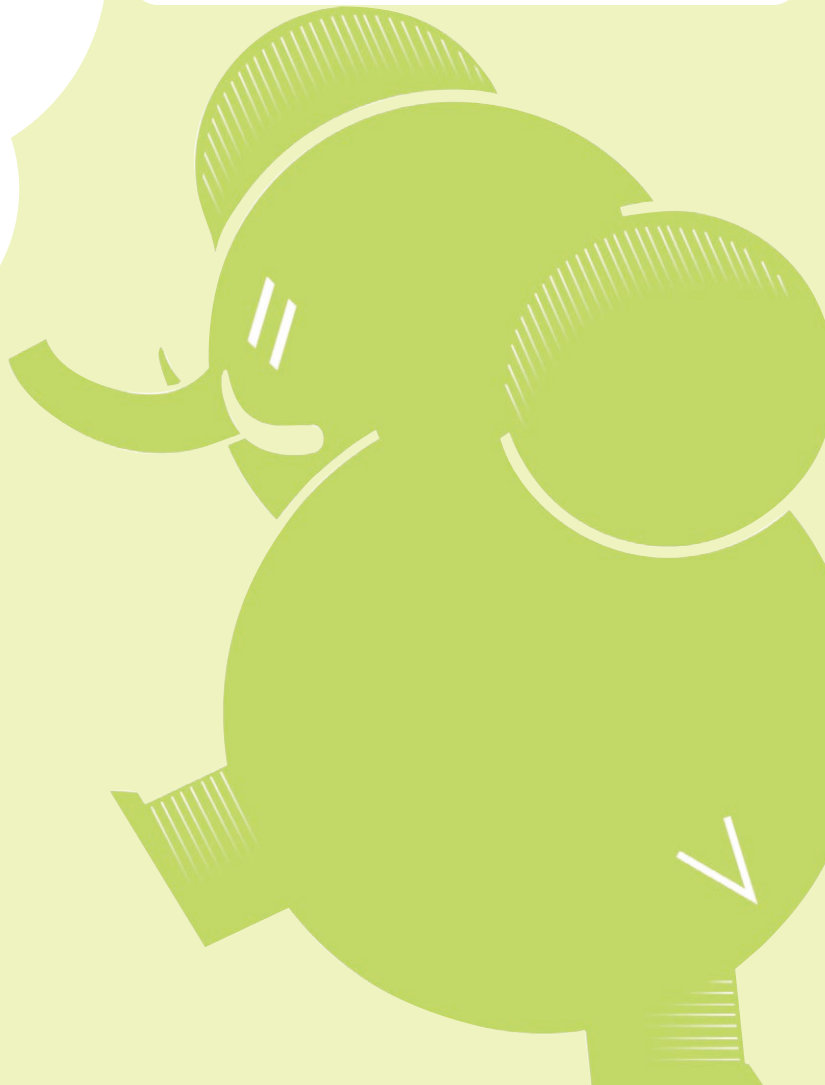


연습문제 풀이

[연습문제 풀이]

1. 움직이는 점선테이블 만들기
2. 테두리선 일부만 보이게 하기
3. 회원 가입폼 만들기
4. 글자에 그림자 효과 주기
5. 이미지에 외곽 테두리 만들기
6. 익스플로러 로고 화면 만들기
7. 문서의 제목 만들기
8. 태그를 이용한 폰트 디자인
9. 구구단 출력 프로그램 만들기
10. 1부터 500까지의 수 중에서 3의 배수를 제외한 수의 합 구하기
11. 마우스를 따라다니는 이미지 만들기
12. 선택한 이미지를 배경이미지로 바꾸기
13. 연도를 입력하면 해당 연도의 띠를 표시하기
14. 특정 경로 설정하기
15. 포커스 이동하기
16. 버튼에 현재 시간 출력하기
17. 링크에 마우스 올리면 사운드가 나오게 만들기
18. 접속할 때마다 배경 음악이 바뀌게 만들기
19. 로딩하는데 걸린 시간 표시하기



TEST

HTML & CSS 연습문제

01 움직이는 점선 테두리 만들기

(책 p. 237)



부록CD/별책부록/HTML&CSS/ani_dot.html



형식은 간단합니다. 2개의 테이블을 사용하여 전체 바깥테이블에 점선 애니메이션을 삽입하고, 그 내부에는 또 하나의 테이블을 삽입한 형식입니다. 내부의 테이블이 외곽 테이블을 덮어버려 테두리가 보이지 않는 것을 생각하여 내부 테두리의 cellspacing의 값을 1로 설정합니다. 또 cellspacing=10은 내부 콘텐츠와 테두리까지의 간격을 10픽셀 만큼 적용합니다.

