
…	Emphasis	부분 강조
<cite> … </cite>	Citation	부분 인용
<code> … </code>	Code	타이핑된 코드
<kbd> … </kbd>	Keyboard	사용자 키보드 입력
<samp> … </samp>	Sample	예제
…	Strong	강한 강조
<var> … </var>	Variable	변수 이름

5. HTML 문자 모양 만들기

■ 문자의 물리적 스타일

태그	이름	용도
…	Boldface	볼드체
…	Italic	이탤릭체
<u> … </u>	Underlined	밑줄 문자
<tt> … </tt>	Typewriter	타지기체
<blink> … </blink>	Blinking	문자 깜빡임
_…	Subscript	아래첨자
[…]	Superscript	위첨자

■ 스페셜 태그

태그	용도
<pre> … </pre>	에디터에서 편집되어 있는 모양 그대로 브라우저에 보여준다
<blockquote> … </blockquote>	인용문단을 다른 문단과 구별해서 보여준다.
<address> … </address>	HTML을 만든 사람이 누구인지 알리기 위해 사용된다.

6. HTML 로 목록과 표 만들기

- 문자의 논리적 스타일

설명	예제 코드	화면결과
순서 있는 목록	 - 사과 - 바나나 - 키위 	1. 사과 2. 바나나 3. 키위
순서 없는 목록	 - 사과 - 바나나 - 키위 	• 사과 • 바나나 • 키위
메뉴 목록	<menu> - 사과 - 바나나 - 키위 </menu>	• 사과 • 바나나 • 키위
디렉토리 목록	<dir> - 사과 - 바나나 - 키위 </dir>	• 사과 • 바나나 • 키위

7. HTML 문서의 표

■ <table> 태그의 속성

속성	의미
align="정렬"	표의 정렬 방식 지정(left,center,right)
bgcolor="색"	표의 배경색
border="n"	표 테두리 두께 지정 (0 값을 사용하면 테두리를 그리지 않는다)
cellspacing="n"	셀과 셀간의 간격 지정
cellpadding="n"	셀 내부와 테두리와 내용간의 간격 지정
width="px"	표의 너비
height="px"	표의 높이

■ <table>태그 안에 들어가는 태그들

- <caption> : 표에 대한 설명을 넣기위한 태그
- <th> : 표의 헤더를 지정하기 위한 태그
- <td> : 표의 셀을 지정하기 위한 태그
- <tr> : 표의 한 행을 지정하기 위한 태그

7. HTML 문서의 표

- <th> 태그의 속성

<th>태그의 속성	속성의 역할
align	헤더의 정렬방식을 지정하는 속성. left,,center,right중 한가지 사용
rowspan	헤더의 셀이 몇 개의 행을 병합 한 것인지에 대한 속성
colspan	헤더의 셀이 몇 개의 열을 병합 한 것이지에 대한 속성

8. HTML 문서의 링크와 이미지 사용

<a> 태그

- 하이퍼텍스트 연결을 위해 사용하는 것으로 같은 문서의 특정 위치나 다른 문서, 다른 웹사이트 문서를 연결하기 위해 사용된다.
- **다른 사이트로 연결**

```
<a href="http://cafe.naver.com/sunschool>썬스쿨바로가기</a>
```

- **문서내에서 연결**

```
<a href="#test">테스트문으로 이동</a>
<a name="test">여기는 테스트문입니다.</a>
```

- **같은 사이트 내에서 이동**

```
<a href="spring/hello.htm">인사 화면으로 이동합니다.</a>
```

<img> 태그

- HTML문서에서 그림을 삽입하기 위해 사용한다.

```
<img src="screen.jpg"  align="top"> 글자의 위치가 그림의 위쪽에  정렬<br/>
<img src="screen.jpg"  align="middle"> 글자의 위치가 그림의 가운데 정렬<br/>
<img src="screen.jpg"  align="bottom"> 글자의 위치가 그림의 아래쪽에 정렬 <br/>
```

- 이미지를 통한 문서 연결

```
<a href="http://www.sun.com"><img src="sun.jpg"></a>
```

9. HTML에서 입력양식(form) 만들기

- <form> 태그
  - 입력 양식을 사용하려면 먼저 <form>태그를 만들어야 한다.
  - action 속성을 사용하여 사용자가 입력한 데이터를 처리할 프로그램의 URL을 지정한다.
  - method 속성을 사용하여 입력값을 헤더에 붙여서 보내는 “get”방식 이거나 입력값을 body에 붙여서 보내는 “post” 방식중 하나를 선택한다. 지정하지 않으면 “get”방식으로 입력값이 전송된다.

```
<form action=“hello.do” method=“post”>
</form>
```

<input> 태그를 이용한 입력처리(1)

<input type="" name="" value="" size="" maxlength="">

- type : 입력 형식 지정
  - text : 문자열 입력
  - password : 암호를 입력할 때 사용
  - checkbox : 체크박스 형태로 여러 개 값을 선택할 수 있다.
  - radio : 여러 개 중 하나의 값만 선택할 수 있다.
  - submit : 폼에서 기술한 내용을 웹 서버로 전송한다.
  - reset : 폼에 기술한 내용들을 초기 값으로 변경한다.
  - image : 이미지 맵을 이용하여 이미지에서 마우스가 클릭된 곳의 위치를 리턴 한다.
  - hidden : 화면에 보이지 않는 값을 웹 서버에 전달할 때 사용한다.
  - button : 클릭할 수 있는 버튼을 나타낸다.
  - file : 파일을 선택할 수 있는 창을 나타낸다. 파일을 업로드 할 때 사용한다.
- name : 현재 input 태그의 이름을 기술한다.
- size : 텍스트 입력양식의 길이를 지정/디폴트는 "20"이다
- maxlength : 사용자가 최대로 입력할 수 있는 글자의 수
- value : 텍스트 입력양식에 미리 입력하고 싶은 문자열 지정
- checked : type이 checkbox나 radio의 경우 디폴트로 선택됨을 의미한다.
- readonly : type이 text나 password의 경우 읽기 전용으로 만든다.

