

2

[JavaScript]

1. JavaScript
2. 자바스크립트 사용법
3. 변수(Variables)
4. 연산자
5. 제어문
6. 팝업 박스(Popup Boxes)
7. 함수와 이벤트
8. 예외 처리
9. 특수문자(Special Text)
10. 자바스크립트 객체

1. JavaScript

자바스크립트는 웹에서 사용되는 스크립팅 언어입니다. 자바스크립트는 디자인을 개선하거나, 쿠키를 생성하거나, 웹 페이지에 폼을 검증한다거나, 브라우저 종류를 파악하는 등의 기능을 추가하기 위해 많은 사람들이 사용하고 있습니다. 자바스크립트는 익스플로러, 파이어폭스, 오페라 등의 브라우저에서 잘 동작하며 가장 많이 사용되는 스크립팅 언어입니다.

1.1 자바스크립트의 특징?

- 자바스크립트는 HTML 페이지에 대화기능을 추가하기 위해 고안되었습니다.
- 자바스크립트는 스크립팅 언어⁸⁾이며, 인터프리터 언어입니다.
- 스크립팅 언어는 경량(lightweight) 프로그래밍 언어입니다.
- 자바스크립트는 일반적으로 HTML 문서에 바로 포함되어집니다.
- 누구나 저작권 없이 자바스크립트를 사용할 수 있습니다.

1.2 자바스크립트로 할 수 있는 일은?

- 자바스크립트는 HTML 디자이너에게 프로그래밍 툴을 제공합니다. HTML 제작자는 보통 프로그래머가 아닙니다. 그러나 자바스크립트는 매우 간단한 문법을 가지고 있는 스크립팅 언어입니다. 대부분 누구든지 자신의 HTML 페이지에 코드 일부분을 복사하여 사용할 수 있습니다.
- 자바스크립트는 HTML 문서에 동적 텍스트를 줄 수 있습니다. `document.write("<h1>" + name + "</h1>")` 같은 자바스크립트 문장은 HTML 페이지에 변수의 값을 넣을 수 있도록 해줍니다.
- 자바스크립트는 이벤트를 줄 수 있습니다. 자바스크립트는 페이지 로딩이 끝나거나, 사용자가 HTML 요소를 클릭하는 등 어떤 동작이 발생했을 때 실행되도록 할 수 있습니다.
- 자바스크립트는 HTML 요소를 읽고 쓸 수 있습니다. 자바스크립트는 HTML 요소의 내용을 읽고 변경시킬 수 있습니다.
- 자바스크립트는 데이터의 유효성을 검증할 수 있습니다. 자바스크립트는 서버에 데이터를 전송하기 전에 폼의 데이터 유효성을 검증할 수 있습니다.
- 자바스크립트는 방문자의 브라우저를 알아낼 수 있습니다. 자바스크립트는 방문자의 브라우저를 알아내고, 그 브라우저 종류에 따라 다른 페이지를 나타낼 수 있습니다.
- 자바스크립트는 쿠키를 생성하여 방문자의 컴퓨터에 저장하고 이 정보를 다시 검색하여 사용할 수 있습니다.

8) 언어란 컴파일 없이 소스코드를 바로 실행키는 언어입니다.

2. 사용법

2.1 script 태그

자바스크립트를 HTML 페이지에 넣기 위해서는 `<script>` 태그를 이용합니다. `<script>` 태그는 스크립팅 언어를 정의하는 `type` 속성을 가지고 있습니다. 그래서 자바스크립트의 시작은 `<script type="text/javascript">`로 시작하고 끝은 `</script>`로 끝납니다.

```
<html>
<body>
<script type="text/javascript">
...
</script>
</body>
</html>
```

2.2 자바스크립트 주석

다음은 자바스크립트를 이용하여 HTML 페이지에 문자열을 나타내는 예입니다. 소스코드에서 `<!--` 와 `//-->`는 자바스크립트를 지원하지 않는 브라우저에서는 HTML 주석으로 간주하여 무시하라는 의미입니다. 끝나는 주석의 `//`는 자바스크립트 주석입니다. 끝부분에 이것을 넣는 이유는 `-->` 부분을 자바스크립트 코드로 인식하지 않게 하기 위해서입니다. 자바스크립트에서 여러 줄 주석을 넣기 위해서는 `/*` 으로 시작하고 `*/` 로 끝내면 됩니다.

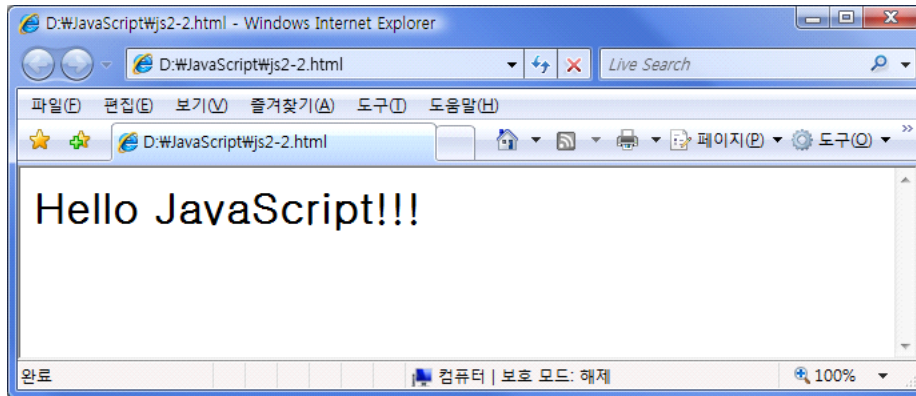
● 예제 코드

```
1: <!-- js2-2.html -->
2:
3: <html>
4: <body>
5:
6: <script type="text/javascript">
7: <!--
8: document.write("<h1>Hello JavaScript!!!</h1>");
9: //-->
10: </script>
11:
12: </body>
13: </html>
```

자바스크립트는 `<body>`와 `<head>`에 넣을 수 있습니다. `<body>` 부분에 넣었을 경

우에는 페이지가 로드될 때 실행되고, <head>부분에 넣었을 경우에는 호출되었을 때 실행됩니다. 자바스크립트를 페이지에 넣었을 경우, HTML 페이지가 브라우저에 로드될 때 바로 실행되길 원할 경우도 있지만 사용자의 이벤트에 의해 실행되게 하고 싶은 경우도 있습니다.

● 실행 결과



2.3 Break up 코드 라인

자바스크립트의 문장은 ";"로 끝나야 합니다. 그러나 문장이 길어질 경우 두 줄에 걸쳐 코드를 입력하고 실행 시에는 한 줄로 실행되게 하기 위해서는 문장의 맨 뒤에 "\"(backslash)를 넣어주면 됩니다. 다음은 정상적으로 실행됩니다.

```
document.write("Hello \
World!");
```

그러나 다음처럼 사용하는 것은 잘못된 예입니다.

```
document.write \
("Hello World!");
```

2.4 자바스크립트 위치

자바스크립트를 HTML 문서에 포함할 경우에는 다음 3가지 경우가 있습니다.

- <head> 부분 : 이 경우에는 이벤트가 발생했을 경우 스크립트가 호출되어 실행됩니다. <head>부분에 스크립트를 넣었을 경우, 스크립트를 사용하기 전에 로드되어야 합니다.

```
<html>
<head>
```

```
<script type="text/javascript">
...
</script>
</head>
```

- <body> 부분 : 이 경우에는 페이지가 로드되고 <body>안의 자바스크립트가 정의되어 있는 곳에서 자바스크립트가 실행됩니다.

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
...
</script>
</body>
```

- <head>와 <body> 부분 : 문서 내에서 자바스크립트의 수는 제한이 없습니다. 그러므로 자바스크립트는 <head>부분에 올 수 있으며, <body> 부분에도 올 수 있습니다.

```
<html>
<head>
<script type="text/javascript">
...
</script>
</head>
<body>
<script type="text/javascript">
...
</script>
</body>
```

2.5 외부 자바스크립트 사용

같은 자바스크립트를 여러 페이지에서 사용할 때가 있습니다. 이럴 경우 자바스크립트를 외부 파일로 만들어 놓을 수 있습니다. 이때 자바스크립트 파일의 확장자는 js입니다.

그리고, 자바스크립트를 외부 파일로 만들어 놓을 때에는 <script> 태그를 넣으면 안 된다는 것을 주의하세요.

그리고 HTML 페이지에서 외부의 파일(xxx.js)을 사용하기 위해서는 <script>태그의 src 속성을 이용하여 자바스크립트 파일 명을 입력하면 됩니다.

```
<html>
```

```
<head>
<script type="text/javascript" src="xxx.js"></script>
</head>
<body>
</body>
</html>
```

3. (Variables)

자바스크립트에서 변수는 값을 저장하거나, 표현하는데 사용합니다. 다른 언어와는 다르게 자바스크립트에서는 변수 선언 시 데이터형을 지정할 필요가 없습니다. 변수 선언 시 var 키워드를 이용하여 선언할 수 있습니다. 그러나 var를 빼도 같은 의미가 됩니다.

```
var x;  
var name = "Mr. Kim";  
y = 10;
```

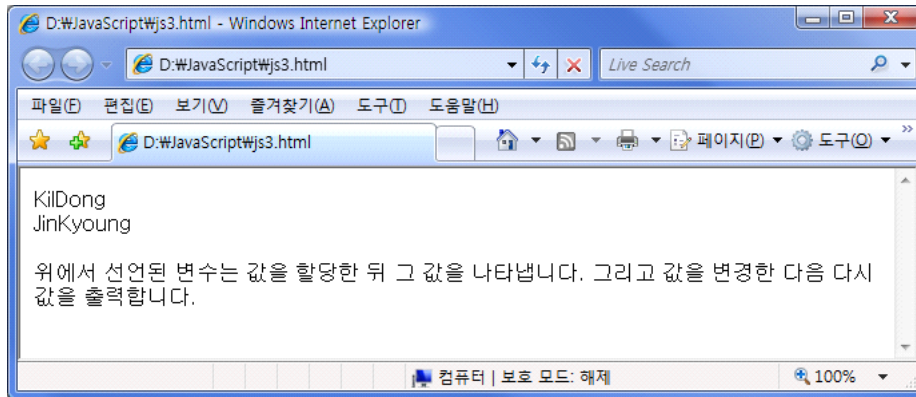
자바스크립트 변수 이름은 다음 두 가지 규칙을 가지고 있습니다.

- 변수 이름은 대/소문자를 구분합니다. 즉 변수 y와 Y는 다른 변수입니다.
- 변수 이름은 문자 또는 "_" (underscore character)로 시작해야 합니다.

● 예제 코드

```
1: <!-- js3.html -->  
2:  
3: <html>  
4: <body>  
5:  
6: <script type="text/javascript">  
7: var name;  
8: name="KilDong";  
9: document.write(name);  
10: document.write("<br />");  
11: name="JinKyoung";  
12: document.write(name);  
13: </script>  
14:  
15: <p> 선언된 변수는 값을 할당한 뒤 그 값을 나타냅니다. 그리고 값을 변경한 다음 다시 값을  
출력합니다.</p>  
16:  
17: </body>  
18: </html>
```

- 실행 결과



4.

4.1 산술 연산자(Arithmetic operators)

산술 연산자는 변수 또는 값을 사이 산술 연산을 수행합니다. 다음은 $y=5$ 일 경우 각 산술 연산자들에 대한 설명입니다.

연산자	설명	예	결과
+	Addition	$x=y+2$	$x=7$
-	Subtraction	$x=y-2$	$x=3$
*	Multiplication	$x=y*2$	$x=10$
/	Division	$x=y/2$	$x=2.5$
%	Modulus(나머지)	$x=y\%2$	$x=1$
++	Increment	$x=++y$	$x=6$
--	Decrement	$x=--y$	$x=4$

4.2 할당 연산자(Assignment operators)

할당 연산자는 값을 변수에 저장하기 위해 사용합니다. 다음은 $x=10$, $y=5$ 일 경우 각 할당 연산자들에 대한 설명입니다.

연산자	예	같은 의미	결과
=	$x=y$		$x=5$
+=	$x+=y$	$x=x+y$	$x=15$
-=	$x-=y$	$x=x-y$	$x=5$
=	$x=y$	$x=x*y$	$x=50$
/=	$x/=y$	$x=x/y$	$x=2$
%=	$x\%=y$	$x=x\%y$	$x=0$

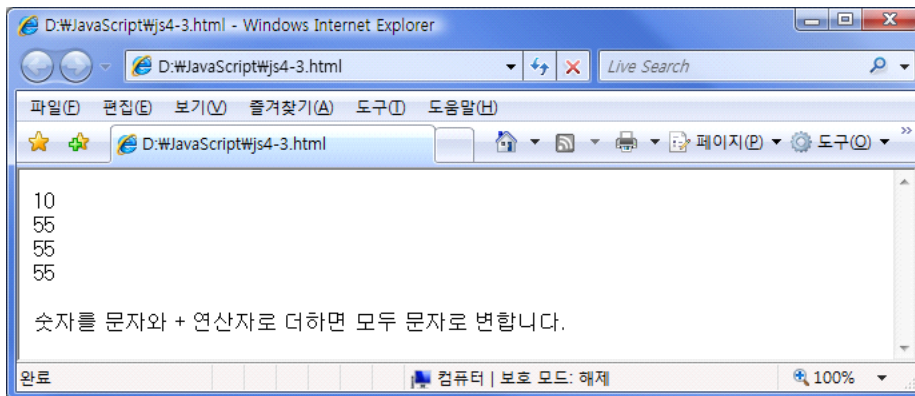
4.3 + 연산자

+ 연산자는 숫자를 더해주기 위해 사용될 수도 있지만 문자열의 연결해서 나타낼 때에도 사용합니다.

- 예제 코드

```
1: <!-- js4-3.html -->
2:
3: <html>
4: <body>
5:
6: <script type="text/javascript">
7:   x=5+5;
8:   document.write(x);
9:   document.write("<br />");
10:  x="5"+"5";
11:  document.write(x);
12:  document.write("<br />");
13:  x=5+"5";
14:  document.write(x);
15:  document.write("<br />");
16:  x="5"+5;
17:  document.write(x);
18:  document.write("<br />");
19: </script>
20:
21: <p>    문자와 + 연산자로 더하면 모두 문자로 변환됩니다.</p>
22:
23: </body>
24: </html>
```

- 실행 결과



4.4 비교 연산자(Comparison Operators)

비교 연산자는 변수들 또는 값들 사이의 동등성을 결정하는데 사용하는 연산자입니다. 연산 결과는 true 또는 false로 나타냅니다. 다음은 x=5일 경우 각 비교연산자들의 예입니다.

연산자	설명	예
==	is equal to	x==8 is false
===	is exactly equal to value and type	x===5 is true, x==="5" is false
!=	is not equal	x!=8 is true
>	is greater than	x>8 is false
<	is less than	x<8 is true
>=	is greater than or equal to	x>=8 is false
<=	is less than or equal to	x<=8 is true

4.5 논리 연산자(Logical Operators)

논리 연산자는 변수들 또는 값들 사이의 논리 연산을 수행합니다. 다음은 x=6, y=3일 경우 각 논리연산자들의 예입니다.

연산자	설명	예
&&	and	(x < 10 && y > 1) is true
	or	(x==5 y==5) is false
!	not	!(x==y) is true

4.6 조건 연산자(Conditional Operator)

조건 연산자는 어떤 조건에 따라 값이 결정되는 연산자입니다. 다음은 조건연산자의 사용 방법을 보이고 있습니다. condition의 결과 값이 true이면 value1의 값이 variablename에 저장되고, 그렇지 않으면 value2의 값이 저장됩니다.