Source

```
<HTML>
<HEAD>
  <TITLE>움직이는 점선 테두리 만들기</TITLE>
  <META http-equiv="Content-Type" content="text/HTML; charset=euc-kr">
</HEAD>

<BODY bgcolor="#FFFFFF" text="#000000">
  <TABLE width="250" border="0" cellspacing="0" cellpadding="0">
    <TR>
      <TD background="img/ani_black.gif">
        <TABLE width="250" border="0" cellspacing="1" cellpadding="10">
          <TR>
            <TD bgcolor="#FFFFFF">
              <FONT t size=2 color=RED><B><CENTER>움직이는 점선 테두리 만들기</CENTER></B></FONT><BR>
              <FONT size=2>1차적으로 움직이는 점선 테두리의<BR>
              원리를 이해 하시다<BR>
              그런 다음, 테이블의 구성 방식으로<BR>
              접근해 보시다.<BR></FONT>
            </TD>
          </TR>
        </TABLE>
      </TD>
    </TR>
  </TABLE>
</BODY>
</HTML>
```

점선 이미지

바깥 테이블에 점선 애니메이션이 적용됨

외곽 테이블을 보이게 하기 위한 설정

안쪽 테이블

02 테이블 테두리 일부만 보이게 하기

(책 p.237)



부록CD/별책부록/HTML&CSS/L_border.html

<BODY></BODY> 태그 사이에 다음 소스를 삽입합니다.

Source

```
<TABLE width="500" border="0" cellspacing="0" cellpadding="0" height="300">
<TR>
<TD bgcolor="FF9800">
<!--두 번째 테이블이 들어옵니다. //-->
<TABLE width="100%" border="0" cellspacing="1" cellpadding="5" height="300">
  <TR>
    <TD bgcolor="#FFFFCC">&nbsp;</TD>
    <TD bgcolor="#FFFFCC">&nbsp;</TD>
    <TD bgcolor="#FFFFCC">&nbsp;</TD>
  </TR>
  <TR>
    <TD bgcolor="#FFFFFF">&nbsp;</TD>
    <TD bgcolor="#FFFFFF">&nbsp;</TD>
    <TD bgcolor="#FFFFFF">&nbsp;</TD>
  </TR>
  <TR>
    <TD bgcolor="#FFFFFF">&nbsp;</TD>
    <TD bgcolor="#FFFFFF">&nbsp;</TD>
    <TD bgcolor="#FFFFFF">&nbsp;</TD>
  </TR>
  <TR>
    <TD bgcolor="#FFFFFF">&nbsp;</TD>
    <TD bgcolor="#FFFFFF">&nbsp;</TD>
    <TD bgcolor="#FFFFFF">&nbsp;</TD>
  </TR>
  <TR>
    <TD bgcolor="#FFFFFF">&nbsp;</TD>
    <TD bgcolor="#FFFFFF">&nbsp;</TD>
    <TD bgcolor="#FFFFFF">&nbsp;</TD>
  </TR>
  <TR>
    <TD bgcolor="#FFFFFF">&nbsp;</TD>
    <TD bgcolor="#FFFFFF">&nbsp;</TD>
    <TD bgcolor="#FFFFFF">&nbsp;</TD>
  </TR>
  <TR>
    <TD bgcolor="#FFFFFF">&nbsp;</TD>
    <TD bgcolor="#FFFFFF">&nbsp;</TD>
    <TD bgcolor="#FFFFFF">&nbsp;</TD>
  </TR>
  <TR>
    <TD bgcolor="#FFFFFF">&nbsp;</TD>
    <TD bgcolor="#FFFFFF">&nbsp;</TD>
    <TD bgcolor="#FFFFFF">&nbsp;</TD>
  </TR>
</TABLE>
</TD>
</TR>
</TABLE>
```

(책 p.281)



Source

<STYLE type="text/css">

<STYLE type="text/css">

< !-----

```
A:active { font-family: "굴림", "Arial", "Helvetica"; font-size: 9pt; color: #3366CC; text-decoration: none }
A:hover { font-family: "굴림", "Arial", "Helvetica"; font-size: 9pt; color: #CC6600; text-decoration: underline }
A:link { font-family: "굴림", "Arial", "Helvetica"; font-size: 9pt; color: #333333; text-decoration: none }
A:visited { font-family: "굴림", "Arial", "Helvetica"; font-size: 9pt; color: #CC6600; text-decoration: none }
.geston {font-size: 9pt; line-height: 130% ; color:#333333; font-family: "굴림", "Arial"}
.gestonblack {font-size: 9pt; line-height: 130% ; color:#000000}
-->
</STYLE>
```

<TABLE width="600" border="0" cellspacing="0" cellpadding="0" align="center">

<TR>

<TD align="center"> </TD>

<TR>

 |

<p> <TABLE width="100%" border="0" cellspacing="0" cellpadding="0"> </p>	<p> <TABLE border="1" style="width: 100%; border-collapse: collapse;"> </p>
--	---

| <TR bgcolor="#FFCC66"> | |
 |

	<TABLE width="100%" border="0" cellspacing="1" cellpadding="4">
--	---

| <TR bgcolor="#FFF9CE"> | |
 | | $\langle \text{TR} \rangle$

<TD bgcolor= "#FFFFFF" width= "20%" class= "gestonblack" align= "center">회원아이디	
--	--

</TD>

<TD bgcolor="#FFFFFF" class="gestonblack">

```
<INPUT type="text" name="textfield" size="8" maxlength="10" style="border:solid 1">
```


영문소문자/숫자조합3~8자 </TD>

</TR>

<TR>

 <TD bgcolor=“#FFFFFF” class=“gestonblack” align=“center”>비밀번호 |

<TD bgcolor="#FFFFFF" class="gestonblack">

```
<INPUT type="password" name="textfield2" size="8" maxlength="10" style="border:solid 1">
```