<input> 태그를 이용한 입력처리(2)

- 텍스트 박스

```
<input type="text" name="email"  size="20" maxlength="20"  value="이메일주소입력">
```

- 암호 입력 양식

```
<input type="password" name="passwd">
```

- 체크박스(여러 개 선택가능)

```
<input type="checkbox" name="device" value="disk">디스크
<input type="checkbox" name="device" value="cd">CD/DVD
<input type="checkbox" name="device" value="usb" checked>USB Memory
```

- 라디오버튼(한 개만 선택가능)

```
<input type="radio" name="device" value="disk"  checked>디스크
<input type="radio" name="device" value="cd">CD/DVD
<input type="radio" name="device" value="usb">USB Memory
```

- 버튼

```
<input type="button" name="ok" value="확인"  onclick="check();">
```

<input> 태그를 이용한 입력처리(3)

- 데이터 보내기 버튼

```
<input type="submit" name="ok" value="데이터보내기" >
```

- 리셋버튼

```
<input type="reset" value="입력취소" >
```

- 이미지 입력 양식

```
<input type="image"  name="duke"  src="duke.jpg" >
```

- 숨겨진 입력 양식

```
<input type="hidden"  name="id" value="123456" >
```

<select>, <option> 태그

- select 태그는 Drog-down 형태의 메뉴에서 하나 혹은 여러 개의 아이템을 선택할때 사용된다. 아이템들을 표현하기 위해 option 태그를 사용한다.
- select 태그의 속성
  - name : 현재 select 태그의 이름
  - size : 한번에 보일수 있는 <option>태그의 아이템의 수
  - multiple : 한번에 여러 개를 선택할 수 있는지 여부를 기술한다. multiple선언되어 있으면 한번에 여러 개의 아이템을 선택할 수 있다.
- option 태크의 속성
  - value : 서버로 전달될 값을 기술한다. 값이 기술되어 있지 않으면 option 태그의 내용이 값으로 전달된다.
  - selected : select 태그에서 현재 option이 디폴트로 선택되었음을 표시한다.
  - disabled: 특정 목록을 지정할 수 없도록 할 때 사용된다.
- 사용예

```
<form action="nothing.cgi">
      <select name="animal">
         <option>개
         <option>고양이
          <option selected>말
      </select >
</form>
```

<texarea>태그

- 여러 줄에 걸쳐 문자열을 입력 받을때 사용된다.

```
<textarea name="" rows="" cols=""></textarea>
```

- textarea 속성
  - name : 현재 textarea 태그의 이름
  - rows : 화면에 보일 라인 수 지정
  - cols : 화면에 보일 문자의 수 지정
  - readonly : 읽기전용으로 만든다.
  - onfocus : 마우스의 포커스를 받았을 때 실행할 자바 스크립트 함수를 기술한다.
  - onblur : 마우스가 포커스를 잃었을 때 실행할 자바 스크립트 함수를 기술한다.
  - onselect : type이 text인 경우 혹은 textarea에서 텍스트 내용을 마우스로 선택한 경우 실행할 자바 스크립트
    함수를 기술한다.
- 사용예

```
<form action="nothingcgi">
    <textarea name="test" rows="4" cols="30">
      이곳에 내용을 입력하세요.
    </textarea>
</form>
```

<input> 태그에서 사용 가능한 이벤트들

- onfocus : 마우스의 포커스를 받았을 때 실행할 자바 스크립트 함수를 기술한다.
- onblur : 마우스가 포커스를 잃었을 때 실행할 자바 스크립트 함수를 기술한다.
- onselect : type이 text인 경우 혹은 textarea에서 텍스트 내용을 마우스로 선택한 경우 실행할 자바 스크립트 함수를 기술한다.
- onchange : 마우스의 포커스를 잃고 내용이 변경된 경우에 실행할 자바 스크립트를 기술한다.
- onclick : 마우스가 클릭되었을때 실행할 자바 스크립트 함수를 기술한다.

```
<input type="button" value="누르세요"  onclick="sayHello();">
```

로그인을 처리하는 login.html 예제

```
<HTML>
   <script type="text/javascript" >
   <!--
     function check(){
        if(Loginform.uid.value =="") {
             alert("id를 입력하세요")
            return false;
         }
        else if(Loginform.pwd.value =="") {
             alert("pwd를 입력하세요")
             return false;
        }
        else {    return true;    }
     }
   //-->
   </script>

   <body>
   <FORM name="Loginform" METHOD="post" ACTION="login.jsp" >
      로그인명 :<INPUT TYPE="text" NAME="uid">
      패스워드 :<INPUT TYPE="password" NAME="pwd">
     <INPUT TYPE="image" src="submit.gif" onclick="return check()">   //type=submit 이나 type=image 둘 다 submit 버튼이다. 그래
   서 return으로 처리
     <INPUT TYPE="reset" value="재작성">
   </FORM>
   </body>
   </HTML>
```

CSS (Cascading Style Sheet)

yskim@suned.co.kr
http://cafe.naver.com/sunschool

1. CSS란?

- cascading style sheets의 약어로 스타일 시트라고 합니다.
- 웹 문서의 스타일을 저장해 둔 파일입니다.
- HTML,XHTML 문서에서 자주 사용하는 서체나 색상, 정렬, 각 요소들의 배치 등의 유형에 대한 다양한 스타일을 한군데 모아놓은 것을 '스타일 시트'라고 합니다.
- 유지 보수가 아주 간편하게 된다는 이점이 있고 브라우저를 가리지 않습니다.

2. 스타일 시트를 사용해야 하는 이유

- **웹 문서의 디자인과 내용을 분리할 수 있습니다.**
  CSS를 이용해서 웹문서를 디자인할 경우 HTML 소스 부분은 그대루 두고 CSS 소스만 수정 하면 되기 때문에 작업이 훨씬 효율적이다.

- **다양한 매체에 적합한 문서를 만들 수 있습니다.**
  HTML로 작성된 내용은 그대로 두고 CSS에서 대상 매체(Media)가 프린터인지, 모바일 기기인지만 지정해주면 같은 내용을 여러 매체에서 사용할 수 있다.

3. CSS 명령어의 구조

■ **선택자 { 프로퍼티 : 값 }**

ex) body { color : black }

- 선택자 : 주로 태그가 기술됨. 이 태그에 프로퍼티와 값이 적용됨
- 프로퍼티 : 왼쪽의 선택자에 적용될 속성을 기술함(프로퍼티 = 속성)
- 값 : 프로퍼티의 값. 즉, 속성값을 기술함. 이 값은 프로퍼티마다 다름

①. 모든 CSS 구문은 위와 같이 선택자, 프로퍼티, 값으로 구성

②. 프로퍼티와 값은 중괄호 { }내에 기술되며, 콜론(:)으로 구분합니

③. 값이 2개 이상의 단어로 구성되면 따옴표 내에 기술합니다.

```
p {font-family:"sans serif"}
```

④. 하나의 태그에 여러 개의 프로퍼티:값 쌍을 지정할 경우는 값 프로퍼티:값 쌍을 세미콜론(;)으로 구분합니다.

```
p {text-align:center; color:red }
```