```
variablename = (condition) ? value1 : value2
```

다음은 visitor의 값이 남자일 경우와 그렇지 않을 경우 각각 greeting의 값이 다르게 됩니다.

```
19: greeting = (visitor==" ") ? "안녕하세요. 사장님!" : "안녕하세요. 사모님";
```

5.

5.1 if...else

if...else 문은 switch 문과 함께 조건문 중 하나입니다. if 다음에는 조건식을 표현하며, 조건식문은 조건식의 결과에 따라 다른 행동을 수행하려고 할 때 사용합니다. 자바스크립트에서는 if와 관련된 조건문이 3가지가 있습니다.

- if 문장 : else 문이 없는 if문은 조건이 참일 때만 수행하도록 하기 위해서 사용합니다.
- if...else 문장 : 조건이 참일 때 어떤 문장을 실행 시키고, 조건이 거짓일 경우 다른 문장을 수행시킬 때 사용합니다.
- if...else if...else 문장 : 여러 개 블록 중 하나를 실행시킬 때 사용합니다. 이런 상황에는 switch 문이 사용되기도 합니다.

5.1.1 if

else가 없이 단독으로 사용되는 if문은 지정된 조건식이 참일 경우에만 실행시키기 위해 사용합니다. if는 소문자로 작성해야 합니다.

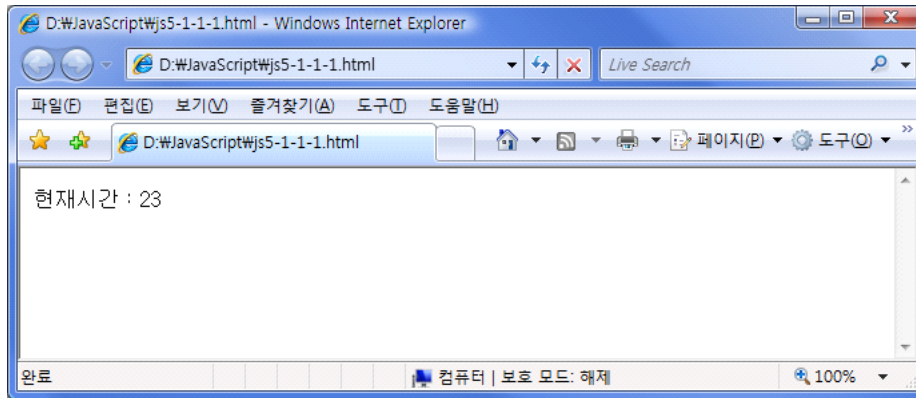
```
if (condition)
{
    //조건식이 참일 경우 실행
}
```

● 예제 코드 1

```
1: <!-- js5-1-1-1.html -->
2:
3: <html>
4: <body>
5:
6: <script type="text/javascript">
7: //time 10보다 작으면 "Good morning" 이라고 출력합니다.
8: var d=new Date();
9: var time=d.getHours();
10:
11: document.write("현재시간 : " + time + "<br>");
12:
13: if (time<10) {
14:     document.write("<b>Good morning</b>");
15: }
16: </script>
17:
18: </body>
```

```
19: </html>
```

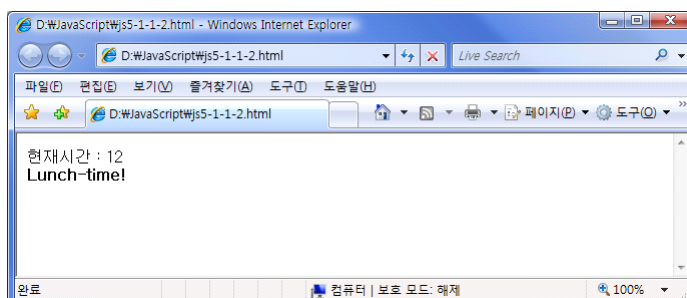
● 실행 결과



● 예제 코드 2

```
1: <!-- js5-1-1-2.html -->
2:
3: <html>
4: <body>
5:
6: <script type="text/javascript">
7: //time 11이면 "Lunch-time!" 이라고 출력합니다.
8: var d=new Date();
9: var time=d.getHours();
10:
11: document.write("현재시간 : " + time + "<br>");
12:
13: if (time==12) {
14:     document.write("<b>Lunch-time!</b>");
15: }
16: </script>
17:
18: </body>
19: </html>
```

● 실행 결과



5.1.2 if...else

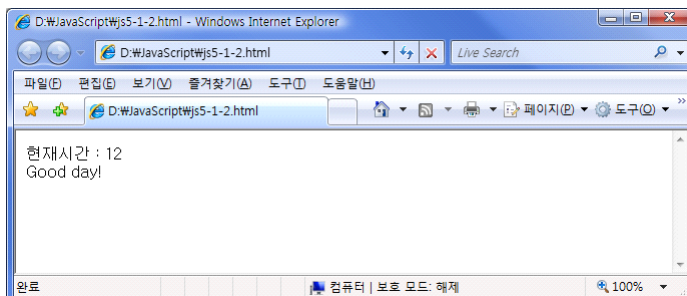
if...else 문은 조건식이 참일 경우에는 if 블록의 내용을 실행시키고, 조건식이 거짓일 경우에는 else 블록의 내용을 실행시킵니다.

```
if (condition)
{
    //조건식이 참이면 실행
}
else {
    //조건식이 거짓이면 실행
}
```

● 예제 코드

```
1: <!-- js5-1-2.html -->
2:
3: <html>
4: <body>
5:
6: <script type="text/javascript">
7: //time 10보다 작으면 "Good morning"이라고 출력하고,
8: //그렇지 않으면 "Good day" 라고 출력합니다.
9: var d = new Date();
10: var time = d.getHours();
11:
12: document.write("현재시간 : " + time + "<br>");
13:
14: if (time < 10) {
15:     document.write("Good morning!");
16: }
17: else {
18:     document.write("Good day!");
19: }
20: </script>
21:
22: </body>
23: </html>
```

● 실행 결과



5.1.3 if...else if...else

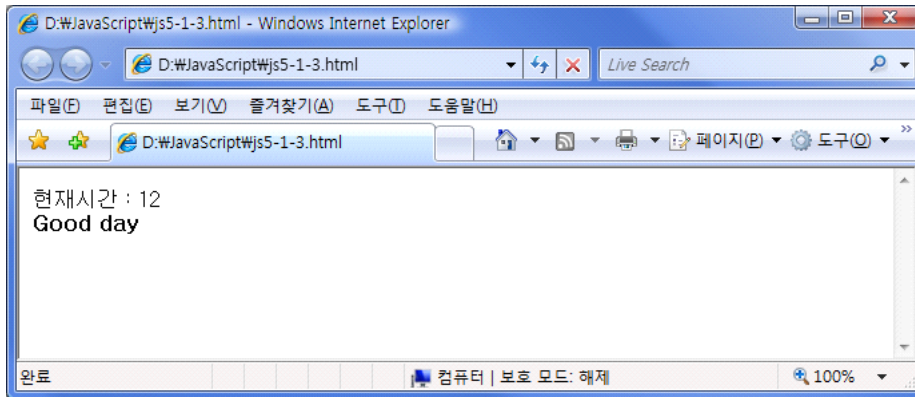
if 다음에 else if 문이 나올 경우에는 여러 개 조건을 두고 조건에 따라 서로 다르게 실행시키기 위해 사용합니다.

```
if (condition1)
{
    //condition1이 참일 경우 실행
}
else if (condition2)
{
    //condition2가 참일 경우 실행
}
else
{
    //condition1과 condition2가 모두 거짓 경우 실행
}
```

● 예제 코드

```
1:  <!-- js5-1-3.html -->
2:
3:  <html>
4:  <body>
5:
6:  <script type="text/javascript">
7:  var d = new Date()
8:  var time = d.getHours()
9:
10: document.write("      : " + time + "<br>");
11:
12: if (time<10) {
13:     document.write("<b>Good morning</b>");
14: }
15: else if (time>10 && time<16) {
16:     document.write("<b>Good day</b>");
17: }
18: else {
19:     document.write("<b>Hello World!</b>");
20: }
21: </script>
22:
23: </body>
24: </html>
```

- 실행 결과



5.2 switch

switch 문은 if 문과 함께 조건문 중 하나입니다. switch 문은 여러 개 블록 중 하나를 선택해서 실행시킬 경우 사용합니다.

```
switch(수식)
{
case 값1:
    수식이 결과가 값1과 같을 때 실행
    break;
case 값2:
    수식이 결과가 값1과 같을 때 실행
    break;
default:
    수식이 결과와 맞는 case의 값이 없을 때 실행
    break;
}
```

switch문에는 수식이 오며 수식의 결과에 따라 case 문장을 실행합니다. 보통의 경우에는 수식에 변수가 오고 변수의 값과 일치하는 case의 문장을 실행합니다. 만일 수식의 값과 일치하는 case 값이 존재하지 않을 경우에는 default 문장이 실행됩니다. 주의해야 할 점은 각 case 문장의 마지막에는 break 문장이 나오는데, 이것은 자동으로 다음 case 문장이 실행되는 것을 방지하기 위한 것입니다. 그리고 case문의 순서는 값과는 무관합니다.

● 예제 코드

```
<!-- js5-2.html -->

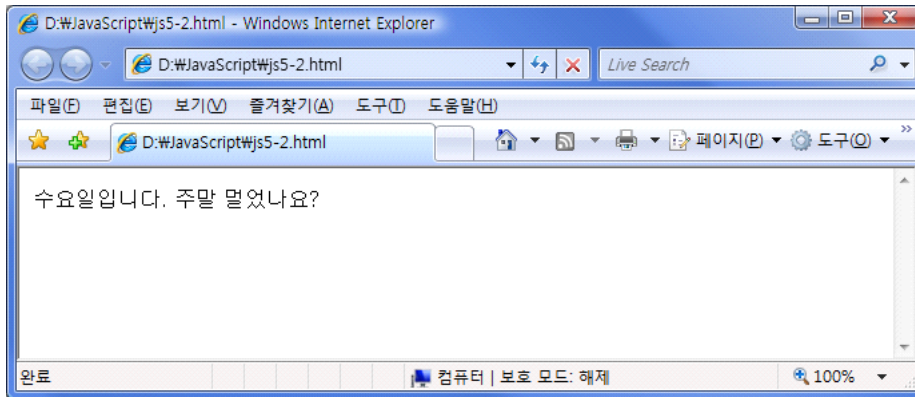
<html>
<body>

<script type="text/javascript">
//      어떤 요일인지에 따라 다른 내용의 인사를 나타냅니다.
//일요일=0, 월요일=1, 화요일=2, ...

var d=new Date();
theDay=d.getDay();
switch (theDay)
{
case 1:
    document.write("월요일입니다. 한 주를 힘차게 시작하세요.");
    break;
case 2:
    document.write("화요일입니다.");
    break;
case 3:
    document.write("수요일입니다. 주말 멀었나요?");
    break;
case 4:
    document.write("목요일입니다.");
    break;
case 5:
    document.write("금요일입니다. 내일이면 즐거운 주말이...");
    break;
case 6:
    document.write("주말입니다. 여행을 떠날까요?");
    break;
case 0:
    document.write("일요일입니다. 아~ 내일 출근해야 합니다.");
    break;
default:
    document.write("오늘이 무슨 요일일까요?");
}
</script>

</body>
</html>
```

● 실행 결과



5.3 for

특정 블록을 반복해서 실행시키고 싶을 때에는 반복문을 사용합니다. 반복문에는 for 문과 while 문이 있습니다.

- for : 반복횟수가 정해져 있을 경우 사용합니다.
- while : 반복 횟수가 지정되어 있지 않을 경우 사용합니다. 지정된 조건이 참일 동안 계속 실행합니다.

for 문은 스크립트를 실행할 문장의 반복 횟수를 알고 있을 경우에 사용합니다.