영문소문자/숫자조합3~8자 **<비밀번호확인>**

```
<INPUT type="password" name="textfield22" size="8" maxlength="8" style="border: solid 1">
```

</TD>

 $\langle \text{TR} \rangle$ <TD bgcolor="#FFFFFF" class="gestonblack" align="center">E-mail</TD> |

<TD bgcolor="#FFFFFF" class="gestonblack">

```
<INPUT type="text" name="textfield3" size="20" maxlength="50" style="border:solid 1">
```

여)at21@at21.com </TD>

스타일시트 적용부분

❶ 이어서

```

</TR>
<TR bgcolor="#FFF9CE">
<TD colspan="2" align="center"><IMG src="img/new002.gif" width="130" height="17"></TD>
</TR>
<TR>
<TD bgcolor="#FFFFFF" align="center" class="gestonblack">한글이름
</TD>
<TD bgcolor="#FFFFFF" class="gestonblack">
  <INPUT type="text" name="textfield4" size="15" maxlength="8" style="border:solid 1">
  성과 이름을 붙여주세요 (예)홍길동 </TD>
</TR>
<TR>
<TD height="18" bgcolor="#FFFFFF" align="center" class="gestonblack">주민등록번호
</TD>
<TD height="18" bgcolor="#FFFFFF" class="gestonblack">
  <INPUT type="text" name="textfield5" size="7" maxlength="6" style="border:solid 1"> -
  <INPUT type="text" name="textfield6" size="7" maxlength="7" style="border:solid 1">
  비밀번호 분실시 이용됩니다.</TD>
</TR>
<TR>
<TD bgcolor="#FFFFFF" align="center" class="gestonblack">실제생년월일
</TD>
<TD bgcolor="#FFFFFF" class="gestonblack">
  <INPUT type="radio" name="radiobutton" value="radiobutton"> 양력
  <INPUT type="radio" name="radiobutton" value="radiobutton"> 음력
  <INPUT type="text" name="textfield7" size="10" maxlength="10" style="border:solid 1"> 예)2000/05/22 </TD>
</TR>
<TR>
<TD bgcolor="#FFFFFF" align="center" class="gestonblack">전화번호
</TD>
<TD bgcolor="#FFFFFF" class="gestonblack">
  <INPUT type="text" name="textfield8" size="4" maxlength="4" style="border:solid 1"> -
  <INPUT type="text" name="textfield9" size="4" maxlength="4" style="border:solid 1"> -
  <INPUT type="text" name="textfield10" size="4" maxlength="4" style="border:solid 1"> 예)051-000-0000 </TD>
</TR>
<TR>
<TD bgcolor="#FFFFFF" align="center" class="gestonblack">휴대전화번호</TD>
<TD bgcolor="#FFFFFF" class="gestonblack">
  <INPUT type="text" name="textfield11" size="4" maxlength="4" style="border:solid 1"> -
  <INPUT type="text" name="textfield12" size="4" maxlength="4" style="border:solid 1"> -
  <INPUT type="text" name="textfield13" size="4" maxlength="4" style="border:solid 1"> 예) 011-000-0000 </TD>
</TR>
</TABLE></TD></TR></TABLE></TD></TR>
<TR>
<TD class="gestonblack" align="right" height="20">&nbsp;</TD></TR>
<TR> <TD class="gestonblack" align="center" height="20">
<TABLE width="22%" border="0" cellspacing="0" cellpadding="0" align="center">
<TR> <TD align="center"><IMG src="img/ok.gif" width="59" height="20"></TD>
<TD align="center"><IMG src="img/cancel.gif" width="59"></TR>
</TABLE></TD></TR></TABLE>

```

04 글자에 그림자 효과 주기

(책 p. 379)

부록CD/별책부록/HTML&CSS/text_shadow.html

설명

border의 position 효과를 이용한 것으로 작업 방법은 의외로 간단합니다.

2개의 <DIV></DIV> 태그를 사용하여, z-index 1과 2를 적용합니다. 하나는 오렌지색으로 앞쪽에 나타나게 설정하고, 나머지 하나는 진한 회색으로 색상을 맞추어 뒤쪽에 보이게 합니다. 효과는 position을 사용하여 2개가 3픽셀만큼 어긋나게 제작합니다.

Source

```
<HTML>
<HEAD>
  <TITLE>글자에 그림자 효과 주기</TITLE>
  <META http-equiv="Content-Type" content="text/HTML;charset=euc-kr">
</HEAD>
<BODY bgcolor="#FFFFFF" text="#000000">
  <DIV style="position:absolute; top:33; left:33; font-size:33pt; font-weight:bold; font-family:궁서체; color:#666666; z-index:1">글자에 그림자 효과 주기</DIV>
  <DIV style="position:absolute; top:30; left:30; font-size:33pt; font-weight:bold; font-family:궁서체; color:#FF9900; z-index:2">글자에 그림자 효과주기</DIV>
</BODY>
</HTML>
```

position을 이용하여 3픽셀만큼 차이를 둡니다

05 이미지에 외곽 테두리 만들기

(책 p. 341)

부록CD/별책부록/HTML&CSS/gift.html

설명

테두리 없는 이미지는 CSS를 활용하여 간단하게 테두리를 만들어 보았습니다.