⑤. 여러 개의 프로퍼티를 지정할 경우, CSS 구문을 읽기 쉽도록 한 개 라인에 한 개의 프로퍼티:값 쌍을 기술할 수도 있습니다.

```
p {
        text-align: center;
        color: black;
        font-family: Arial, 바탕, 굴림
   }
```

4. CSS 를 이용하는 3가지 방법

- CSS 를 이용하는 3가지 방법

  1. <head> ... </head> 사이에 기술(내부 파일 방식) [우선순위 1]

  2. HTML 태그 내에 직접 기술(인라인 방식) [우선순위 2]

  3. 별도의 .css 파일에 기술(외부 파일 방식) [우선순위 3]
     @import! 문 이용

❖ CSS 입력 위치(1)

- <head> … </head> 사이

```
<head>
    <style type="text/css">
        body {background-color: yellow}
        h1 {font-size: 36pt}
        h2 {color: blue}
        p {margin-left: 50px}
    </style>
 </head>
```

❖ CSS 입력 위치(2)

- html 태그 내에 직접 기술(인라인 방식)

```
<body style="backgrount-color: yellow">

  <h1 style="font-size: 36pt"> 이 제목은 36 포인트입니다.</h1>
  <h2 style="color: blue"> 이 제목은 푸른색입니다.</h2>
  <p style="margin-left: 50px">이 문장은 왼쪽 여백이 50픽셀 입니다.</p>

</body>
```

❖ CSS 입력 위치(3)

- 별도의 .css 파일에 기술(외부 파일 방식)

```
<head>
   <link rel="stylesheet" type="text/css" href="style12.css" />
 </head>

# cat   style12.css
   body {background-color: yellow}
   h1 {font-size: 36pt}
   h2 {color: blue}
   p {margin-left:50px}
```

❖ CSS 입력 위치(4)

- @import! 문 이용

```
<style type="text/css">
    @import! "외부 스타일 시트 파일"
</style>
```

- @import! 문 사용시 <link> 태그와 함께 사용

```
<link rel="stylesheet" href="style1.css">
<style type="text/css">
    @import! "style2.css"
</style>
```

❖ 주석 사용하기

■ 주석 사용하기

/* 와 */ 사이에 기술한다(여러 줄 주석)

```
<style type="text/css">
  /* 이것은 주석입니다. */
 p {
      text-align: center;
          /* 이것도 주석입니다. */
      color: black;
      font-family: arial
 }
```

5. 스타일시트의 CLASS 속성 정의하기

- **CLASS 속성 사용하기 (Tag.class name)**

class 속성은 두 가지 경우에 유용하다.
첫 번째는 하나의 태그에 여러 개의 스타일을 적용하는 경우이고,
두 번째는 여러 개의 태그에 하나의 스타일을 적용하는 경우이다.

```
<style type="text/css">
      p.right {text-align: right}        -> P 태그 하나에 두 개의 스타일을 정의함
      p.center {text-align: center}
</style>

<p class="right"> 이 문장은 오른쪽으로 정렬됩니다. </p>
<p class="center"> 이 문장은 가운데로 정렬됩니다. </p>

<style type="text/css">
   .center {text-align: center}
</style>

<h1 class="center"> 이 제목은 가운데로 정렬됩니다.</h1>
<p class="center">이 문장도 가운데로 정렬됩니다.</p>
```

6. 스타일 시트의 ID 속성 사용하기

- ID 속성 사용하기 (#ID name)

  ID 속성은 CLASS 속성과 유사하다. ID 속성도 두 가지 경우에 유용하다.
  첫 번째는 동일한 ID를 가진 모든 태그들에 동일한 스타일이 적용되는 경우이고,
  두 번째는 동일한 태그라도 해당 ID를 가진 경우만 스타일이 적용된다.

```
<style type="text/css">
#intro   {
   font-size:110%;
   font-weight:bold;
   color:#0000ff;
   background-color:transparent
}
</style>

<h1 id="intro"> 이 제목은 intro 스타일로 표시됩니다. </h1>
<p id="intro"> 이 문장도 intro 스타일로 표시됩니다.</p>
```

7. 폰트 지정을 위한 스타일 시트

■ 폰트크기: font-size(css)/fontSize(javascript)

크기 지정 방법	의미/지정할 수 있는 값	사용예
절대크기	폰트의 절대적 크기(서로 1.5배 차이) x-small, small, medium,large,x-large,xx-large	h3 { font-size: small; }
상대크기	상위 요소에 대한 상대적 크기 Larger, smaller	h3 { font-size: larger; }
숫자	폰트 크기	h3 { font-size: 12px; }
백분율	상위 요소에 대한 상대적 크기	h3 { font-size: 150%; }

■ 폰트 종류: font-family/fontFamily

- 하나이상의 폰트를 나열할 수 있으며, 이 중 컴퓨터에서 사용 가능한 첫번째 폰트가 사용된다.
- serif, sans-serif,굴림,맑은고딕,Arial 등등

■ 폰트 굵기: font-weight/fontWeight

- normal, bold, bolder,lighter 을 이용한 절대 굵기를 지정하거나 100에서 900사이의 숫자를 사용하여 지정할 수 도 있다.

7. 폰트 지정을 위한 스타일 시트(2)

- 폰트 스타일: font-style/fontStyle
  - normal, italic, oblique 를 값으로 사용할 수 있다.

- 폰트 밀도: font-strech/fontStrech
  - ultra-codensed, extra-codensed, codensed, semi-codensed,
  - Normal, semi-expanded, expanded, extra-expanded, ultra-expanded
  - Wider, narrower

- 폰트 색: color/color

```
h3 { color: green; }
```

- 폰트: font/font
  - 여러가지 폰트 속성을 한번에 지정할 때 사용
  - font속성은 font-style,font-weight,font-size,font-family 순서로 지정

```
h3 {
        font:  normal, bold, 12px, arial;
}
```

8. 텍스트 지정을 위한 스타일 시트(1)

- 줄 높이: line-height/lineHeight
  - 각 줄 사이 간격을 지정하기 위해 사용하는 속성
  - line-height: 1.5 <- 현재 폰트의 크기에 대한 비율을 의미
  - 길이: 절대 길이를 의미
  - 백분율: 상위 요소에 대한 상대적인 크기

- 텍스트 장식: text-decoration/textDecoration
  - 문자에 더해질 수 있는 여러가지 효과를 지정하기 위해 사용하는 속성

속성	효과
none	텍스트 효과 없음
underline	밑줄
overline	윗줄
line-through	텍스트 가운데를 관통하는 줄
blink	깜빡임

8. 텍스트 지정을 위한 스타일 시트(2)