```
for (var=시작값; 조건식; var=var+증가값)
{
    //실행할 코드
}
```

다음 예제는 i=0부터 시작하여 i값이 10보다 작거나 같을 동안 실행합니다. 그리고 i값은 반복문이 실행된 다음 1씩 증가합니다. 증가 값은 반드시 양수일 필요는 없습니다. 음수가 나올 수도 있으며, 비교문장인 <= 대신 다른 비교 문장이 나올 수 있습니다.

● 예제 코드

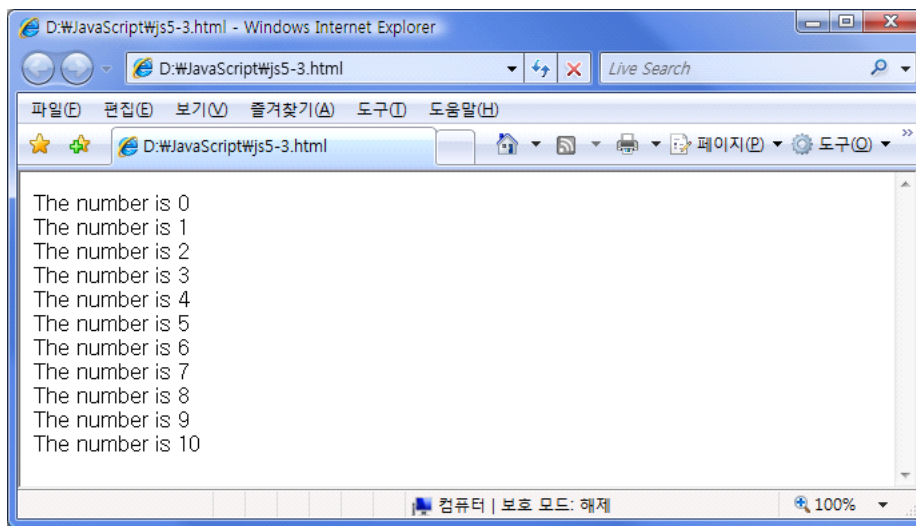
```
1: <!-- js5-3.html -->
2:
3: <html>
4: <body>
5: <script type="text/javascript">
```

```

6:  var i=0;
7:  for (i=0;i<=10;i++)
8:  {
9:      document.write("The number is " + i);
10:     document.write("<br />");
11: }
12: </script>
13: </body>
14: </html>

```

● 실행 결과



5.4 while

while 문은 for 문처럼 반복문으로 사용됩니다. while 문은 주로 반복 횟수가 정해지지 않았을 경우 사용하며, 주어진 조건이 참일 경우 계속 실행합니다.

```

while (조건식)
{
    //실행할 코드
}

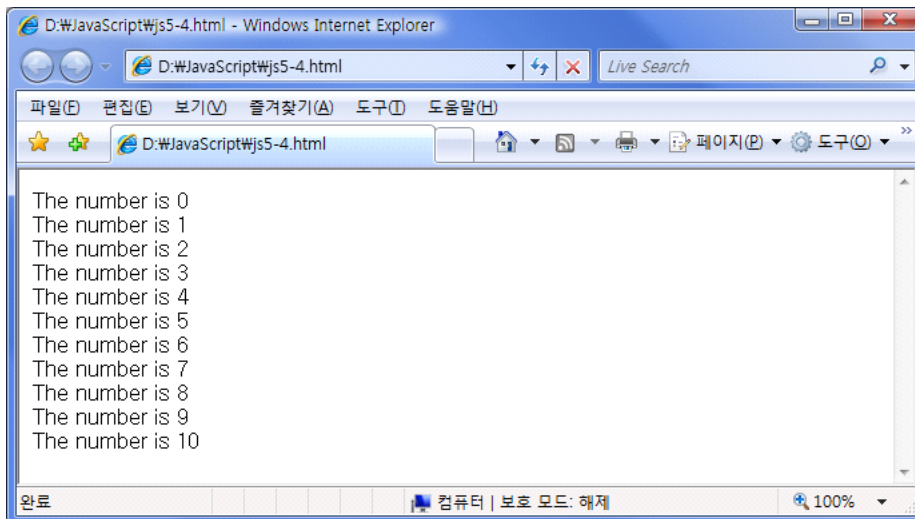
```

다음 예제는 i=0으로 시작하여, i값이 10보다 작거나 같을 때까지 반복 실행합니다. 그리고 i값은 while문의 맨 마지막에서 1씩 증가시킵니다.

● 예제 코드

```
1: <!-- js5-4.html -->
2:
3: <html>
4: <body>
5: <script type="text/javascript">
6:   var i=0;
7:   while (i<=10)
8:   {
9:     document.write("The number is " + i);
10:    document.write("<br />");
11:    i=i+1;
12:  }
13: </script>
14: </body>
15: </html>
```

● 실행 결과



5.5 do...while

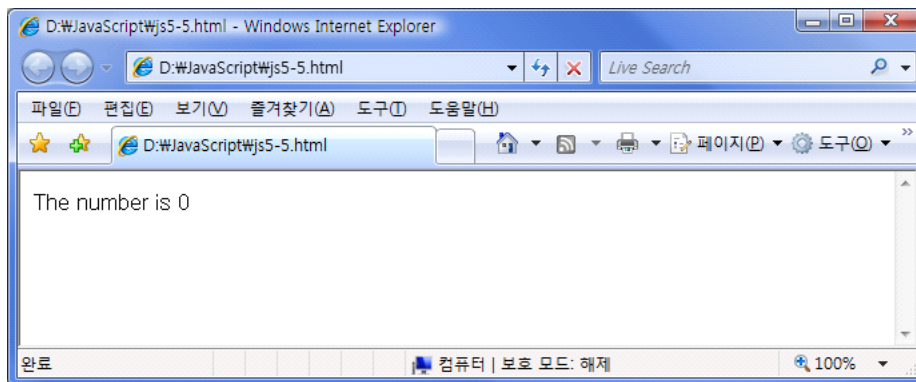
do...while 반복문은 while 반복문의 변형입니다. do...while 문은 조건이 테스트되기 전에 코드 블록이 먼저 실행되기 때문에 조건식의 결과가 참이든 거짓이든 최소한 한번은 수행됩니다.

```
do
{
    실행할 코드
}
while (조건식);
```

- 예제 코드

```
1: <!-- js5-5.html -->
2:
3: <html>
4: <body>
5: <script type="text/javascript">
6: var i=0;
7: do
8: {
9:     document.write("The number is " + i);
10:    document.write("<br />");
11:    i=i+1;
12: }
13: while (i<0);
14: </script>
15: </body>
16: </html>
```

- 실행 결과



5.6 break와 continue

break와 continue문은 반복문 안에서 사용되며, 반복문을 빠져 나가기 위해 사용됩니다.

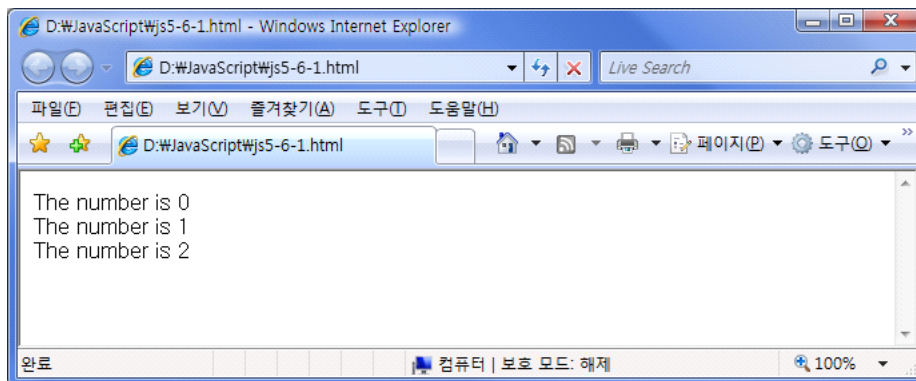
5.6.1 break

break 문은 반복문을 더 이상 실행키치지 않고 빠져 나가는 명령입니다.

● 예제 코드

```
1: <!-- js5-6-1.html -->
2:
3: <html>
4: <body>
5: <script type="text/javascript">
6:   var i=0;
7:   for (i=0;i<=10;i++)
8:   {
9:     if (i==3)
10:    {
11:      break;
12:    }
13:    document.write("The number is " + i);
14:    document.write("<br />");
15:  }
16: </script>
17: </body>
18: </html>
```

● 실행 결과



5.6.2 Continue

continue 문은 반복문 안에서 현재 반복을 건너뛰고 다음 반복 단계로 넘어가는 명령입니다. continue를 만날 때마다 반복문을 skip 한다고 생각하면 됩니다.

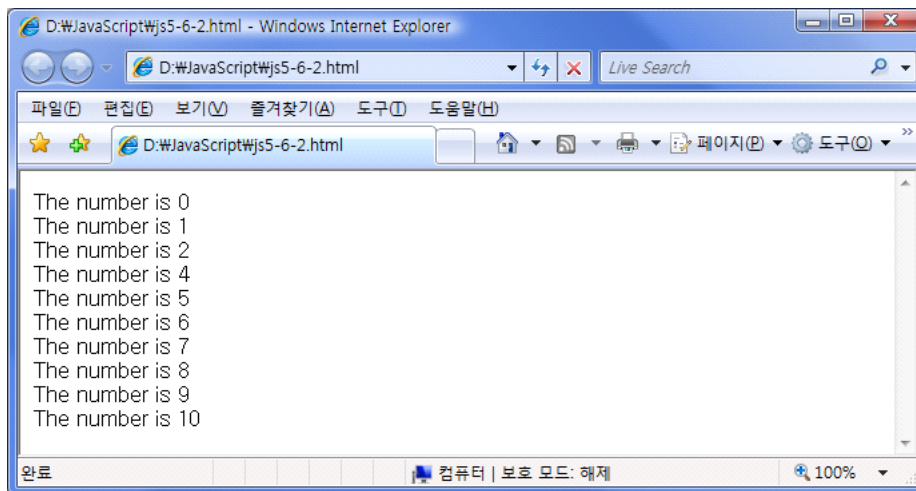
● 예제 코드

```

1:  <!-- js5-6-2.html -->
2:
3:  <html>
4:  <body>
5:  <script type="text/javascript">
6:  var i=0
7:  for (i=0;i<=10;i++)
8:  {
9:      if (i==3)
10:     {
11:         continue;
12:     }
13:     document.write("The number is " + i);
14:     document.write("<br />");
15: }
16: </script>
17: </body>
18: </html>

```

● 실행 결과



5.7 for...in

for...in 문장은 배열 등 열거형 데이터의 값을 자동으로 소비하면서 실행시키는 반복문입니다. 열거형 데이터의 요소 하나당 한 번씩 반복문이 실행됩니다.

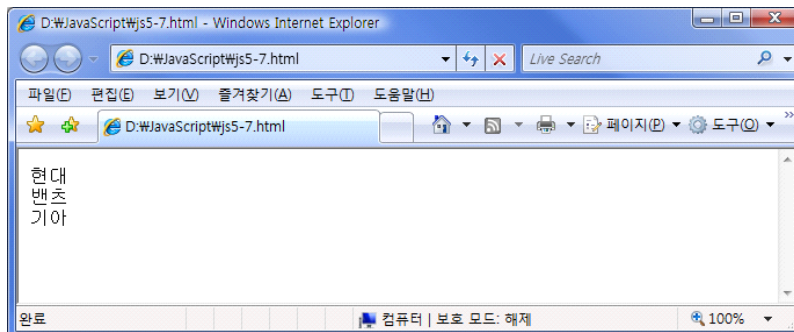
```
for (변수 in 열거형객체)
{
    실행할 코드
}
```

다음 예제는 배열의 데이터를 자동으로 출력합니다.

● 예제 코드

```
20: <!-- js5-7.html -->
21:
22: <html>
23: <body>
24:
25: <script type="text/javascript">
26: var x;
27: var mycars = new Array();
28: mycars[0] = " ";
29: mycars[1] = "벤츠";
30: mycars[2] = "기아";
31:
32: for (x in mycars)
33: {
34:     document.write(mycars[x] + "<br />");
35: }
36: </script>
37:
38: </body>
39: </html>
```

● 실행 결과



6. 박스(Popup Boxes)

자바스크립트에서 alert, confirm 그리고 prompt 팝업 박스가 있으며 이들은 사용자에게 정보를 보여주거나 확인시키도록 하는 기능을 합니다.

6.1 Alert Box

alert 박스는 사용자에게 정보를 확인시키는 기능을 합니다. alert 박스가 나타나면 사용자는 "OK" 버튼을 눌러 팝업 박스를 닫은 후 다음 단계로 진행합니다.

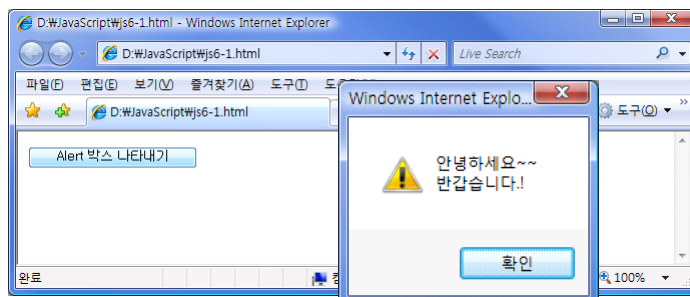
```
alert("sometext");
```

● 예제 코드

```
40: <!-- js6-1.html -->
41:
42: <html>
43: <head>
44: <script type="text/javascript">
45: function disp_alert() {
46:     alert("      ~~" + '\n' + "반갑습니다.!");
47: }
48: </script>
49: </head>
50: <body>
51:
52: <input type="button" onclick="disp_alert()" value="Alert 박스 나타내기" />
53:
54: </body>
55: </html>
```

"\n"은 팝업 박스에 나타내는 문자열에 줄 바꿈 기능을 넣고 싶을 때 사용합니다.

● 실행 결과



6.2 Confirm Box

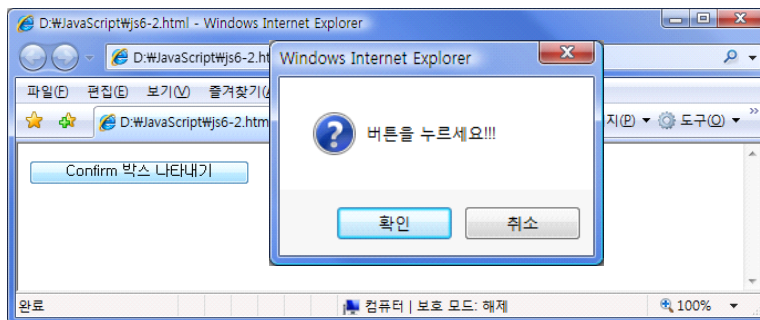
confirm 박스는 사용자가 어떤 것을 받아들이도록 확인시키는 기능을 합니다. confirm 박스가 나타나면 사용자는 "OK" 또는 "Cancel" 중 하나를 선택해서 진행 합니다. 만일 사용자가 "OK"를 클릭하면 true 값이 반환되고, "Cancel"이면 false 값이 반환됩니다.