Source

```
<IMG src="img/gift01.gif" width="50" height="50" style="border:solid 1px">
<IMG src="img/gift02.gif" width="50" height="50" style="border:solid 1px">
```

깔끔한 실선 테두리

위 형식은 border의 shorthand 형식을 간단하게 적용한 방법으로 shorthand를 이용하지 않으면 style="border-style:solid;border-width:1px"와 같은 형식으로 표현할 수 있습니다.

참고로 border는 border-style이 적용되지 않으면 width값을 적용할 수 없으므로 border-style 형식은 반드시 입력해야 합니다.

06 익스플로러 로고화면 만들기

(책 p. 379)



부록CD/별책부록/HTML&CSS/explor.html

설명

문제의 익스플로러 로고화면은 모두 `position` 속성을 사용하고, `z-index`는 별도로 사용하지 않습니다.

`z-index` 속성을 적용했을 경우 숫자가 높으면 높을수록 앞에 나타나며, `z-index` 속성을 적용하지 않았을 경우는 소스의 상단에 적히는 것보다 하단으로 내려갈수록 앞에 나타납니다.

여기서는 `z-index` 속성을 사용할 필요 없이 모두 내림차순으로 정리한 것이므로 제일 아래 소스가 가장 앞에 나타나게 됩니다.

이번 문제는 일종의 사고의 전환이라고 할 수 있습니다.

CSS를 공부하는 했으니 적용에 착각을 세우지 않으면 배운 것도 큰 의미가 없는 법입니다. 소스를 하나 하나 따져보면 똑같이 만들 수는 있겠지만 그것을 본인이 어떻게 제작해 가느냐가 중요하다고 생각합니다. 문제를 통해 소스를 분석한 다음 다른 부분에 적용할 방법은 없는지 고민하는 습관을 갖도록 합니다.

Source

```
<HTML>
<HEAD>
  <TITLE>Box Model</TITLE>
</HEAD>
<BODY bgcolor= "#FFFFFF">
  <DIV style= "position:absolute; left: 120px; top: 50px; border-color: #97dc74; width: 60px; height: 60px; border-width: 8px; border-style: solid"></DIV>
  <DIV style= "position:absolute; left: 66px; top: 86px; border-color: #ff3300; width: 70px; height: 70px; border-width: 8px;border-style: solid"></DIV>
  <DIV style= "position:absolute; left: 85px; top: 117px; border-color: #ffe05e; width: 100px; height: 100px; border-width: 10px;border-style: solid"></DIV>
  <DIV style= "position:absolute; left: 165px; top: 36px; width: 110px; height: 110px; border-width: 13px; border-color: #0099ff; border-style: solid"></DIV>
  <DIV style= "position:absolute; left: 141px; top: 162px; width: 500px ; height:50px; border: none; font-family: Arial Black, Helvetica, sans-serif; font-weight: bold; font-size: 28pt ;letter-spacing:-4"><STRONG>Internet Explorer</STRONG></DIV>
  <DIV style= "position:absolute; left: 117px; top: 117pt; width:100px ; height:40px; border: none; font-family: Arial , Helvetica, sans-serif; font-size: 15pt;letter-spacing:-1">
    <STRONG>Microsoft</STRONG></DIV>
</BODY>
</HTML>
```

07 문서의 제목 만들기

(책 p. 387)



부록CD/별책부록/HTML&CSS/title.html

설명

CSS 전반에 걸쳐 자주 등장하는 부분들이 이 곳에 많이 사용됩니다.

.title이라는 키워드를 지정하여 글자크기(font-size)는 18pt로 하며 굵게(font-weight) 나타나게 스타일시트를 설정합니다. 그리고 텍스트는 상자 내부에서 중앙정렬(text-align) 되도록 설정합니다. 전체 상자의 배경색(background-color)은 #FFFF99로 연한 노란색을 설정하고, 상자의 테두리(border)는 실선(solid)에 빨간색, 두께는 5pt로 설정합니다. 또한 상자 내부의 글자와 빨간 테두리 간의 간격을 두기 위해 padding 속성 값을 5pt로 설정합니다.

Source

```
<HTML>
<HEAD>
  <TITLE>문서의 제목 만들기</TITLE>
  <META http-equiv="Content-Type" content="text/HTML; charset=euc-kr">
  <STYLE type="text/css">
    <!--
    .title{font-size: 18pt ; font-weight:bold ; text-align : center ; width:400 ; border:solid red 5pt ; padding:5pt ;
    background-color:#FFFF99}
    -->
  </STYLE>
</HEAD>
<BODY bgcolor="#FFFFFF" text="#000000">
  <DIV class="title">문서의 제목 만들기</DIV>
</BODY>
</HTML>
```


08 태그를 이용한 폰트 디자인

(책 p. 387)



부록CD/별책부록/HTML&CSS/font-design.html

설명

태그를 이용해서 문자디자이너를 할 때는 inline Style Sheet와 position 속성, font 속성을 이용하여 제작합니다.

Macromedia라는 글자는 소스를 보시면 쉽게 이해하리라 봅니다. 그림, Dreamweaver4.0이라는 글자에 적용된 내용을 살펴보겠습니다.

우선 Dreamweaver4.0이라는 글자가 Macromedia 글자 위에 겹쳐져야 하므로, 위치를 선정해야 합니다. position 속성을 이용하여 글자의 위치를 설정합니다. 글자는 왼편에서 85px, 상단에서 27px 떨어진 곳에 위치시키면 기본 작업은 거의 다 마무리 한 셈입니다.