- 수직 정렬: vertical-align/verticalAlign
  - 텍스트의 수직 정력방식을 지정하기 위해 사용하는 속성

속성	효과
baseline	상위 요소의 베이스 라인에 정렬
super	위 첨자
sub	아래 첨자
top	가장 긴 요소의 상단에 정렬
bottom	가장 작은 요소의 하단에 정렬
text-top	상위 요소의 상단에 텍스트 상단을 정렬
text-bottom	상위 요소의 하단에 텍스트 하단을 정렬
middle	텍스트 중간 부분으로 정렬
백분율	line-height에 대한 비율

8. 텍스트 지정을 위한 스타일 시트(3)

- 텍스트 변환: text-transform/textTransform
  - 텍스트 변환 방식지정
    - none : 변환없음
    - capitalize : 각 단어의 첫번째 글자를 대문자로 변환
    - uppercase : 모든 글자들을 대문자로 변환
    - lowercase: 모든 글자들을 소문자로 변환

- 텍스트 정렬: text-align/textAlign
  - 텍스트의 수평 정렬 방식 지정
    - left : 좌측 정렬
    - right: 우측정렬
    - center : 가운데 정렬
    - justify : 고른 정렬

- 텍스트 들여 쓰기: text-indent/textIndent
  - 각문단의 첫번째 글자를 앞에서 얼마나 들여쓸지를 지정
    - 숫자: 글자가 들어가는 크기
    - 백분율: 상위 요소의 너비에 대한 비율

8. 텍스트 지정을 위한 스타일 시트(4)

- 글자 간격: letter-spacing/letterSpacing
  - 글자간의 간격지정
    - Normal : 일반크기
    - 숫자: 글자 간격 길이, 음수도 사용가능

      p { letter-spacing: 2px; }

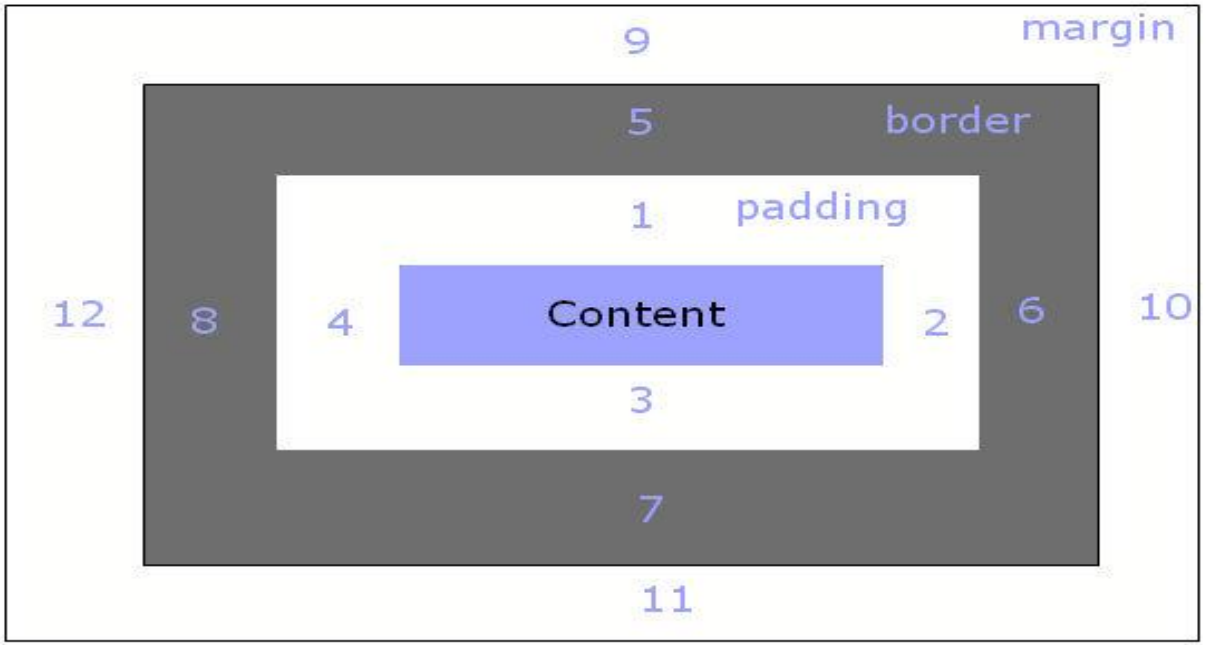
- 단어 간격: word-spacing/wordSpacing
  - 단어 사이 간격 지정
    - Normal : 일반크기
    - 숫자: 글자 간격 길이, 음수도 사용가능

      p { word-spacing: -2px; }

- 줄바꾸기 설정: white-spacing/whiteSpacing
  - 줄 바꾸기를 어떻게 할 것인지를 지정
    - normal: 문자열이 주어진 너비를 넘어서는 경우 자동으로 줄바꿈
    - nowrap: 줄바꾸기를 허용하지 않음
    - pre: <PRE>태그처럼 모든 공백을 그대로 인정

9. 텍스트 포맷 지정을 위한 스타일 시트(1)

- Box Position
- Margin(바깥 여백), Boder(선), Padding(안쪽 여백) 명칭



[1]. padding-top [2].padding-right [3]. padding-bottom [4].padding-left

[5]. border-top [6]. border-right [7].border-bottom [8]. border-left

[9]. margin-top [10]. margin-right [11]. margin-bottom [12].margin-left

9. 텍스트 포맷 지정을 위한 스타일 시트(2)

여백

- 텍스트의 테두리 여백을 주기 위해 margin-left, margin-right, margin-top, margin-bottom 속성을 사용.
- margin을 이용하여 상우하좌 순서의 여백을 한번에 지정

```
<style type="text/css">
  .main10 {
        margin-top: 8px;
        margin-right: 9px;
        margin-top: 10px;
        margin-bottom:  7px;
   }
</style>
```

테두리와 내용간의 거리

- 텍스트의 테두리와 내용간의 거리를 지정하기위해 padding-left, padding-right, padding-top, padding-bottom 속성을 사용
- padding을 이용하여 상우하좌 순서의 간격을 한번에 지정

```
<style type="text/css">
    .main10 {
        padding-top: 8px;
        padding-right: 9px;
        padding-top: 10px;
        padding-bottom:  7px;
    }
</style>
```

9. 텍스트 포맷 지정을 위한 스타일 시트(3)

- 테두리의 두께
  - 텍스트의 테두리 여백을 주기 위해 border-left-width, border-right-width, border-top-width, border-bottom -width속성을 사용.
  - Border-width을 이용하여 상우하좌 순서의 테두리 두께를 한번에 지정
  - 속성값은 숫자나 "thin", "medium", "thick" 를 사용
  - 숫자의 단위에 "em"은 현재 폰트의 크기를 의미, 2em 이면 현재 폰트의 2배 크기