```
confirm("sometext");
```

● 예제코드

```
1: <!-- js6-2.html -->
2:
3: <html>
4: <head>
5: <script type="text/javascript">
6: function disp_confirm() {
7:     var r=confirm("    누르세요!!!");
8:     if (r==true) {
9:         document.write("OK를 선택 하였습니다.");
10:    }
11:    else {
12:        document.write("Cancel을 선택 하였습니다.");
13:    }
14: }
15: </script>
16: </head>
17: <body>
18:
19: <input type="button" onclick="disp_confirm()" value="Confirm 박스 나타내기" />
20:
21: </body>
22: </html>
```

● 실행 결과



6.2.3 Prompt Box

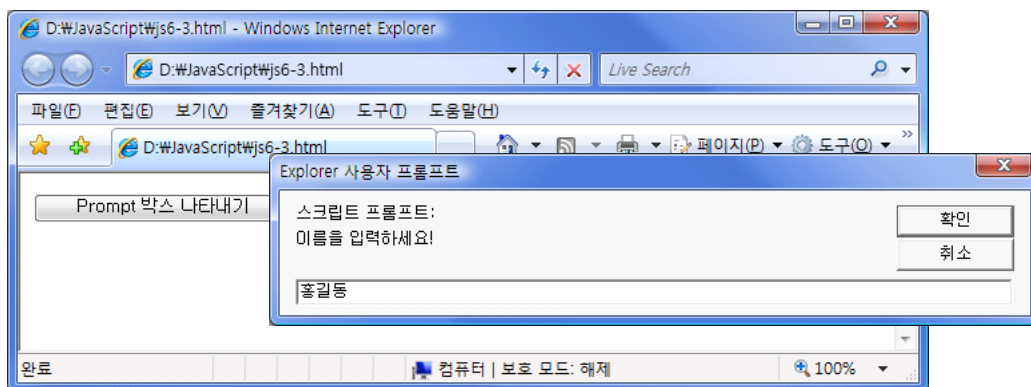
prompt 박스는 사용자가 해당 페이지를 실행하기 전에 특정한 값을 받아들이기 때
사용합니다. prompt 박스가 나타나면 값을 입력한 다음 "OK" 또는 "Cancel" 버튼
을 클릭합니다. 만일 "OK"버튼을 클릭되면 입력된 값을 리턴하고, "Cancel" 버튼을
클릭하면 null 값을 리턴 합니다.

```
prompt("sometext","defaultvalue");
```

● 예제 코드

```
1: <!-- js6-3.html -->
2:
3: <html>
4: <head>
5: <script type="text/javascript">
6: function disp_prompt() {
7:     var name=prompt("    입력하세요!", "홍길동");
8:     if (name!=null && name!="") {
9:         document.write("안녕하세요. " + name + "님! 반갑습니다.");
10:    }
11: }
12: </script>
13: </head>
14: <body>
15:
16: <input type="button" onclick="disp_prompt()" value="Prompt 박스 나타내기" />
17:
18: </body>
19: </html>
```

● 실행 결과



7. 이벤트

7.1 함수(Functions)

자바스크립트 함수는 이벤트에 의해 실행되어지거나 다른 함수에 의해 호출될 때 실행되는 재사용 가능한 코드 블록입니다. 함수를 이용하면 페이지가 로드될 때 브라우저가 특정 스크립트를 실행 시킬 수 있습니다. 함수는 웹페이지 어디서나 호출될 수 있습니다. 또는 자바스크립트가 외부 .js파일로 포함되었다면 다른 페이지에서도 호출될 수 있습니다.

7.1.1 정의

함수는 <head> 부분에 포함될 수 있고, <body> 부분에도 포함될 수 있습니다. 그렇지만 함수는 브라우저가 호출되기 전에 로드되어야 하므로 <head>부분에 넣는 것이 좋습니다. 함수 안의 코드는 반드시 들여쓰기를 할 필요는 없습니다. 그러나 코드를 읽기 좋게 하기 위해서 들여쓰기를 적절하게 하는 것이 좋습니다. 만일 파라미터를 갖지 않는 함수를 정의한다면 함수 이름 뒤에 ()를 붙이면 됩니다. { 와 } 는 함수 시작과 끝을 정의합니다.

```
function functionname()
{
    some code
}
```

자바스크립트는 대/소문자를 구분합니다. 그러므로 함수를 정의할 때와 호출할 때 대/소문자를 구분하여 사용해야 합니다. 그런 면에서 함수는 소문자로 사용하는 것이 좋을 듯 합니다.

● 예제 코드

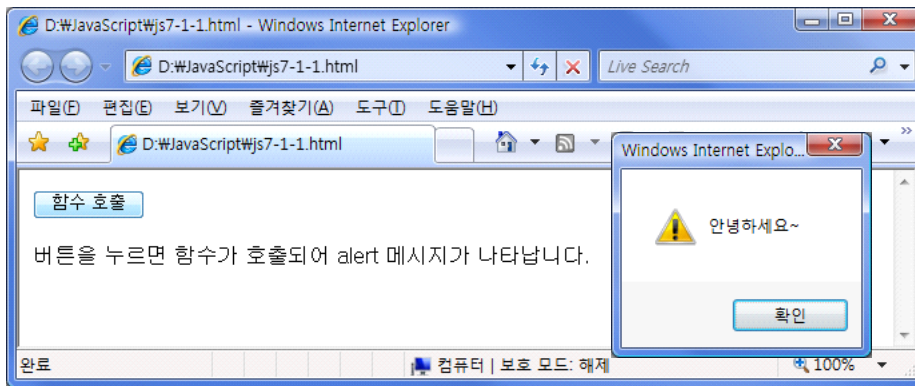
```
1: <!-- js7-1-1.html -->
2:
3: <html>
4: <head>
5:
6: <script type="text/javascript">
7:   function myfunction()
8:   {
9:     alert("      ~");
10:  }
11: </script>
12:
```

```

13: </head>
14: <body>
15:
16: <form>
17: <input type="button" onclick="myfunction()" value="호출">
18: </form>
19:
20: <p>버튼을 누르면 함수가 호출되어 alert 메시지가 나타납니다.</p>
21:
22: </body>
23: </html>

```

● 실행 결과



7.1.2 있는 함수

var1, var2 은 함수에 전달되는 변수들 또는 값들입니다. 이들을 이용하면 함수가 실행되기 위해 필요로 하는 값들을 전달할 수 있습니다.

```

function functionname(var1, var2, ..., varX)
{
    some code
}

```

● 예제 코드

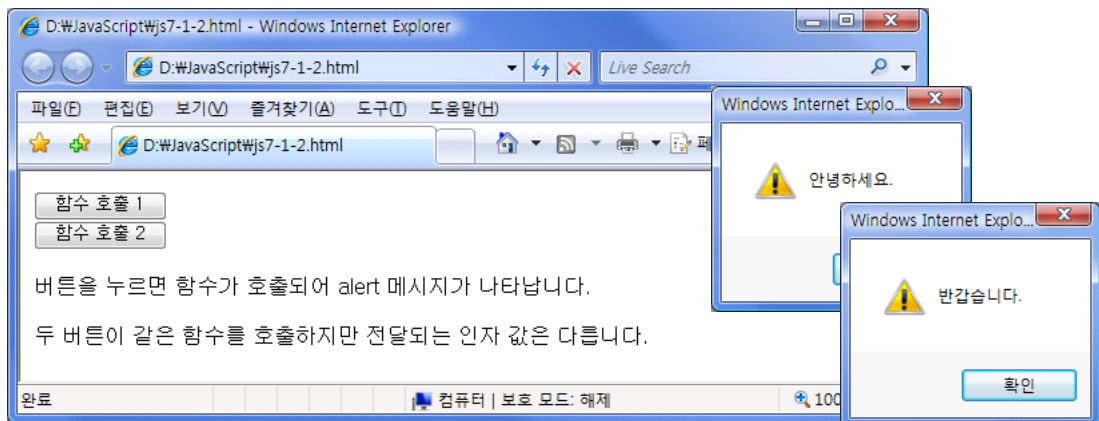
```

1: <!-- js7-1-2.html -->
2:
3: <html>
4: <head>
5:
6: <script type="text/javascript">

```

```
7: function myfunction(txt)
8: {
9:     alert(txt);
10: }
11: </script>
12:
13: </head>
14: <body>
15:
16: <form>
17: <input type="button" onclick="myfunction('      .') "value="함수 호출 1">
18: <br>
19: <input type="button" onclick="myfunction('반갑습니다.') "value="함수 호출 2">
20: </form>
21:
22: <p>버튼을 누르면 함수가 호출되어 alert 메시지가 나타납니다.</p>
23: <p>두 버튼이 같은 함수를 호출하지만 전달되는 인자 값은 다릅니다.</p>
24:
25: </body>
26: </html>
```

● 실행 결과



7.1.3 문(return)

리턴문은 함수로부터 반환되는 값을 지정하기 위해 사용합니다. 값을 반환하기 위한 함수를 작성하기 위해서는 return 문장을 사용해야 합니다.

다음은 두 수(a와 b)의 곱셈 결과를 반환하는 함수입니다.

```
function prod(a,b)
{
    x=a*b;
```

```
    return x;
}
```

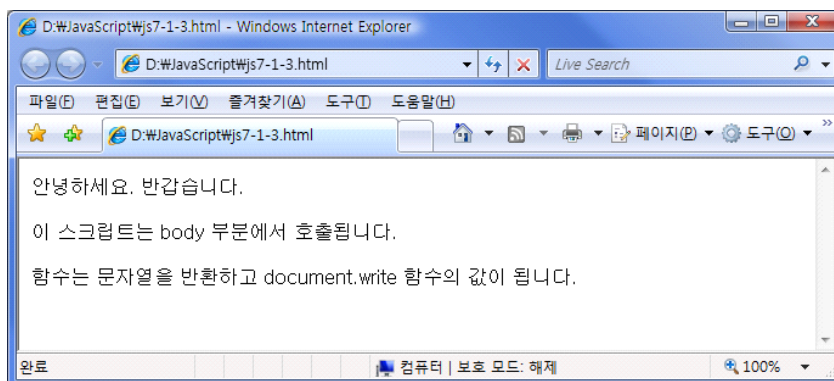
위에 선언된 prod 함수를 호출하기 위해서는 두 개의 파라미터에 값을 넣어줍니다. prod() 함수의 결과는 6이 변수 product에 저장됩니다.

```
product = prod(2,3);
```

● 예제 코드

```
1: <!-- js7-1-3.html -->
2:
3: <html>
4: <head>
5:
6: <script type="text/javascript">
7:   function myFunction()
8:   {
9:     return ("      . 반갑습니다.");
10:  }
11: </script>
12:
13: </head>
14: <body>
15:
16: <script type="text/javascript">
17:   document.write(myFunction())
18: </script>
19:
20: <p>이 스크립트는 body 부분에서 호출됩니다.</p>
21: <p>함수는 문자열을 반환하고 document.write 함수의 값이 됩니다.</p>
22:
23: </body>
24: </html>
```

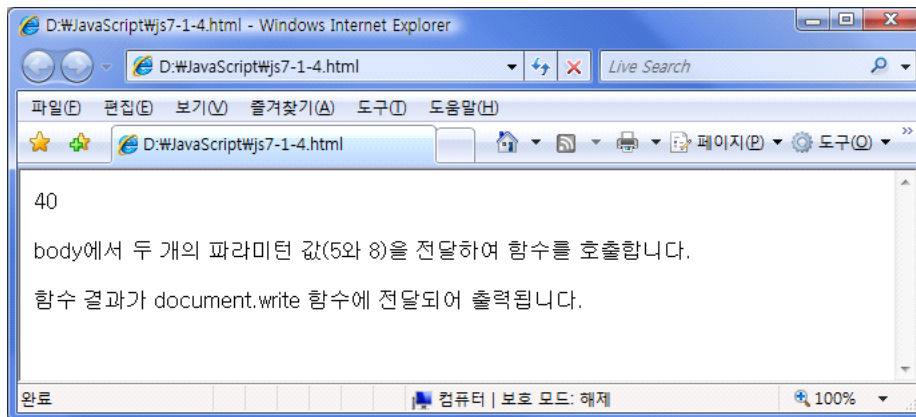
● 실행 결과



- 예제 코드