글자는 font 속성을 이용하여 꾸미시면 됩니다. 크기는 18pt, 폰트는 굵게 표현했습니다. 여기에 이탤릭체를 적용하고 싶으면 font-style:italic을 삽입하면 됩니다.

Source

```
<HTML>
<HEAD>
  <TITLE>태그를 이용한 문자 디자인</TITLE>
  <META http-equiv="Content-Type" content="text/HTML; charset=euc-kr">
</HEAD>

<BODY>
  <font color="#0000FF" style="font-size:18pt;font-weight:bold;">Macromedia</font>
  <DIV style="position:absolute;left:85px ; top:27px ; font-size:18pt;font-weight:bold;">Dreamweaver4.0</DIV>
</BODY>
</HTML>
```

TEST

JavaScript 연습문제

09 구구단 출력 프로그램 만들기

(책 p. 547)



부록CD/별책부록/JavaScript/ex01.html

가장 먼저 작성할 것은 `prompt()` 함수를 이용하여 문자열을 입력 받는 코드를 삽입해야 합니다. 변수 `dan`을 이용하여 문자열 입력을 받습니다.

Source

```
dan = prompt("1부터 9까지의 수를 입력하세요.", "");
```

그리고는 입력 받은 문자열(구구단을 출력하려면 수를 입력 받게 됩니다)을 가지고 구구단을 출력하는 소스를 작성해야 합니다. 구구단은 지정된 수를 1~9까지의 수로 곱하는 형식이니 `for`문을 이용하여 입력 받은 수와 1~9까지의 수를 곱하는 식을 출력합니다.

Source

```
for(i = 1; i < 10; i++) {
    document.write(dan + " * " + i + " = " + dan*i + "<BR>")
}
```

10 1부터 500까지의 수 중에서 3의 배수를 제외한 수의 합 구하기 (책 p. 547)



부록CD/별책부록/JavaScript/ex02.html

1부터 500까지의 합을 구하려면 `for`문을 이용하여 반복문을 만들고 수의 증가를 일정 변수에 누적합 시키면 됩니다. 그러나 여기서 3의 배수를 빼기 위해서는 반복문 안에 수의 증가가 3의 배수가 되면 누적합을 하지 말아야 합니다. 그렇기 때문에 `if`조건문을 사용하여 `if((i%3)!=0)`라고 지정하면 (단, `i`는 1부터 500까지의 증가식을 위한 변수) 1부터 500까지의 증가식에서 값이 3으로 나누어 떨어지지 않을 경우에는 `document.write`를 통해 수를 출력해주고 아울러 증가수를 누적합 시킵니다. 반면에 그렇지 않을 경우에는 아무 작업을 하지 않기 때문에 1부터 500까지의 수 중 3의 배수를 제외한 합을 구할 수 있습니다.

➡ 다음 페이지 계속

이어서


Source

```
<SCRIPT language="JavaScript">
<!--
    take = 0
    for ( i=1; i<=500; i++ ) { // 1부터 500까지의 반복을 위한 for 제어구조
        if((i%3)!=0) { // 3의 배수를 골라내기 위해 증가수가 3으로 떨어지는지 판단합니다.
            document.write(i)
            if(i != 500) document.write(" + ")
            take += i
        }
    }
    document.write(" = " + take)
//-->
</SCRIPT>
```

잠깐 큰 따옴표 내 출력부분에 공백부분이 들어갔다는 것에 주의하세요.

11 마우스 따라다니는 이미지 만들기

(책 p. 551)

 부록CD/별책부록/JavaScript/ex03.html

이미 본문에서 마우스를 따라다니는 문자열상자를 만들어 보았습니다. 이를 간단히 테이블을 제거하고 레이어 내에 이미지를 삽입하게 함으로 해서 간단히 제작할 수 있습니다.

<BODY> 태그 내에 다음과 같이 m_box라는 이름의 레이어를 제작합니다. 아울러 레이어 내에는 마우스를 따라다닐 이미지를 삽입합니다.

Source

```
<LAYER id="m_box" z-index="1">
    <IMG src="ex3_img.gif" border=0>
</LAYER>
```

<HEAD> 태그 내에는 document.m_box를 통해 문서 내의 레이어 객체를 선택하고 left값과 top값을 각각 마우스의 포인터 위치에서 15픽셀씩 떨어진 위치로 지정하며, window.OnMouseMove에 마우스가 이동될 때마다 레이어 객체를 선택하여 위치값을 지정하게 해주면 됩니다.

➡ 다음 페이지 계속

➡ 이어서

Source

```
<SCRIPT language="JavaScript">
<!--
function m_position(take) {
    document.m_box.left=take.pageX+15 // 이미지가 삽입된 레이어의 좌측좌표값을 마우스의 현위치에서 우로 15픽셀이동
    document.m_box.top=take.pageY+15 // 이미지가 삽입된 레이어의 상측좌표값을 마우스의 현위치에서 하로 15픽셀이동
}
window.captureEvents(Event.MOUSEMOVE)
window.onmousemove=m_position // 마우스가 움직일때마다 이미지가 삽입된 레이어의 좌표값 이동
//-->
</SCRIPT>
```

12 선택한 이미지를 배경이미지로 바꾸기

(책 p. 557)



부록CD/별책부록/JavaScript/ex04.html

2개의 이미지 파일을 각각의 변수에 대입하는 코드를 삽입합니다. 물론 이미지 수가 많다면 그만큼 선언문을 넣어주어야 하겠죠? 아울러 clear라는 변수에 비어 있는 이미지 객체를 선언하는 것은 소스를 실행할 때 마우스가 이미지에서 벗어나면 배경 이미지가 나타나지 않도록 하기 위해서입니다.