- 테두리 스타일
  - 테두리 스타일을 지정하기위해 border-left-style, border-right-style, border-top-style, border-bottom -style속성을 사용
  - Border-style을 이용하여 상우하좌 순서의 간격을 한번에 지정

```
p.solid {border-style: solid }          >선 테두리
p.double {border-style: double}         >두줄
p.dashed {border-style: dashed}         >파선
p.dotted {border-style: dotted}         >점선
p.groove {border-style: groove }        >3차원 그루브
p.ridge {border-style:ridge}            >3차원 ridge
p.inset {border-style: inset}           >3차원 inset
p.outset {border-style: outset }        >3차원 outset
p.none {border-style: none }            >테두리 없음
p.hidden {border-style: hidden}         >테두리 없음
```

9. 텍스트 포맷 지정을 위한 스타일 시트(4)

- 테두리 색
  - 테두리 색을 지정하기 위해 border-left-color, border-right-color border-top-color, border-bottom -color속성을 사용.
  - Border-color을 이용하여 상우하좌 순서의 테두리 두께를 한번에 지정

- 텍스트 너비와 높이: width,height
  - Width와 height는 텍스트의 너비와 높이를 지정하기 위해 사용

- **텍스트 배치: align**
  - 정렬방식을 지정

```
img {
        align: left;
}
```

- **비우기: clear**
  - 텍스트나 이미지가 왼쪽이나 오른쪽에 나타나지 못하도록 만들 때 사용하는 속성

10. 기타 스타일 시트(1)

- 배경색 : background-color
  - 배경색 지정
- 배경그림: background-image
  - 배경 그림 지정
- **요소 종류: display**
  - HTML요소를 어떤식으로 보여줄 지를 지정
    - none: 요소 감추기
    - block: block 모양
    - inline: inline모양
    - list-item: 목록 타입 모양
- **목록 마크 타입: list-style-type**
  - 태그에서 목록의 마크를 어떤 모양으로 보여줄 지 지정

    none: 없음
    disc: 디스크 모양
    circle: 원 모양
    square: 사각형 모양
    decimal: 숫자
    lower-roman: 로마자 소문자
    upper-roman: 로마자 대문자
    lower-alpha: 소문자 알파벳
    upper-alpha: 대문자 알파벳

10. 기타 스타일 시트(2)

- 목록 마크 이미지: list-style-image
  - 태그에서 목록의 마크를 이미지로 지정하기 위해 URL 할당
- 요소 감추기: visibility
  - 요소를 감추고자 할때 사용하는 속성
  - visible이면 보여주소, hidden이면 요소를 감춘다
- **위치 지정 방식: position**
  - HTML요소의 위치를 지정하는 방식 지정
    - static: 디폴트값. 위치 지정 없이 HTML 요소들이 순서대로 나오게 됨
    - relative: 현재 위치에서 상대적인 거리 만큼 이동
    - absolute: 상위 요소를 기준으로 지정된 특정 위치에 표시
    - fixed: 브라우저를 기준으로 지정된 위치에 표시

11. <a> 태그 설정과 관련된 CSS

- 문서가 처음 표시될 때
  a:link { color:black; text-decoration:none; font-family:돋움; font-size:9pt;}

- 링크가 한번이라도 클릭 된후
  a:visited{ color:red; text-decoration:none; font-family:돋움; font-size:9pt;}

- 링크를 클릭하고 나면
  a:active{ color:blue; text-decoration:none; font-family:돋움; font-size:9pt;}

- 커서가 링크 위로 올라왔을 때
  a:hover { color:green; text-decoration:underline; font-family:돋움; font-size:9pt; background-color:#0066cc }

Border Style 예제

- Border Style 2 (border width, style, color, boder-right-color)

```
<style type="text/css">
    p.one     {
    border-style: solid;
    border-width: 5px;
    border-color: blue
    /*border-right-color: #ff0000*/
    }

    p.two  {
      /* 색상 두개 지정할 경우, 먼저 색은 위아래 선색, 다음 색은 좌우 선색 */
    border-style: solid;
    border-width: 10px;
    border-color: blue yellow
    /*border-right-color: #ff0000*/
    }
```

❖ Border Style 예제

```
p.three  {
    /* 색상 세개 지정할 경우, 먼저 색은 위, 다음 색은 좌우 색,
       다음 색은 아래로 나타난다. */
    border-style: solid;
    border-width: 15px;
    border-color: blue yellow red
    /*border-right-color: #ff0000*/
    }

p.four   { /* 시계 방향으로 적용됨 */
    border-style: solid;
    border-width: 20px;
    border-color: blue yellow red gray
    /*border-right-color: #ff0000*/
    }
    </style>
```

❖ 테이블에 CSS 적용하기(1)

- 테이블에 CSS 적용하기

```
<HTML>
<HEAD>
<TITLE> 테이블에 스타일시트 적용하기  </TITLE>

    <STYLE type="text/css">

    TABLE {
        border-style:double;        /*테두리 종류: double*/
        border-width:5px;           /*테두리 굵기: 5px*/
        border-color:#006600        /*테두리 색: #006600*/
     }
```

❖ 테이블에 CSS 적용하기(2)

```
TH {
  border-style:solid;          /*테두리 종류: solid*/
  border-top-width:0px;   /*border-top-width :위쪽 테두리 굵기: 0px*/
  border-bottom-width:5px;   /*아래쪽 테두리 굵기: 5px*/
  border-right-width:0px;      /*오른쪽 테두리 굵기: 0px*/
  border-left-width:0px;       /*왼쪽 테두리 굵기: 0px*/
  border-color:#006600;        /*테두리 색: #006600*/
  padding:5px 5px 5px 5px; /*padding :셀안의 데이터와 테두리 간 여백*/
  background-color:#ffffff;    /*배경색 : #ffffff*/
  color:#006699;               /*글자색: #006699*/
  font-size:15pt;              /*폰트크기: 15pt*/
  font-family:바탕;            /*폰트체: 바탕*/
  letter-spacing:10px          /*글자 간격: 10px*/
}
```

❖ 테이블에 CSS 적용하기(3)

```
TD {
  border-style:dotted; /*테두리 종류: dotted*/
  border-top-width:0px; /*border-top-width :위쪽 테두리 굵기: 0px*/
  border-bottom-width:1px; /*아래쪽 테두리 굵기: 1px*/
  border-right-width:0px; /*오른쪽 테두리 굵기: 0px*/
  border-left-width:0px; /*왼쪽 테두리 굵기: 0px*/
  border-color:#006600; /*테두리 색: #006600*/
  padding:5px 5px 5px 5px; /*padding :셀안의 데이터와 테두리 간 여백*
/
  line-height:150%; /*칸 높이 : line-height: 150%*/
  font-size:15pt; /*폰트크기: 15pt*/
  font-family:바탕7 /*폰트체: 바탕*/
}
</STYLE>
</HEAD>
```

Java Script

yskim@suned.co.kr
http://cafe.naver.com/sunschool

1. 자바 스크립트란?