```
1: <!-- js7-1-4.html -->
2:
3: <html>
4: <head>
5: <script type="text/javascript">
6:   function product(a, b)
7:   {
8:     return a*b;
9:   }
10: </script>
11: </head>
12:
13: <body>
14: <script type="text/javascript">
15:   document.write(product(5,8));
16: </script>
17:
18: <p>body    두 개의 파라미터 값 (5와 8)을 전달하여 함수를 호출합니다.</p>
19: <p>함수 결과가 document.write 함수에 전달되어 출력됩니다.</p>
20: </body>
21: </html>
```

- 실행 결과



7.2 이벤트(Events)

이벤트는 자바스크립트에 의해 탐지될 수 있는 동작을 뜻합니다. 이벤트를 이용하면 좀 더 동적인 웹페이지(dynamic web page)를 만들 수 있습니다. 웹페이지의 모든 요소(elements)들은 자바스크립트의 내장 함수를 호출할 수 있는 특정한 이벤트들을 가지고 있습니다.

사용자가 버튼을 클릭하였을 때 버튼은 그 동작을 감지하여 함수를 실행시키고 싶을 경우 마우스의 onClick 이벤트를 사용할 수 있습니다. 이벤트는 HTML 태그 안에 정의합니다. 이벤트는 일반적으로 함수와 같이 사용합니다. 그리고 함수는 이벤트가 발생하기 전에는 실행되지 않을 것입니다.

● 이벤트 예

- 마우스 클릭
- 웹 페이지 또는 이미지 로딩
- 마우스가 웹페이지의 특정 지점 위로 올려지는 것
- HTML 폼에서 선택하거나 값을 입력 하는 것
- HTML 폼의 값 전송
- 키 입력

7.2.1 핸들러

HTML 4.0 이후부터 브라우저에서 HTML 이벤트를 사용할 수 있도록 하였습니다. 이벤트를 다루기 위해서는 어떤 종류의 이벤트 들이 있는지 먼저 알아야 합니다. 다음은 HTML 태그에 삽입되어 질 수 있는 이벤트 액션들의 목록입니다.

속성	이벤트가 발생할 때...
onabort	이미지 로딩이 중지 되었을 때
onblur	포커스를 잃을 때
onchange	필드의 내용이 변경 되었을 때
onclick	마우스 클릭
ondblclick	마우스 더블클릭
onerror	문서 또는 이미지 로딩 중 에러가 발생할 때
onfocus	포커스를 얻을 때
onkeydown	키보드의 키가 눌러질 때
onkeypress	키보드의 키가 눌러지거나 눌러져 있는 때
onkeyup	키보드의 키가 놓일 때

onload	페이지 또는 이미지 로딩이 완료 되었을 때
onmousedown	마우스 버튼일 눌러질 때
onmousemove	마우스 포인터가 움직일 때
onmouseout	마우스가 지정된 영역 밖으로 나갈 때
onmouseover	마우스가 지정된 영역 안으로 들어올 때
onmouseup	마우스 버튼이 놓일 때
onreset	reset 버튼일 클릭되었을 때
onresize	윈도우 또는 프레임의 크기가 변경될 때
onselect	텍스트가 선택되어질 때
onsubmit	submit 버튼이 클릭되어질 때
onunload	페이지를 나갈 때

7.2.2 onClick

onClick 이벤트는 객체가 클릭되었을 때 발생합니다.

● 지원하는 HTML 태그

- onClick : <a>, <address>, <area>, , <bdo>, <big>, <blockquote>, <body>, <button>, <caption>, <cite>, <code>, <dd>, <dfn>, <div>, <dl>, <dt>, , <fieldset>, <form>, <h1> to <h6>, <hr>, <i>, , <input>, <kbd>, <label>, <legend>, , <map>, <object>, , <p>, <pre>, <samp>, <select>, <small>, , , <sub>, <sup>, <table>, <tbody>, <td>, <textarea>, <tfoot>, <th>, <thead>, <tr>, <tt>, , <var>

● 지원하는 자바스크립트 객체

- onClick : button, document, checkbox, link, radio, reset, submit

● 예제 코드 (onClick)

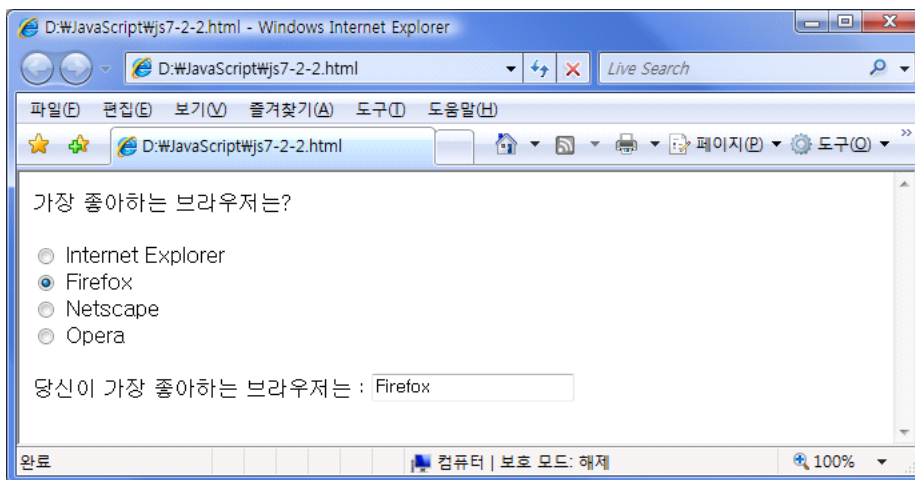
```
1: <!-- js7-2-2.html -->
2:
3: <html>
4: <head>
5: <script type="text/javascript">
```

```

6: function check(browser)
7: {
8:     document.getElementById("answer").value=browser;
9: }
10: </script>
11: </head>
12: <body>
13:
14: <p>    좋아하는 브라우저는?</p>
15:
16: <form>
17: <input type="radio" name="browser" onclick="check(this.value)" value="Internet Explorer"> Internet Explorer<br />
18: <input type="radio" name="browser" onclick="check(this.value)" value="Firefox"> Firefox<br />
19: <input type="radio" name="browser" onclick="check(this.value)" value="Netscape"> Netscape<br />
20: <input type="radio" name="browser" onclick="check(this.value)" value="Opera"> Opera<br />
21: <br />
22: 당신이 가장 좋아하는 브라우저는 : <input type="text" id="answer" size="20">
23: </form>
24:
25: </body>
26: </html>

```

● 실행 결과



7.2.3 onLoad and onUnload

onLoad와 onUnload 이벤트는 사용자가 페이지에 들어오거나 나갈 때 발생합니다. onLoad 이벤트는 방문자가 브라우저의 버전이나 타입을 체크하여 그 정보에 따라서 웹페이지를 보여주기 위해 사용될 수 있습니다. 그리고 onLoad와 onUnload 이벤트는 사용자가 페이지에 들어오거나 나갈 때 쿠키의 값을 지정하기 위해 사용할 수도 있습니다. 예를 들면, 사용자가 당신의 페이지에 처음 들어왔을 때 당신의 이름을 묻는 popop 대화상자가 있었다면 입력한 이름이 쿠키에 저장되고, 다음 방문했을 때에는 "안녕하세요. 홍길동님." 과 같은 다른 popup 대화상자를 나타낼 수 있을 것입니다.

onLoad 이벤트는 페이지 또는 이미지가 로딩 되었을 경우, onUnload 이벤트는 브라우저를 종료하거나, 다른 페이지로 이동 등으로 인해 페이지를 벗어날 경우 발생합니다.

● 지원하는 HTML 태그

- onLoad : <body>, <frame>, <frameset>, <iframe>, , <link>, <script>
- onUnload : <body>, <frameset>

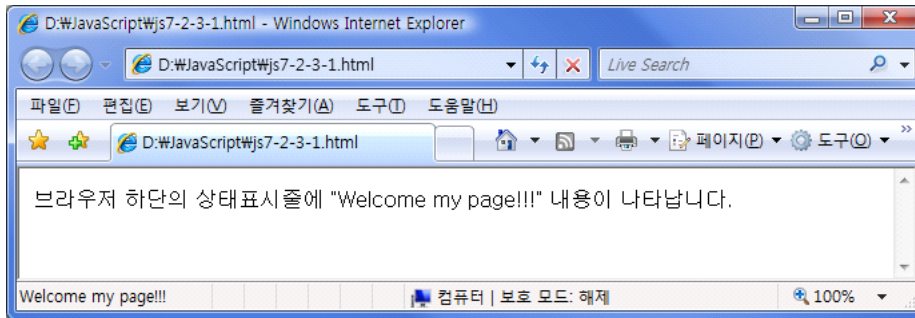
● 지원하는 자바스크립트 객체

- onLoad : image, layer, window
- onUnload : window

● 예제 코드 (onLoad)

```
1:  <!-- js7-2-3-1.html -->
2:
3:  <html>
4:  <head>
5:  <script type="text/javascript">
6:  function load()
7:  {
8:  window.status="Welcome my page!!!";
9:  }
10: </script>
11: </head>
12:
13: <body onload="load()">
14:     하단의 상태표시줄에 "Welcome my page!!!" 내용이 나타납니다.
15: </body>
16:
17: </html>
```

- 실행 결과



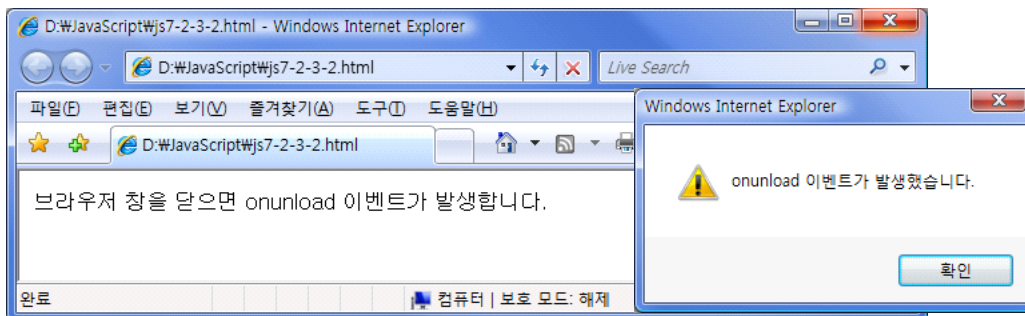
- 예제 코드 (onUnload)

```

56: <!-- js7-2-3-2.html -->
57:
58: <html>
59:
60: <body onunload="alert('onunload      발생했습니다.')" ">
61: <p>브라우저 창을 닫으면 onunload 이벤트가 발생합니다.</p>
62: </body>
63:
64: </html>

```

- 실행 결과



7.2.4 onFocus, onBlur and onChange

onFocus, onBlur 그리고 onChange 이벤트들은 폼 필드들에서 주로 사용됩니다. onFocus 이벤트는 포커스를 얻으면 이벤트가 발생하고, onBlur 이벤트는 onFocus와는 반대로 포커스를 잃으면 이벤트가 발생합니다. onChange 이벤트는 폼 필드에서 내용이 변경되면 이벤트가 발생합니다.

- 지원하는 HTML 태그

- onFocus, onBlur : <a>, <acronym>, <address>, <area>, , <bdo>, <big>, <blockquote>, <button>, <caption>, <cite>, <dd>, , <dfn>, <div>, <dl>, <dt>, , <fieldset>, <form>, <frame>, <frameset>, <h1> to <h6>, <hr>, <i>, <iframe>, , <input>, <ins>, <kbd>, <label>, <legend>, , <object>, , <p>, <pre>, <q>, <samp>, <select>, <small>, , , <sub>, <sup>, <table>, <tbody>, <td>, <textarea>, <tfoot>, <th>, <thead>, <tr>, <tt>, , <var>
- onChange : <input type="text">, <select>, <textarea>

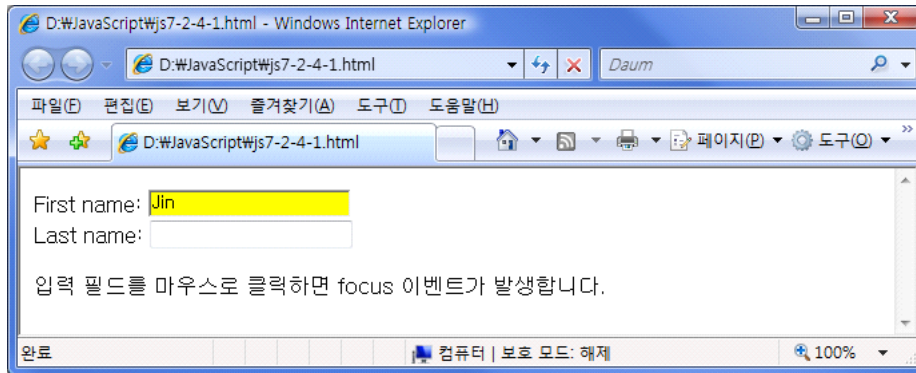
- 지원하는 자바스크립트 객체

- onFocus, onBlur : button, checkbox, fileUpload, layer, frame, password, radio, reset, select, submit, text, textarea, window
- onChange : fileUpload, select, text, textarea

- 예제 코드 (onFocus)

```
1:  <!-- js7-2-4-1.html -->
2:
3:  <html>
4:  <head>
5:  <script type="text/javascript">
6:  function setStyle(x)
7:  {
8:      document.getElementById(x).style.background = "yellow";
9:  }
10: </script>
11: </head>
12: <body>
13:
14: First name: <input type="text" onfocus="setStyle(this.id)" id="fname">
15: <br />
16: Last name: <input type="text" onfocus="setStyle(this.id)" id="lname">
17:
18: <p> 필드를 마우스로 클릭하면 focus 이벤트가 발생합니다.</p>
19:
20: </body>
21: </html>
```

- 실행 결과



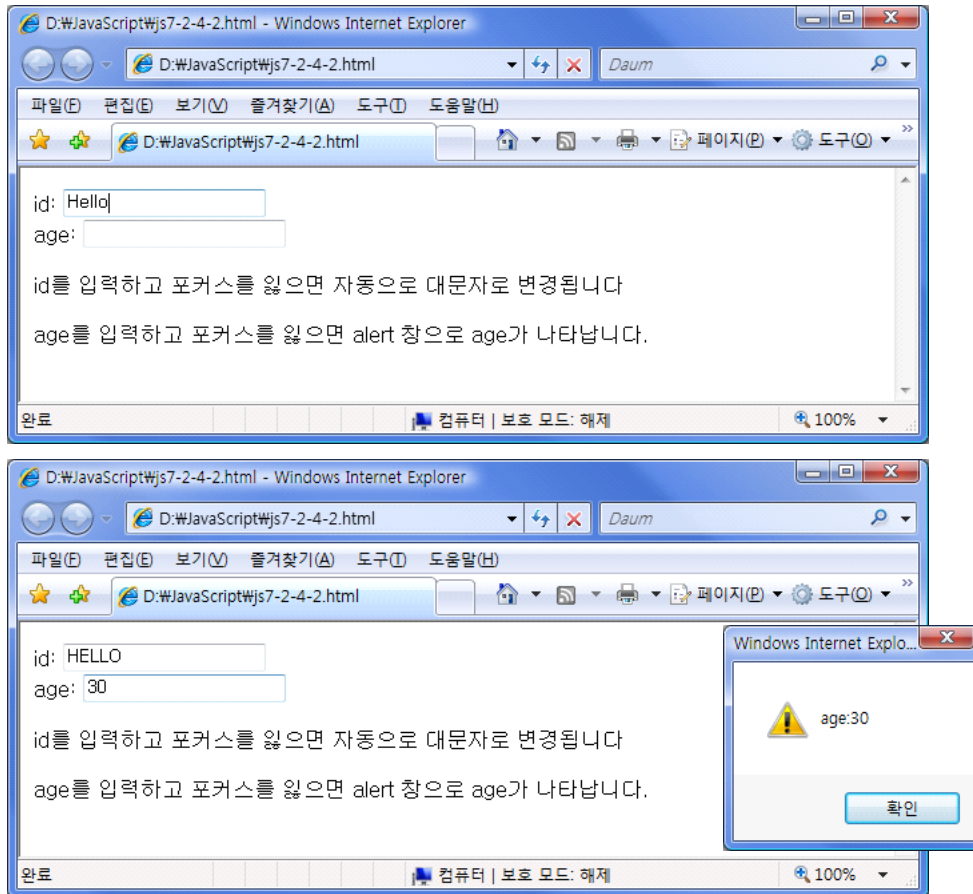
- 예제 코드 (onBlur)