Source

```
clear = new Image() // clear에 비어있는 이미지객체를 선언합니다.

bg_img1 = "ex4_01.jpg" // bg_img1에 이미지 파일명 지정
bg_img2 = "ex4_02.jpg" // bg_img2에 이미지 파일명 지정
```

다음으로 BG_change() 라는 함수를 제작하는데 이곳에는 문서의 배경그림을 지정하여 해당 매개변수값 즉 이미지를 입력하게 됩니다. 그로 인한 배경이미지를 변경할 수가 있게 되는 것입니다.

Source

```
function BG_change(imgName) { // 매개변수로 받아온 이미지를 배경이미지로 지정하는 함수
    document.body.background = eval(imgName);
}
```

➡ 다음 페이지 계속

➡ 이어서

이렇게 자바스크립트 소스를 작성해 놓고 문서 내의 이미지삽입 부분에서 `onMouseover`와 `onMouseout` 이벤트 핸들러를 이용하여 `BG_change()` 함수를 호출합니다. 이때 `onMouseover`에는 이미지명을 `onMouseout`에는 `clear`를 매개변수로 사용하여 마우스 오버 시에는 배경이미지가 변하고 마우스 아웃 시에는 배경이미지가 없는 효과를 만들어 냅니다.

Source

```
<A onmouseover="BG_change('bg_img1');" onmouseout="BG_change('clear');" >IMG src="ex4_01.jpg" border="0"</A>
<A onmouseover="BG_change('bg_img2');" onmouseout="BG_change('clear');" >IMG src="ex4_02.jpg" border="0"</A>
```

13 연도를 입력하면 해당 연도의 띠를 표시하기

(책 p. 613)



부록CD/별책부록/JavaScript/ex05.html

<BODY> 태그 내에 연도를 입력할 수 있는 폼을 구성합니다. 아울러 버튼을 삽입하고 클릭했을 때는 띠를 처리하기 위한 `take_show()` 함수를 호출합니다.

Source

```
<FORM name="frm">
  <INPUT type="text" SIZE="4" NAME="inyear" value="1975">
  <INPUT type="button" VALUE="띠맞추기" onClick="take_show()"><p>
  <INPUT type="text" SIZE="9" NAME="birth">
</FORM>
```

`take_show()` 함수에는 띠를 구하기 위한 기본값을 입력합니다. 소스에는 2009년도를 입력하였고 이수는 12로 나누어 계산을 수행할 때 해당 년도의 띠와 맞아 떨어지는 수입니다. 그렇기 때문에 2009가 아니더라도 1997이나 2021을 입력해도 됩니다. 그리고 `birthyear` 변수에는 폼에서 넘겨받은 연도를 대입하고 `birthpet`에 기본값으로 “잘못된 연도”라는 문자열을 입력합니다. 다음으로 초기에 지정한 연도에서 입력 받은 연도를 빼고 12로 나눈 나머지를 가지고 12가지의 띠를 구하면 됩니다.

➡ 다음 페이지 계속

➡ 이어서

Source

```

<SCRIPT language="JavaScript">
<!--
function take_show () {
    var toyear = 2009 // 띠를 구하기 위한 수입력(12배수)
    var birthyear = document.frm.inyear.value
    var birthpet = "잘못된 연도입니다."
    x = (toyear - birthyear) % 12
    if ((x == 1) || (x == -11)) {birthpet="쥐띠"}
    else {
        if (x == 0) {
            birthpet="잘못된 년도입니다."}
        else {
            if ((x == 11) || (x == -1)) {birthpet="호랑이띠"}
            else {
                if ((x == 10) || (x == -2)) {birthpet="토끼띠"}
                else {
                    if ((x == 9) || (x == -3)) {birthpet="용띠"}
                    else {
                        if ((x == 8) || (x == -4)) {birthpet="뱀띠"}
                        else {
                            if ((x == 7) || (x == -5)) {birthpet="말띠"}
                            else {
                                if ((x == 6) || (x == -6)) {birthpet="양띠"}
                                else {
                                    if ((x == 5) || (x == -7)) {birthpet="원숭이띠"}
                                    else {
                                        if ((x == 4) || (x == -8)) {birthpet="닭띠"}
                                        else {
                                            if ((x == 3) || (x == -9)) {birthpet="개띠"}
                                            else {
                                                if ((x == 2) || (x == -10)) {birthpet="돼지띠"}
                                                }}}}}}}}}}}
                        document.frm.birth.value = birthpet
                    }
                }
            }
        }
    }
}
//-->
</SCRIPT>

```

14 특정 경로 설정하기

(책 p. 613)



부록CD/별책부록/JavaScript/ex06.html

본 소스는 너무나도 간단하게 제작할 수 있습니다. 보기에는 정말 특이한 현상 같지만 기본적으로 윈도우시스템에서 익스플로러와 윈도우 탐색기는 서로 연관되어 있기에 윈도우 탐색기에서도 URL을 입력하여 웹서핑이 가능합니다. 또한 URL을 `http://`가 아닌 `file://`이라고 지정하면 컴퓨터의 하드 드라이브를 지정할 수 있습니다. 그렇기 때문에 `file:///C:/`라고 지정한다면 컴퓨터의 C드라이브를 웹브라우저에서도 볼 수 있는 것입니다.