- 자바스크립트는 네스케이프사가 개발한 객체지향 스크립트언어로 HTML문서내에 작성하고 웹브라우저에 의해 실행된다

- 자바 스크립트 작성방법

  • HTML문서 내에 작성할 때

```
<SCRIPT LANGUAGE="JavaScript">
        자바스크립트 코드
</SCRIPT>
```

  • HTML문서 외부에 따로 둘 때

```
<SCRIPT LANGUAGE="JavaScript" SRC="aaa.js"></SCRIPT>
```

  확장자는 js이고 텍스트문서이다. 다른 사이트의 js파일도 가능하다
  (http://www.dragoneye.co.kr/aaa.js)
  웹문서를 간결하게 해주고 여러문서가 공통으로 js파일을 사용할 수 있다.

  • 자바스크립트를 이해하지 못하는 브라우저를 위해 <!-- -->

```
<SCRIPT LANGUAGE="JavaScript">
<!--
        자바스크립트 코드
-->
</SCRIPT>
```

2. 자바스크립트 위치

- HEAD 태그내에

```
<HEAD>
 <SCRIPT LANGUAGE="JavaScript">
      자바스크립트 코드
 </SCRIPT>
 </HEAD>
```

- BODY 태그 위에

```
<HEAD><title>~~~</title></HEAD>
 <SCRIPT LANGUAGE="JavaScript">
       자바스크립트 코드
 </SCRIPT>
 <BODY>
```

- 필요시 문서내 적당한 곳에

```
</table>
<SCRIPT LANGUAGE="JavaScript">
<!--
      자바스크립트 코드
-->
</SCRIPT>
<table>
```

3. 자바 스크립트 기본 문법

- **변수**

  var 변수명 [=초기값][,변수명[=초기값]]

  var firstName;

  var lastName="김";

- **배열**

  var 배열명=new Attay(); 배열명.push(1); 배열명[0]=1;

  var a=new Array();

  a[0]=1;

- **산술연산자**

  +, -, *, /, %, ++, --

- **대입연산자**

  = , += , -= , *= , /= , %=

- **비교연산자**

  != , == , > , < , >= , <=

- **논리연산자**

  && , || , !

- **조건연산자**

  (조건) ? 조건이 참일때의 값 : 조건이 거짓일때의 값; ex) b= (a>10)? 3 : 0 ;

자바스크립트의 구문과 함수

- if문

```
if  ( a>b ) {
      document.write("a가 b보가 큽니다");
} else if ( a == b) {
      document.write("a와 b는 값습니다.");
} else {
      document.write("a가 b보다 작습니다.");
}
```

- switch 문

```
switch ( a % 2 ) {
      case 0 :
            document.write("짝수"); break;
      case 1 :
            document.write("홀수"); break;
      default:
            document.write("숫자가 아닙니다."); break;
}
```

자바스크립트의 구문과 함수

- for문

```
for ( var  a=1; a < 10 ; a++ ) {
        sum+=a;
}
```

- while 문

```
while ( a <= 10 ) {
    sum += a;
    a++;
}
```

- try ~ catch문

```
try {
     document.write(name+"입니다");
} catch(e) {
     document.write(e.message);
}
```

자바스크립트의 구문과 함수

- 매개 변수가 없는 함수

```
<script   language="javascript">
      function sayHello() {
         document.write("안녕하세요?");
      }
</script>
```

메서드 이름과 변수이름이 반드시 달라야 한다.

- 매개 변수가 있는 함수

```
<script   language="javascript">
      function sayHello(name) {
         document.write(name+"님 안녕하세요?");
      }
</script>
```

- 리턴값이 있는 함수

```
<script   language="javascript">
      function sayHello(name) {
         return name+"님 안녕하세요?";
      }
</script>
```

함수 선언시 리턴 타입을 표시 하지 않음

자바스크립트로 웹브라우저 창 제어

- alert() 사용자에게 메시지를 보여주기

```
<script   language="javascript">
      alert("안녕하세요?");
</script>
```

- prompt() 사용자에게 질문하기

```
<script   language="javascript">
    var  name= prompt("이름을 입력하세요");
    alert(name+"님 환영합니다");
</script>
```

- confirm() 사용자의 선택 알아내기

```
<script   language="javascript">
      var result = confirm("삭제하시겠습니까? ");
      if  ( result ) {
            alert("삭제합니다");
      } else {
            alert("삭제작업 취소");
      }
</script>
```

자바스크립트로 웹브라우저 창 제어

- open() 팝업창 띄우기

```
<script   language="javascript">
      window.open("http://www.sun.com","새로운창","left=0,top=0,width=300,height=500,
      location=yes,status=no");
</script>
```

- setTimeout() 일정시간이 지나면 함수를 실행하기

```
<script   language="javascript">
      var timer = setTimeout(sayHello,3000)
</script>
```

3초뒤에 sayHello 함수 호출
clearTimeout(timer); 함수호출 동작을 중단한다.

- setInterval() 일정 시간 동안 특정 함수를 반복 실행하기

```
<script   language="javascript">
   var  timer = setInterval(sayHello, 5000);
</script>
```

5초 간격으로 sayHello 함수 호출
clearTimeout(timer); 함수호출 동작을 중단한다.

4. 자바 스크립트 함수

- **함수 정의 하기**

```
function 함수이름(매개변수1, 매개변수2,. . . ) {
        함수 문장
}
```

- **반환값이 있는 경우**

```
function testFunction(var1) {
      var var2;
      var2=confirm(var1);
      return  var2;
}
```

- **자바 스크립트 함수 호출하기**
  - 다른 자바스크립트 구문에서 앞의 예제와 같이 명시적으로 호출하여 실행한다.
  - 링크에 의해 실행하기
  - body 태그에서 <body onLoad="testA()"> 페이지 로딩시 실행 된다.
  - onClick , onMouseOver 등등과 같은 다양한 이벤트에 의해 에 의해 실행된다.