```

1:  <!-- js7-2-4-2.html -->
2:
3:  <html>
4:
5:  <head>
6:  <script type="text/javascript">
7:    function upperCase()
8:    {
9:      var x = document.getElementById("fid").value;
10:     document.getElementById("fid").value = x.toUpperCase();
11:    }
12:  </script>
13: </head>
14:
15: <body>
16:
17: id: <input type="text" id="fid" onblur="upperCase()"><br />
18: age: <input type="text" id="age" onblur="alert(this.id + ":" + this.id.value)">
19:
20: <p>id   입력하고 포커스를 잃으면 자동으로 대문자로 변경됩니다</p>
21: <p>age를 입력하고 포커스를 잃으면 alert 창으로 age가 나타납니다.</p>
22:
23: </body>
24: </html>

```

- 실행 결과



- 예제 코드 (onChange)

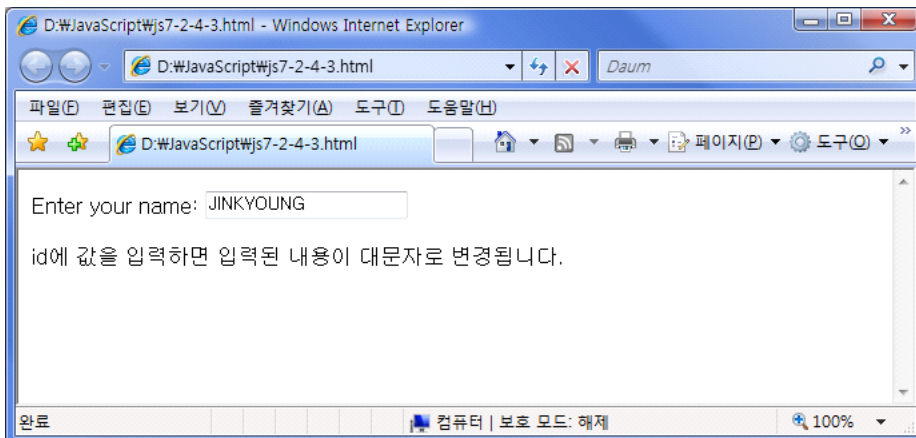
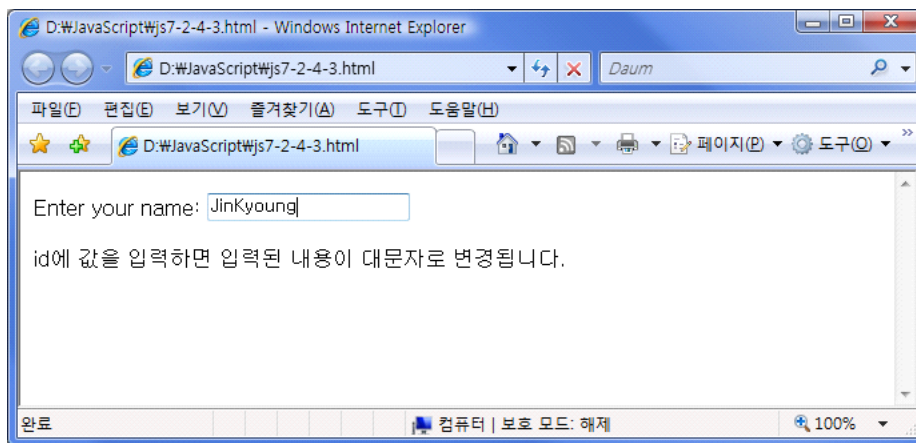
```
1: <!-- js7-2-4-3.html -->
2:
3: <html>
4:
5: <head>
6: <script type="text/javascript">
7:   function upperCase(x)
8:   {
9:     var y = document.getElementById(x).value;
10:    document.getElementById(x).value = y.toUpperCase();
11:  }
12: </script>
13: </head>
14:
```

```

15: <body>
16:
17: Enter your name: <input type="text" id="fname" onchange="upperCase(this.id)">
18:
19: <p>id 값을 입력하면 입력된 내용이 대문자로 변경됩니다.</p>
20:
21: </body>
22: </html>

```

- 실행 결과



7.2.5 onSubmit

onsubmit 폼에서 submit 버튼이 클릭되었을 경우 발생합니다. 이 이벤트를 사용하면 폼 필드의 내용들이 전송하기 전에 값들이 올바르게 입력되었는지 확인할 수 있습니다.

- 지원하는 HTML 태그

- onSubmit : <form>

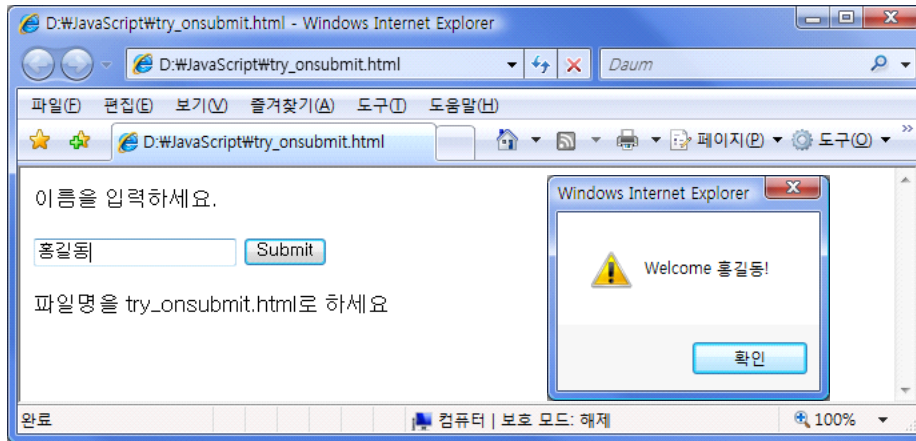
- 지원하는 자바스크립트 객체

- onSubmit : form

- 예제 코드 (onSubmit)

```
1:  <!-- try_onsubmit.html -->
2:
3:  <html>
4:  <head>
5:  <script type="text/javascript">
6:  function welcome()
7:  {
8:      alert("Welcome " + document.forms["myform"]["fname"].value + "!")
9:  }
10: </script>
11: </head>
12:
13: <body>
14:
15:     입력하세요.<br />
16: <form name="myform" action="try_onsubmit.html" onSubmit="welcome()">
17:     <input type="text" name="fname" size="20">
18:     <input type="submit" value="Submit">
19: </form>
20:
21: <p>파일명을 try_onsubmit.html로 하세요</p>
22:
23: </body>
24: </html>
```

- 실행 결과



7.2.6 onMouseOver and onMouseOut

onMouseOver 와 onMouseOut 이벤트는 "animated" 버튼을 만들 때 사용합니다. onMouseOver 이벤트는 지정된 객체 위로 마우스 포인터가 움직일 경우 발생되고, 지정된 객체 위에서 마우스 포인터가 옮겨 나갈 경우 발생합니다.

- 지원하는 HTML 태그

- onMouseOver, onMouseOut : <a>, <address>, <area>, , <bdo>, <big>, <blockquote>, <body>, <button>, <caption>, <cite>, <code>, <dd>, <dfn>, <div>, <dl>, <dt>, , <fieldset>, <form>, <h1> to <h6>, <hr>, <i>, , <input>, <kbd>, <label>, <legend>, , <map>, , <p>, <pre>, <samp>, <select>, <small>, , , <sub>, <sup>, <table>, <tbody>, <td>, <textarea>, <tfoot>, <th>, <thead>, <tr>, <tt>, , <var>

- 지원하는 자바스크립트 객체

- onMouseOve, onMouseOutr : layer, link

- 예제 코드 (onMouseOver, onMouseOut)

```
1: <!-- js7-2-6.html -->
2:
3: <html>
4: <head>
```

```
5: <script type="text/javascript">
6: function mouseOver()
7: {
8:     document.getElementById("b1").src = "1.jpg";
9: }
10: function mouseOut()
11: {
12:     document.getElementById("b1").src = "2.jpg";
13: }
14: </script>
15: </head>
16:
17: <body>
18: <a href="http://www.javaspecialist.co.kr" target="_blank">
19: </a>
20:
21: <p>    위로 마우스를 가져가면 이미지가 바뀝니다.</p>
22: <p>예제 테스트를 위해서 두개의 다른 그림파일(1.jpg, 2.jpg)이 필요합니다.</p>
23:
24: </body>
25: </html>
```

● 실행 결과



7.2.7 onKeyDown

onKeyDown 이벤트는 키보드의 키가 눌러질 때 발생합니다. 브라우저에 따라서 눌러진 문자를 받아들이는 방법이 다릅니다. 익스플로러에서는 event.keyCode를 이용하여 눌러진 문자를 알 수 있고, Netscape, Firefox, Opera 에서는 event.which를 이용해야 합니다.

● 지원하는 HTML 태그

- onClick : <a>, <acronym>, <address>, <area>, , <bdo>, <big>, <blockquote>, <body>, <button>, <caption>, <cite>, <code>, <dd>, , <dfn>, <div>, <dt>, , <fieldset>, <form>, <h1> to <h6>, <hr>, <i>, <input>, <kbd>, <label>, <legend>, , <map>, <object>, , <p>, <pre>, <q>, <samp>, <select>, <small>, , , <sub>, <sup>, <table>, <tbody>, <td>, <textarea>, <tfoot>, <th>, <thead>, <tr>, <tt>, , <var>

● 지원하는 자바스크립트 객체

- onClick : document, image, link, textarea

● 예제 코드 (onKeyDown)

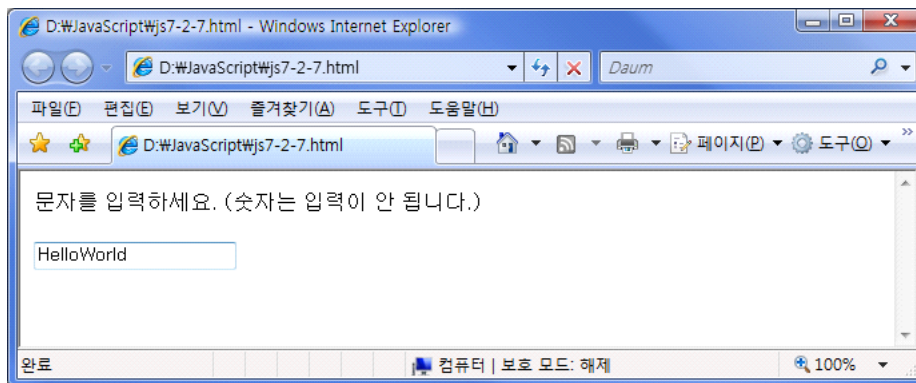
```

1: <!-- js7-2-7.html -->
2:
3: <html>
4: <body>
5: <script type="text/javascript">
6: function noNumbers(e)
7: {
8:   var keynum;
9:   var keychar;
10:  var numcheck;
11:
12:  if(window.event) // IE
13:  {
14:    keynum = e.keyCode;
15:  }
16:  else if(e.which) // Netscape/Firefox/Opera
17:  {
18:    keynum = e.which;
19:  }
20:  keychar = String.fromCharCode(keynum);
21:  numcheck = /\d/;
22:  return !numcheck.test(keychar);

```

```
23: }  
24: </script>  
25:  
26: <form>  
27: <p>    입력하세요. (숫자는 입력이 안 됩니다.)</p>  
28: <input type="text" onkeydown="return noNumbers(event)" />  
29: </form>  
30:  
31: </body>  
32: </html>
```

● 실행 결과



8. 처리

사용자가 인터넷에서 웹페이지를 돌아다닐 때, 런타임 에러로 인해 경고창이 나타날 경우가 있습니다. 이때 "디버그 하겠습니까?"란 경고메시지를 보게 됩니다. 또는 자바스크립트 문법적 오류로 인해 작업표시줄에 "페이지에 오류가 있습니다"란 메시지가 나타나고 자바스크립트가 정상적으로 실행되지 않을 경우를 본 적이 있을 것입니다. 이런 에러 메시지는 개발자에게는 유용한 정보가 되지만 사용자들에게는 불필요한 메시지입니다. 만일 사용자들이 이런 메시지를 보게 된다면 해당 사이트에 대하여 신뢰를 갖지 않을 수도 있습니다.

이 장에서는 자바스크립트 어떻게 에러 메시지를 잡고, 다루는지에 대하여 설명하고 있습니다. 자바스크립트에서 에러를 다루는 방법은 두 가지가 있습니다. 하나는 try...catch 문장을 사용하는 방법이고, 다른 하나는 onerror 이벤트를 사용하는 방법입니다. try...catch 를 이용하는 방법은 IE5+, Mozilla 1.0, Netscape 6 버전 이상에서 사용 가능합니다.

8.1 Try...Catch

try...catch 문장은 에러가 발생 가능한 문장 블록을 테스트 할 수 있습니다. try 블록은 실행된 코드를 삽입하고, catch 블록은 try 블록의 코드에서 에러가 발생할 경우 실행할 문장을 넣습니다.

try와 catch는 자바스크립트의 키워드 이므로 소문자로 작성해야 합니다.

```
try
{
    //Run some code here
}
catch(err)
{
    //Handle errors here
}
```

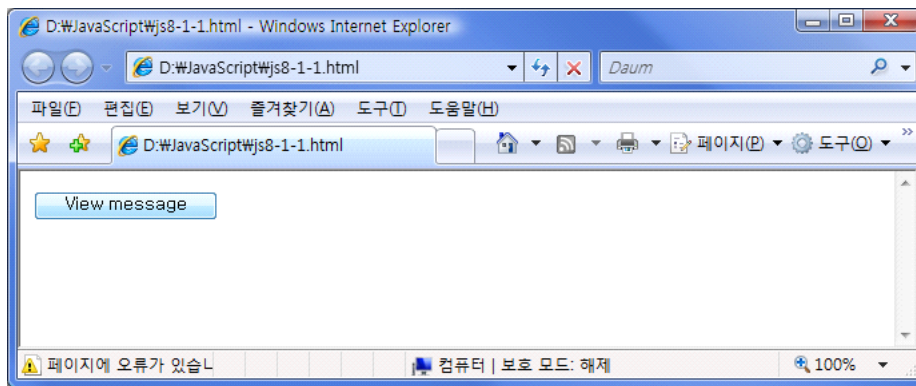
● 예제 코드

다음은 alert() 메소드를 adlert()으로 잘못 입력하여서 정상적으로 작동하지 않는 스크립트 예입니다. 아래 예제에서는 버튼을 클릭하면 정상적으로 실행되지 않고 브라우저의 상태표시줄에 "페이지에 오류가 있습니다"라고 표시됩니다.