즉 소스에서 링크를 클릭하게 되면 `window.location`을 이용하여 URL을 하드 드라이브로 지정해주면 됩니다.

Source

```
<HTML>
<HEAD>
  <TITLE>링크를 클릭하면 하드드라이브가 보이는 소스</TITLE>
  <META http-equiv="Content-Type" content="text/html; charset=euc-kr">
  <SCRIPT language="JavaScript">
    <!--
      function Sho
      wdrive()
      {
        window.location= 'file:///C:/'
      }
    //-->
  </SCRIPT>
</HEAD>
<BODY bgcolor="#FFFFFF" text="#000000">
  <CENTER>
    <A href="#" onClick="Showdrive()">클릭하시면 드라이브가 보입니다.</A>
  </CENTER>
</BODY>
</HTML>
```

15 포커스 이동하기

(책 p. 645)



부록CD/별책부록/JavaScript/ex10.html

가장 먼저 <HEAD> 태그 내에 다음과 같이 함수를 만들어 줍니다. 함수의 내용은 매개변수로 받은 폼객체를 선택하게 합니다.

Source

```
<SCRIPT language="JavaScript">
<!--

function selectField(frmName, fldName)
{
    document.forms[frmName].elements[fldName].select();
}
//-->
</SCRIPT>
```

다음으로 문서내의 <BODY> 태그 안에는 폼을 지정하고 두 개의 입력상자를 만들어 줍니다. 각각의 name값은 다르게 필히 지정해주셔야 앞서 제작한 함수에서 선택할 수 있습니다. 그리고 두 개의 버튼을 만드는데 onClick 시 폼명과 입력상자의 name값을 전달하여 함수를 실행하게 합니다.

Source

```
<FORM name="form1">
  <INPUT type="text" name="strText" size=30><BR>
  <INPUT type="text" name="strText2" size=30><BR>
  <INPUT type="button" name="butDoit" value="첫번째 폼에 포커스주기" onClick="selectField('form1', 'strText')"><BR>
  <INPUT type="button" name="butDoit2" value="두번째 폼에 포커스주기" onClick="selectField('form1', 'strText2')">
</FORM>
```

이렇게 하면 각 버튼을 클릭했을 때 해당 매개변수로 전달하는 입력상자를 선택할 수 있게 되어 포커스가 이동합니다. 또한 현 소스 자체만으로는 굳이 폼명을 전달시킬 필요는 없지만 폼을 여러 개 사용하여 제작한다면 반드시 매개변수에 폼명도 삽입 해주어야 합니다.

16 버튼에 현재 시간 출력하기

(책 p. 645)

 부록CD/별책부록/JavaScript/ex07.html

버튼에 시간을 출력하기 위해 <BODY> 태그 내에 다음과 같이 폼을 이용하여 버튼을 삽입합니다. 본 버튼을 제어하기 위해 반드시 폼과 버튼에는 name 값을 부여합니다.

Source

```
<FORM name=form1>
  <INPUT name="bu_time" type="button">
</FORM>
```

<HEAD> 태그 내에는 시간을 출력하기 위해 시간을 구하는 tt() 함수를 제작합니다. today라는 변수에 현재의 Date() 값을 대입하며 폼버튼에 today의 시간을 선택하여 입력합니다. 단 여기서 버튼에 대입할 때 버튼의 value에 대입해야 출력이 가능합니다. 그런 다음 setTimeout() 메소드를 이용하여 1초에 한번씩 실행하게 해줍니다. 즉 1초에 한번씩 실시간대를 맞춰줘서 시간이 지속되는 효과를 연출합니다.

Source

```
<SCRIPT language="JavaScript">
<!--
function tt() {
    today = new Date()
    document.form1.bu_time.value = today.getHours() + "시 " + today.getMinutes() + "분 " + today.getSeconds() + "초"
    setTimeout( 'tt()', 1000)
}
//-->
</SCRIPT>
```

마지막으로 페이지를 로드할 때 함수를 호출할 수 있도록 onLoad 이벤트를 핸들러를 사용하여 <BODY> 태그에 삽입합니다.

Source

```
<BODY bgcolor="#FFFFFF" text="#000000" onLoad="tt()">
```

17 링크에 마우스를 올리면 사운드가 나오게 만들기

(책 p. 682)



부록CD/별책부록/JavaScript/ex08.html

링크가 걸려있는 이미지나 텍스트에 마우스를 올려놓았을 때 사운드가 출력되게 하려면 **OnMouseOver** 이벤트 핸들러를 사용하여 **eval()** 함수에 사운드 파일을 지정해주면 됩니다.

<HEAD> 태그 내에 다음과 같이 사운드 파일을 매개변수로 받아 재생을 해주는 코드를 삽입합니다.