5. 자바 스크립트 객체

- **객체 정의하기**

  자바 스크립트에서 객체는 생성자함수(constructor function)라는 것을 통해 정의

```
function computer(cpu,memory,disk) {
    this.cpu=cpu;
    this.memory=memory;
    this.disk=disk;
}
```

- **객체 만들기**

```
var myCom=new computer("2 core","8G","500G");
```

- **객체의 메서드 정의하기**

```
function computer(cpu,memory,disk) {
    this.cpu=cpu;
    this.memory=memory;
    this.disk=disk;
    this.print = function() {
        document.write("CPU: "+this.cpu+ "  메모리: "+this.memory+"  디스크: "+this.disk+"<p>");
    }
```

6. 자바 스크립트 내장 객체

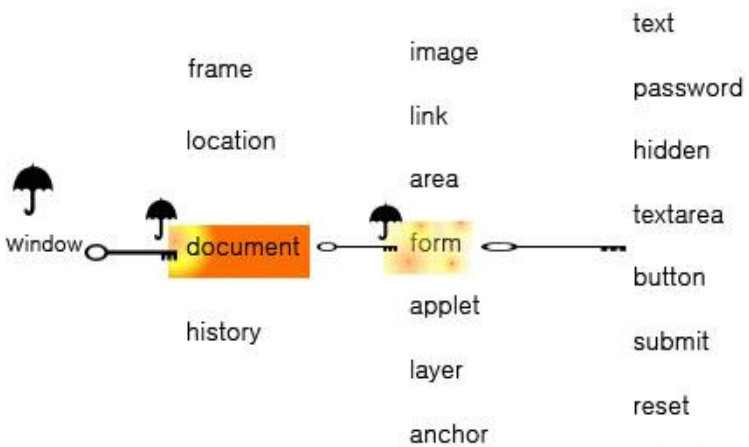
- **자바 스크립트 내장 객체란?**
  자바 스크립트에서 사용할 수 있도록 미리 만들어진 객체.
  String, Date, Math, Array 같은 객체가 많이 사용되는 내장 객체이다.

- **객체의 메서드 정의하기**

```
function computer(cpu,memory,disk) {
    this.cpu=cpu;
    this.memory=memory;
    this.disk=disk;
    this.print = function() {
        document.write("CPU: "+this.cpu+ "  메모리: "+this.memory+"  디스크: "+this.disk+"<p>");
    }
```

브라우저내장객체

- 익스플로러나 네비게이터 등의 브라우저가 내장하고 있는 객체를 자바스크립트를 이용하여 이용할 수 있다. 다음은 웹문서에서 기본적으로 만들어지는 내장객체이다. 이 부분이 자바스크립트의 상당부분을 차지할 정도로 중요하고 이용빈도 또한 아주 높다.



브라우저 내장 객체

내장객체	설 명
window	브라우저 창이 열릴 때마다 하나씩 만들어지는 객체이다. 브라우저 창 안에 존재하는 모든 요소의 최상의 객체이다.
navigator	현재 사용중인 브라우저에 대한 정보를 가지고 있는 객체이다.
frame	프레임마다 하나씩 만들어지는 객체로 , <FRAME>태그를 쓸 때마다 하나씩 만들어진다.
document	웹 페이지마다 하나씩 만들어지는 객체로 , <BODY>태그에 의해 만들어진다. HTML문서에 대한 정보를 가지고 있다.
location	현재 페이지에 대한 URL 정보를 가지고 있는 객체이다.
history	현재 창에서 사용자의 방문기록을 저장하고 있는 객체이다.
image	웹 페이지에 삽입된 이미지 정보를 가지고 있는 객체로 <IMG> 태그를 쓸 때마다 객체가 하나씩 만들어진다.
link	웹 페이지의 하이퍼링크 정보를 가지고 있는 객체로 , <A Href = "...">태그를 쓸 때마다 만들어진다.

window 객체 메서드

- alert() : 경고용 대화상자를 보여줌
- confirm() : 확인, 취소를 선택할 수 있는 대화상자를 보여줌
- open() : 새로운 창을 오픈

```
<script language="javascript">
  function winOpen() {          // 페이지로딩시 새창 열기
        window.open("123.html","newWin","width=300,height=200,toolbar=no")
  }
</script>
<body onLoad="winOpen()">
     <font onClick="winOpen()"> 클릭열기 </font>
      <a href="javascript:winOpen()"> 링크열기 </a>
</body>
```

- prompt() : 입력창이 있는 대화상자를 보여줌
- blur() : focus를 이동
- back()한 단계 전 URL(이전화면)로 돌아감. 익스플로러 지원 안함

window 객체 이벤트 핸들러

- onBlur : 브라우저가 포커스를 잃을 때 발생
- onError : 문서를 읽는 중에 에러가 생길 때 발생
- onFocus : 브라우저에 포커스를 얻을 때 발생
- onLoad : 문서를 읽을 때 발생
- onUnload : 현재 문서를 지울려고 할 때 발생

document 객체

■ 최상위 window객체에서 파생된 것으로 <body>~</body> 태그속의 내용과 관련된 처리를 할 수 있다.
많은 기능이 있지만 이 객체는 범위가 넓어서 직접 다루는 일은 별로 없고 실제 이용은 이 객체에서 파생된 하위객체(Layer, Link, Image, Area, Anchor, Applet, Form)를 통해 세세한 접근을 할 수 있다.
상위 객체인 window를 생략하고 사용한다.

- 현재문서의 URL

```
<script language="javascript">
    alert(document.location)
</script>
```

- 이전문서의 URL

```
<script language="javascript">
    alert(document.referer)        // 이 파일을 링크한 페이지에서
</script>                          // 넘어와야 값이 있다.
```

Document 객체와 내장 객체

- title 속성을 이용해서 HTML 문서의 제목 바꾸기

```
<script   language="javascript">
        var  title = window.prompt("제목을 입력하세요");
        document.title=title;
</script>
```

<TITLE>태크의 값을 변경한다.

- bgColor 속성으로 문서의 배경색 변경

```
<script   language="javascript">
        function changeBg(color) {
            document.bgColor=color;
        }
</script>
```

- HTML 문서내의 특정 태그를 찾아내기

메서드	설 명
getElementById	HTML태그의 id 특성값을 이용해 객체를 찾는다
getElementByName	HTML태그의 name 특성값을 이용해 객체를 찾는다
getElementByTagName	HTML태그 이름으로 객체를 찾는다.