```
1: <!-- js8-1-1.html -->
2:
3: <html>
```

```
4: <head>
5: <script type="text/javascript">
6: function message()
7: {
8:     alert("Welcome guest!");
9: }
10: </script>
11: </head>
12:
13: <body>
14: <input type="button" value="View message" onclick="message()" />
15: </body>
16:
17: </html>
```

● 실행 결과



● 예제 코드

다음 try...catch 를 이용하여 앞의 코드를 수정하였습니다. try 블록에서 에러가 발생하였을 경우 catch 블록이 수행되는 것을 볼 수 있습니다.

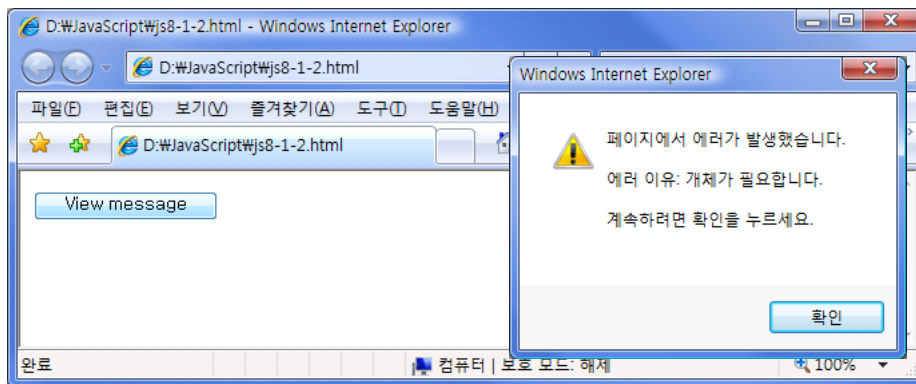
```
1: <!-- js8-1-2.html -->
2:
3: <html>
4: <head>
5: <script type="text/javascript">
6: var txt=""
7: function message()
8: {
9:     try
10:    {
11:        alert("Welcome guest!");
12:    }
13:    catch(err)
```

```

14:  {
15:      txt="      에러가 발생했습니다.\n\n";
16:      txt+="에러 이유: " + err.description + "\n\n";
17:      txt+="계속하려면 확인을 누르세요.\n\n";
18:      alert(txt);
19:  }
20: }
21: </script>
22: </head>
23:
24: <body>
25: <input type="button" value="View message" onclick="message()" />
26: </body>
27:
28: </html>

```

● 실행 결과



8.2 throw

throw 문장은 예외를 만드는데 사용합니다. throw 문장은 try...catch 문장과 함께 사용할 경우 더 정확한 에러메시지를 발생시키고, 프로그램의 흐름을 제어할 수 있습니다. throw 는 키워드 이므로 소문자로 작성해야 합니다.

```
throw(exception)
```

● 예제 코드

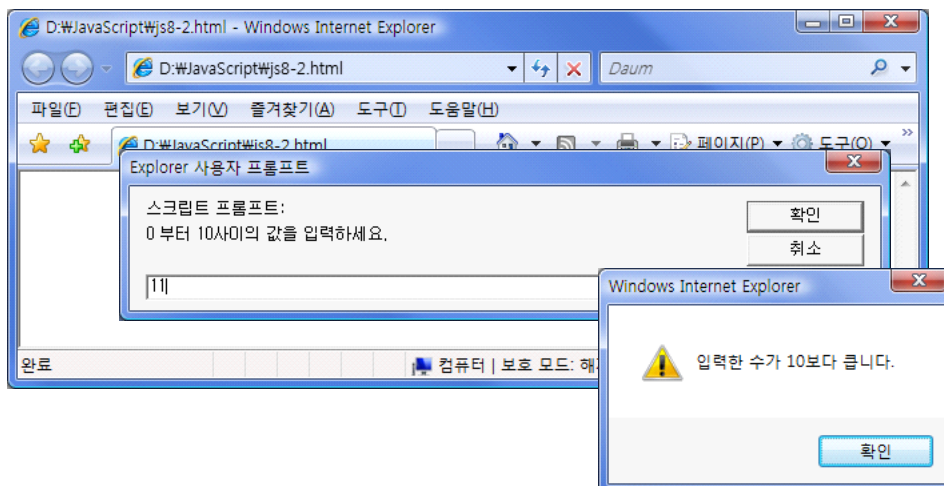
```

1: <!-- js8-2.html -->
2:
3: <html>

```

```
4: <body>
5:
6: <script type="text/javascript">
7: var x=prompt("0    10사이의 값을 입력하세요.", "");
8: try
9: {
10:   if(x>10)
11:   {
12:     throw "Err1";
13:   }
14:   else if(x<0)
15:   {
16:     throw "Err2";
17:   }
18:   document.write(x);
19: }
20: catch(er)
21: {
22:   if(er == "Err1")
23:   {
24:     alert("입력한 수가 10보다 큼니다.");
25:   }
26:   if(er == "Err2")
27:   {
28:     alert("입력한 수가 0보다 작습니다.");
29:   }
30: }
31: </script>
32:
33: </body>
34: </html>
```

● 실행 결과



8.3 onerror 이벤트

onerror 이벤트는 페이지 내에서 스크립트 에러가 있을 때에 발생합니다. onerror 이벤트를 사용하기 위해서는 에러를 다루는 함수를 작성해야 합니다. 이러한 함수를 이벤트 핸들러라 부릅니다. onerror 이벤트가 발생했을 경우 미리 작성된 이벤트 핸들러 함수를 호출하도록 합니다. onerror 이벤트를 처리하는 함수는 msg(에러 메시지), url(에러가 발생한 페이지 url), line(에러가 발생한 라인) 세 인자를 가질 수 있습니다.

```
onerror=handleErr;
```

```
function handleErr(msg, url, line)
{
    //Handle the error here
    return true or false
}
```

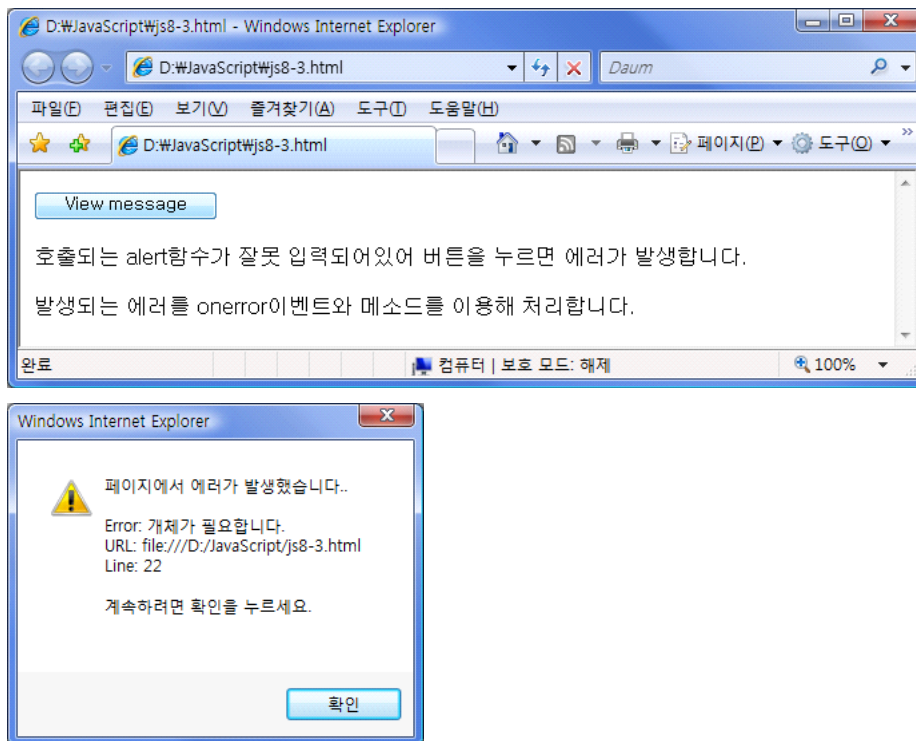
onerror에 의해 반환되는 값은 브라우저가 표준 에러메시지를 표시할 것인가를 결정합니다. 만일 반환되는 값이 false일 경우에는 브라우저는 자바스크립트 콘솔에 표준 에러메시지를 표시합니다. 만일 반환되는 값이 true일 경우에는 표준 에러메시지를 표시하지 않습니다.

● 예제 코드

```
1:  <!-- js8-3.html -->
2:
3:  <html>
4:  <head>
5:  <script type="text/javascript">
6:  onerror=handleErr;
7:  var txt="";
8:
9:  function handleErr(msg,url,l)
10: {
11:     txt="          에러가 발생했습니다..\n\n";
12:     txt+="Error: " + msg + "\n";
13:     txt+="URL: " + url + "\n";
14:     txt+="Line: " + l + "\n\n";
15:     txt+="계속하려면 확인을 누르세요.\n\n";
16:     alert(txt);
17:     return true;
18: }
19:
20: function message()
```

```
21: {  
22:   alert("Welcome guest!");  
23: }  
24: </script>  
25: </head>  
26:  
27: <body>  
28:  
29: <input type="button" value="View message" onclick="message()" />  
30:  
31: <p>       alert함수가 잘못 입력되어있어 버튼을 누르면 에러가 발생합니다.</p>  
32: <p>발생되는 에러를 onerror이벤트와 메소드를 이용해 처리합니다.</p>  
33:  
34: </body>  
35: </html>
```

● 실행 결과



9. (Special Text)

JavaScript에서 "\"(backslash)를 이용하여 '(apostrophe), 줄바꿈, "(quote)등 특수 문자를 사용할 수 있습니다.

다음 코드의 일부를 보겠습니다. 자바스크립트에서는 문자열이 ' 또는 "로 시작하여 끝납니다. 만일 '로 시작하였을 경우에는 '로 끝이 나고, "로 시작하였을 경우에는 "로 끝나야 합니다. 그래서 아래의 코드에서는 text 변수에는 문자열이 "우리는 그를"까지만 저장됩니다.

```
var txt="우리는 그를 "홍길동"이라고 부릅니다.";
document.write(txt);
```

이 문제를 해결하기 위해서 "홍길동"에서 "앞에 \를 붙여주는 것입니다.

```
var txt="우리는 그를 \"홍길동\"이라고 부릅니다.";
document.write(txt);
```

다음은 &를 표시하기 위한 코드입니다.

```
document.write ("You \& I are singing!");
```

다음 표는 \와 함께 사용되는 특수문자들을 보여주고 있습니다.

코드	출력
\'	single quote(')
\"	double quote(")
\&	ampersand(&)
\\	backslash(\)
\n	new line
\r	carriage return
\t	tab
\b	backspace
\f	form feed

10. 객체

● 객체지향 프로그래밍 (Object Oriented Programming)

자바스크립트는 객체지향 프로그래밍(OOP) 언어입니다. 모든 객체지향언어는 자신의 객체 선언하고, 이를 변수의 타입으로 사용할 수 있습니다. 이 장에서는 자바스크립트 내장(built-in) 객체들에 대하여 알아보도록 하겠습니다.

자바스크립트 객체는 속성(property)과 메소드(method)를 갖고 있습니다. 속성은 객체가 가지고 있는 값을 의미하며, 메소드는 객체로 할 수 있는 일을 의미합니다.

● Properties

속성은 객체와 연관된 값을 의미합니다. 다음은 String 객체의 length 속성을 사용한 예입니다. length는 문자열의 길이를 반환합니다. 실행 결과는 12입니다.

```
<script type="text/javascript">
var txt="Hello World!";
document.write(txt.length);
</script>
```

● Methods

메소드는 객체가 할 수 있는 해동을 의미합니다. 다음은 String 객체의 toUpperCase() 메소드를 사용한 예입니다. toUpperCase() 메소드는 영문 소문자를 대문자로 만들어 줍니다.

```
<script type="text/javascript">
var str="Hello world!";
document.write(str.toUpperCase());
</script>
```

10.1 String

String 객체는 문자열 작업에 사용됩니다. String 객체를 만드는 방법은 다음과 같이 new 키워드를 사용하여 만들 수 있고, 바로 변수에 문자열 값을 할당할 수도 있습니다.