Source

```
<SCRIPT language="JavaScript">
<!--
    function over_sound(str){
        eval(str);
    }
//-->
</SCRIPT>
```

그리고 문서에 배경음악 삽입코드를 입력하되 사운드파일명을 지정하지않고 1번만 반복하게 해줍니다.

Source

```
<bgsound id="music" loop="1">
```

그리고 문서에 삽입될 링크에 **OnMouseOver** 이벤트 핸들러를 사용하여 음악파일을 매개변수로 전달시키면 됩니다.

Source

```
<A href="#" OnMouseOver="over_sound( 'music.src=\' ex11_001.wav\'' );">마우스를 올려보세요</A>
```

18 접속할 때마다 배경음악이 바뀌게 만들기

(책 p. 683)



부록CD/별책부록/JavaScript/ex09.html

페이지에 접속할 때마다 배경음악이 바뀌게 하려면 `random()` 함수를 사용하여 난수를 발생시키고 그 값을 가지고 재생할 배경음악을 지정하는 원리를 사용해야 합니다.

제일 먼저 배경음악으로 재생할 각 사운드 파일명을 변수에 할당합니다. 그리고는 변수 `x`에 사운드 파일수 만큼의 난수를 생성시키고 발생한 난수에 따라 변수 `x`에 해당 사운드 파일명을 담은 변수를 대입합니다. 그리고 나서 마지막으로 익스플로러와 넷스케이프에 맞는 사운드 재생 코드를 입력하면 됩니다.

Source

```
<SCRIPT language="JavaScript">
<!--
    var sound1="ex12_001.wma"
    var sound2="ex12_002.wma"
    var sound3="ex12_003.wma"
    var sound4="ex12_004.wma"
    var sound5="ex12_005.wma"
    var x=Math.round(Math.random()*4)
    if (x==0) x=sound1
    else if (x==1) x=sound2
    else if (x==2) x=sound3
    else if (x==3) x=sound4
    else x=sound5
    if (navigator.appName=="Microsoft Internet Explorer")
        document.write( '<bgsound src= ' + " " + x+ " " + ' loop="infinite">' )
    else
        document.write( '<embed src= ' + " " + x+ " " + ' hidden="true" border="0" width="20" height="20" autostart="true" loop="true">' )
    //-->
</SCRIPT>
```

19 로딩하는데 걸린 시간 표시하기

(책 p. 683)



부록CD/별책부록/JavaScript/ex11.html

페이지를 모두 로딩하는데 걸리는 시간을 계산하기 위해서는 쉽게 **onLoad** 이벤트 핸들러를 사용합니다. 단순히 페이지를 로딩했을 때 이벤트를 발생하게 하는 원리를 갖고 있지만 조금만 응용 하면 소스를 쉽게 풀어갈 수 있습니다.

잠깐 <HEAD> 태그에 삽입된 코드는 페이지 접속 시 가장 먼저 실행하게 되어 있습니다. 그런 이유에서 접속했을 때의 시간을 변수에 할당해주고 **onLoad** 이벤트 핸들러를 사용하여 페이지 로딩이 완료되었을 때 함수를 하나 만들고 그곳에 현 시간을 다른 변수에 할당해줍니다. 그 변수에서 초기 접속시의 시간을 빼주게 되면 페이지를 로드하는데 걸린 총 시간을 구할 수 있습니다.

소스를 작성해 보면 제일 먼저 변수 **Start**에 현 시간을 입력합니다. 즉 페이지를 열었을 때의 시간을 의미합니다. 그리고는 **load_time()**이라는 함수명을 만들고 다시 현 시간 즉 페이지의 로딩을 완료했을 때의 시간을 대입합니다. 여기서 잘못 생각할 수 있는 것이 이렇게 되면 변수 **Start**나 **Stop**의 값이 같은 것이 아닌가 할 수도 있겠지만 함수내의 코드는 외부에서 해당 함수를 호출했을 때만 실행을 하게 됩니다. 즉 **load_time()** 함수는 <BODY> 태그 내에 **onLoad** 이벤트 핸들러를 사용하여 페이지 로딩을 완료했을 때만 실행 하게 됩니다. 그렇기 때문에 변수 **Start**와 **Stop**은 서로 다른 값을 갖게 되며 변수 **Stop**은 시간이 경과된 후의 값을 갖게 됩니다. 다음으로 **l_sec** 변수에 **Start**와 **Stop**의 값을 1970년 1월 1일 이후의 시간을 1000분의 1초단위로 나타내는 **getTime()** 메소드를 사용하여 그 차를 대입하고 그 값을 1000으로 나누어 소수 둘째 자리까지 나타내는 초단위로 변환합니다. 이렇게 소스를 작성하면 페이지의 총 로드된 시간을 구할 수 있습니다.

Source

```
<SCRIPT language="JavaScript">
<!--
    Start = new Date() // 페이지를 첫 접속할시의 시간을 Start에 저장합니다.

    function load_time() {
        Stop = new Date() // 페이지 로드가 완료되었을 시의 시간을 Stop에 저장합니다.
        l_sec = Stop.getTime() - Start.getTime();
        l_sec = l_sec / 1000
        alert("이 페이지를 로딩하는데 걸린 시간은 \n\n" + l_sec + " 초 입니다.");
    }
    //-->
</SCRIPT>
```