Document 객체와 내장 객체

- createElement 와 appendChild 메서드로 HTML 태그를 동적으로 생성하기

```
<html>
<head>
<title> 동적으로 테이블 만들기</title>
<script   language="javascript">
        function  makeFriends() {
          var myFriends = window.prompt("친구가 몇 명인가요? ","");
          for ( var i=0; i<parseInt(myFriends); i++) {
            var textBox=document.createElement("input");
            var newLine=document.createElement("br");
            textBox.type="text";
            document.body.appendChild(textBox);
            document.body.appendChild(newLine);
          }
        }
</script>
</head>
<body onload="makeFriends()">
    <h2>내친구 목록</h2>
</body>
</html>
```

<input type="text" name="textBox">

<body>에 붙여 넣는다.

history 객체

- window객체에서 파생된 것으로 window를 생략한다.
  방문자가 최근에 방문한 URL을 웹브라우저가 저장해둔 히스토리정보를 이용해서 앞으로,뒤로 갈 수 있다. URL자체를 알아내지는 못한다. 어떤 일처리를 하고 나서 원래의 페이지로 보낼 때 이용한다.

```
<script language="javascript">
    alert("로그인 실패")
    history.back()     // history.go(-1) 과 동일
</script>
```

메서드	설 명
back()	웹브라우저의 [뒤로]버튼을 클릭한 것과 같이 이전 페이지로 이동
forward()	웹브라우저의 [앞으로]버튼을 클릭한 것과 같이 앞 페이지로 이동
go()	지정된 숫자만큼 이전페이지 또는 앞 페이지로 이동

location 객체

- window객체에서 파생된 것으로 window를 생략한다. 열려있는 문서의 URL정보를 얻을 수 있고, 이보다는 앞으로 이동할 문서의 URL을 설정하는 데 아주 중요하게 사용된다.
- 열릴 문서의 URL설정하기

```
<script language="javascript">
        location.href= "http://www.sun.com"
</script>


<script language="javascript">
      alert("회원가입을 하시겠습니까?")
      location.href= "member.html"
</script>


<script language="javascript">   // 연 창의 URL 이동
   opener.location.href= "member.html"
</script>


<script language="javascript">   // 부모창 지정프레임의 URL 이동
     parent .frame1.location.href= "member.html"
</script>
```

location 객체(2)

```
<script language="javascript">
  // 전체창에 새 페이지 열기
  top.location.href= "index.html“
</script>
```

form 객체

- document 객체 소속으로 html의 <form> 태그에 접근하는 정보를 제공한다. 웹프로그래밍에서 아주 중요한 전송과 관련하여 필수로 사용되는 객체이다.

```
<form name="form1" action="ok.jsp" method="post" target="_self" enctype="" >
  <input type="text" name="tel">
</form>
<form name="form2" action="ok.jsp" method="post" enctype="">
  <input type="text" name="tel">
</form>
```

- 위의 폼태그에서 "tel" 에 접근하려면 아래처럼 배열 또는 이름을 이용할 수 있다.
  document.form1.tel 또는 document.forms[0].elements[0]
  document.form2.tel 또는 document.forms[1].elements[0]

  * 배열정보를 이용할 때는 for 반복문을 통해 모든 폼/모든 요소에 접근할 때 유리하다

1. form 객체 속성

속성	설명
name	<form> 태그의 name속성 값 name 은 자바스크립트에서 폼이름으로 접근하기 이해 설정이 필요
action	<form> 태그의 action속성 값 전송에 사용하고자 할 경우 이 전송을 받아 처리할 페이지를 설정
method	<form> 태그의 method속성 값 전송의 방법으로 post 또는 get 방식을 사용할 수 있다. 특별히 지정하지 않으면 get방식이 사용된다.
target	<form> 태그의 target속성 값 _blank, _self, _top,_parent
encoding	<form> 태그의 enctype속성 값 전송 데이타를 encode하는 방법으로 첨부파일이 있다면 multipart/form-data 을, 첨부가 없다면 생략하면 된다
elements	<form> 태그내의 폼 요소를 배열로 저장

1. Form 객체의 객체

form 객체의 elements는 현재 form 안에 선언되어 있는 모든 입력 양식에 대한 객체 정보를 배열 형태로 포함하고 있다.

ex) document.form1.elements[0].value

2. Form 객체의 메서드

- submit() : 서버로 데이타 전송submit 버튼을 누른 효과
- reset() : 폼요소의 값을 모두 초기화reset 버튼을 누른 효과

3. Form객체의 이벤트 핸들러

- onsubmit : 입력양식에 있는 데이터를 서버로 보낼때 발생하는 submit 이벤트를 처리하기 위한 이벤트 핸들러
- onreset : 입력양식에 있는 리셋버튼을 눌렸을때 발생하는 reset이벤트를 처리하기 위한 이벤트 핸들러이다.

ex)

<form name="form1" action="nothing" onsubmit="return bewforeSubmit()">

HTML 문서에서 발생하는 이벤트 종류

이벤트	이벤트 핸들러	설명
blur	onblur	태그 영역에서 포커스가 없어질 때
change	onchange	텍스트 상자에 입력하거나 드롭다운 리스트에서 목록을 선택할 때
click	onclick	태그 영역을 클릭했을 때
dbclick	ondbclick	태그 영역을 더블클릭 했을때
dragdrop	ondragdrop	태그 영역을 드래그 했을 때
focus	onfocus	태그 영역이 포커스를 얻었을 때
keydown	onkeydown	태그 영역이 포커스를 얻은 후 사용자가 키보드의 키를 누를 때
keyup	onkeyup	태그 영역이 포커스를 얻은 후 사용자가 키보드의 키를 눌렀다 땔 때
keypress	onkeypress	키보드를 눌렀다가 때고 난 뒤
load	onload	<body>나 HTML 태그가 웹브라우저에 의해 해석될 때
unload	onunload	<body>나 HTML 태그가 없어질 때
mousedown	onmousedown	마우스 버튼을 누를 때

HTML 문서에서 발생하는 이벤트 종류

이벤트	이벤트 핸들러	설명
mouseup	onmouseup	마우스 버튼을 눌렀다 뗄 때
mousemove	onmousemove	마우스가 움직일 때
mouseover	onmouseover	태그 영역 안으로 마우스 포인터가 들어오는 순간
mouseout	onmouseout	태그 영역 바깥으로 마우스 포인터가 나가는 순간
resize	onresize	태그 영역의 크기가 조절 될 때
submit	onsubmit	<input type="submit">버튼을 클릭할 때
reset	onreset	<input type="reset">버튼을 클릭할 때

이벤트를 처리하는 3가지 방법

- 이벤트 핸들러를 이용하는 방법

```
<img  onclick="이벤트 핸들러 함수">
```

- HTML 객체에 이벤트 핸들러를 지정하는 방법

```
<script  language="javascript">
    window.onload = function() {
        //스크립트 코드를 추가
    }
</script>
```

- 동적으로 이벤트 처리를 추가하는 방법

```
<input   id="button"  type="button"  value="눌러">
<script  language="javascript">
    var  objButton=document.getElementById("button");
    objButton.attachEvent("onclick",clickEventHandler);
    function clickEventHandler() {
        //버튼이 클릭되었을때 실행된 내용 정의
    }
</script>
```

감사합니다.