```
var myStr1 = new String("문자열");
var myStr2 = "문자열";
```

String 객체의 자주 사용되는 속성과 메소드들을 예를 통하여 알아보겠습니다.

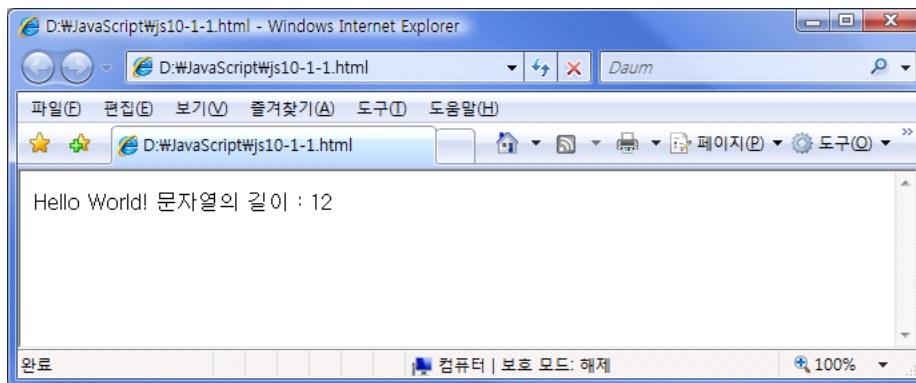
- 예제 코드 (length)

```

1:  <!-- js10-1-1.html -->
2:
3:  <html>
4:  <body>
5:
6:  <script type="text/javascript">
7:
8:  var txt="Hello World!";
9:  document.write(txt + "      길이 : " + txt.length);
10:
11: </script>
12:
13: </body>
14: </html>

```

- 실행 결과



- 예제 코드 (스타일 지정)

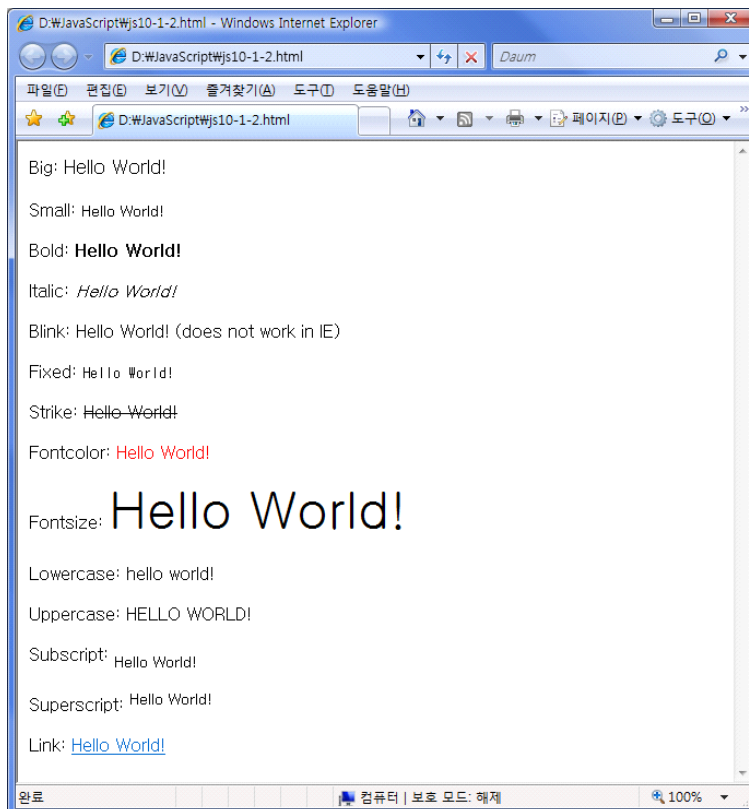
```

1:  <!-- js10-1-2.html -->
2:
3:  <html>
4:  <body>
5:
6:  <script type="text/javascript">
7:
8:  var txt="Hello World!";
9:
10: document.write("<p>Big: " + txt.big() + "</p>");
11: document.write("<p>Small: " + txt.small() + "</p>");
12:
13: document.write("<p>Bold: " + txt.bold() + "</p>");

```

```
14: document.write("<p>Italic: " + txt.italics() + "</p>");
15:
16: document.write("<p>Blink: " + txt.blink() + " (does not work in IE)</p>");
17: document.write("<p>Fixed: " + txt.fixed() + "</p>");
18: document.write("<p>Strike: " + txt.strike() + "</p>");
19:
20: document.write("<p>Fontcolor: " + txt.fontcolor("Red") + "</p>");
21: document.write("<p>Fontsize: " + txt.fontsize(16) + "</p>");
22:
23: document.write("<p>Lowercase: " + txt.toLowerCase() + "</p>");
24: document.write("<p>Uppercase: " + txt.toUpperCase() + "</p>");
25:
26: document.write("<p>Subscript: " + txt.sub() + "</p>");
27: document.write("<p>Superscript: " + txt.sup() + "</p>");
28:
29: document.write("<p>Link: " + txt.link("http://jvaspecialist.co.kr")+ "</p>");
30: </script>
31:
32: </body>
33: </html>
```

● 실행 결과



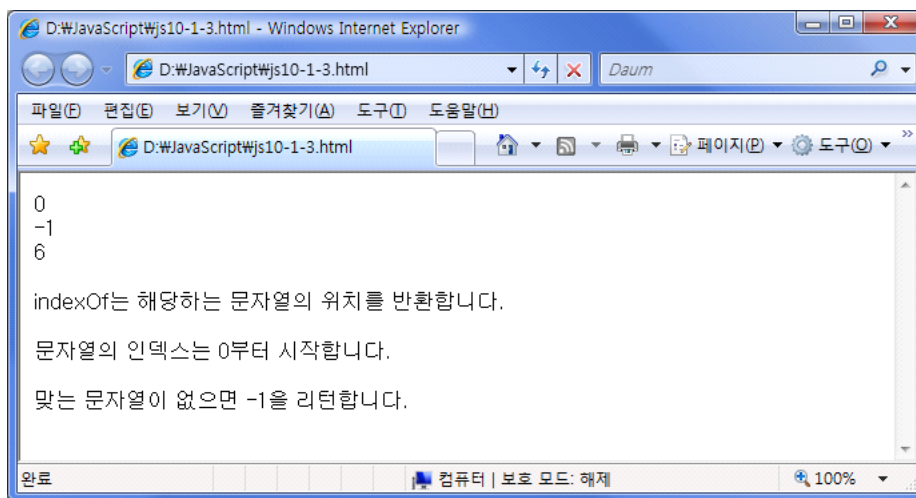
- 예제 코드 (indexOf ())

```

1:  <!-- js10-1-3.html -->
2:
3:  <html>
4:  <body>
5:
6:  <script type="text/javascript">
7:
8:  var str="Hello world!";
9:  document.write(str.indexOf("Hello") + "<br />");
10: document.write(str.indexOf("World") + "<br />");
11: document.write(str.indexOf("world"));
12:
13: </script>
14:
15: <p>indexOf 해당하는 문자열의 위치를 반환합니다.</p>
16: <p>문자열의 인덱스는 0부터 시작합니다.</p>
17: <p>맞는 문자열이 없으면 -1을 리턴합니다.</p>
18:
19: </body>
20: </html>

```

- 실행 결과



- 예제 코드 (replace ())

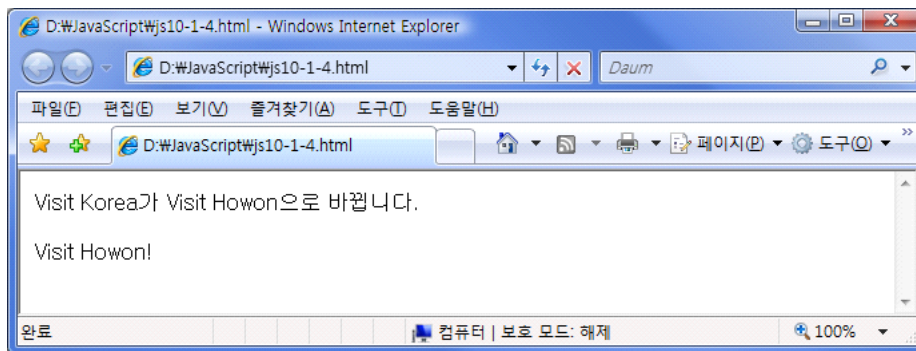
```

1:  <!-- js10-1-4.html -->
2:
3:  <html>
4:  <body>

```

```
5: <p>Visit Korea Visit Howon으로 바뀝니다.</p>
6:
7: <script type="text/javascript">
8:
9: var str="Visit Korea!";
10: document.write(str.replace(/Korea/, "Howon"));
11:
12: </script>
13: </body>
14: </html>
```

● 실행 결과



10.2 Date

Date 객체는 날짜와 시간을 다루기 위해 사용됩니다. Date 객체는 자동으로 현재의 날짜와 시간을 초기값으로 갖습니다.

```
var myDate=new Date()
```

날짜를 변경시키기 위해서는 setFullYear() 메소드를 사용합니다. 만일 2010년 1월 14일로 날짜를 변경시키기 위해서는 다음과 같이 사용합니다. 주의할 점은 월의 경우에는 1월이 0부터 시작한다는 것입니다.

```
var myDate=new Date();
myDate.setFullYear(2010,0,14);
```

● 예제 코드

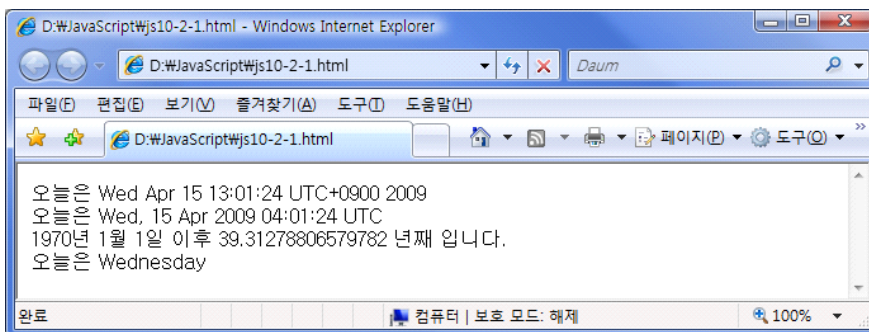
```
1: <!-- js10-2-1.html -->
2:
```

```

3: <html>
4: <body>
5:
6: <script type="text/javascript">
7:
8:   var minutes = 1000*60;
9:   var hours = minutes*60;
10:  var days = hours*24;
11:  var years = days*365;
12:
13:  var d = new Date();
14:
15:  var t = d.getTime();
16:  var y = t/years;
17:
18:  document.write("      " + d);
19:  document.write("<br>");
20:  document.write ("오늘은 " + d.toUTCString());
21:  document.write("<br>");
22:  document.write("1970년 1월 1일 이후 " + y + " 년째 입니다.");
23:  document.write("<br>");
24:
25:  var weekday=new Array(7);
26:  weekday[0]="Sunday";
27:  weekday[1]="Monday";
28:  weekday[2]="Tuesday";
29:  weekday[3]="Wednesday";
30:  weekday[4]="Thursday";
31:  weekday[5]="Friday";
32:  weekday[6]="Saturday";
33:
34:  document.write("오늘은 " + weekday[d.getDay()]);
35: </script>
36:
37: </body>
38: </html>

```

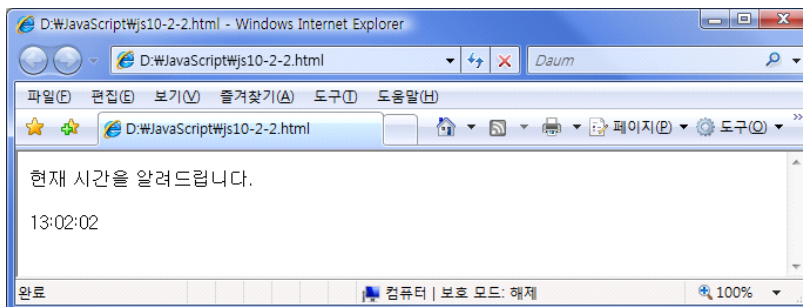
● 실행 결과



- 예제 코드 (시계)

```
1: <!-- js10-2-2.html -->
2:
3: <html>
4: <head>
5: <script type="text/javascript">
6:   function startTime()
7:   {
8:     var today=new Date();
9:     var h=today.getHours();
10:    var m=today.getMinutes();
11:    var s=today.getSeconds();
12:
13:    //      0보다 작으면 앞에 0을 붙여줍니다.
14:    m=checkTime(m);
15:    s=checkTime(s);
16:    document.getElementById('txt').innerHTML=h+":"+m+":"+s;
17:    t=setTimeout('startTime()',500);
18:  }
19:
20:  function checkTime(i)
21:  {
22:    if (i<10)
23:    {
24:      i="0" + i;
25:    }
26:    return i;
27:  }
28: </script>
29: </head>
30:
31: <body onload="startTime()">
32: <p>현재 시간을 알려드립니다.</p>
33: <div id="txt"></div>
34:
35: </body>
36: </html>
```

- 실행 결과



10.3 Array

Array 객체는 하나의 변수에 여러 개의 값을 저장하기 위해 사용합니다.

```
var myCars=new Array();
```

배열에 값을 저장하는 방법은 두 가지가 있습니다. 먼저 배열을 선언한 다음 값을 하나씩 넣는 방법이고, 또 하나는 배열을 생성할 때 값을 지정하는 방법입니다.

```
var myFruits=new Array();
myFruits[0]="사과";
myFruits[1]="바나나";
myFruits[2]="딸기";
```

```
var myFruits=new Array(3);
myFruits[0]="사과";
myFruits[1]="바나나";
myFruits[2]="딸기";
```

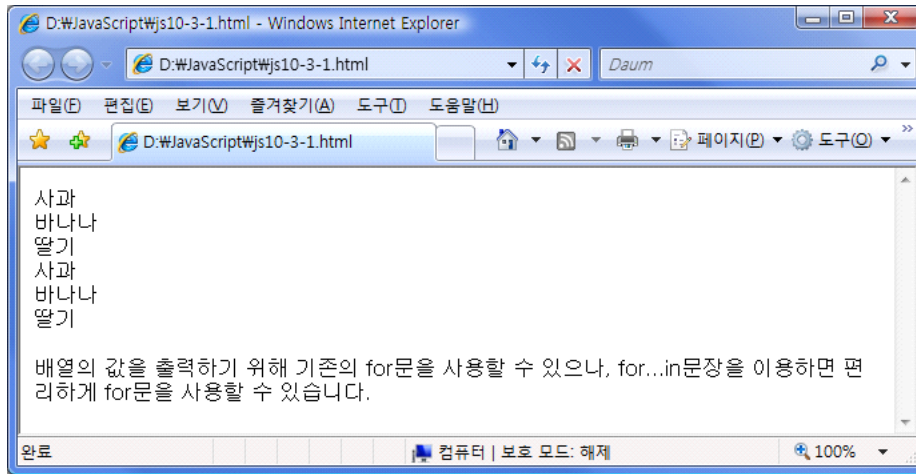
```
var myFruits=new Array("사과","바나나","딸기");
```

● 예제 코드

```
1:  <!-- js10-3-1.html -->
2:
3:  <html>
4:  <body>
5:  <script type="text/javascript">
6:    var x;
7:
8:    var myFruits=new Array(3);
9:    myFruits[0]="  ";
10:   myFruits[1]="바나나";
11:   myFruits[2]="딸기";
12:
13:   for (i=0;i<myFruits.length;i++)
14:   {
15:     document.write(myFruits[i] + "<br />");
16:   }
17:   for (x in myFruits)
18:   {
19:     document.write(myFruits[x] + "<br />");
20:   }
21: </script>
```

```
22: </body>
23: <p>      값을 출력하기 위해 기존의 for문을 사용할 수 있으나, for...in문장을 이용하면 편리하게
        for문을 사용할 수 있습니다.</p>
24: </html>
```

● 실행 결과



● 예제 코드 (concat(), join(), sort())

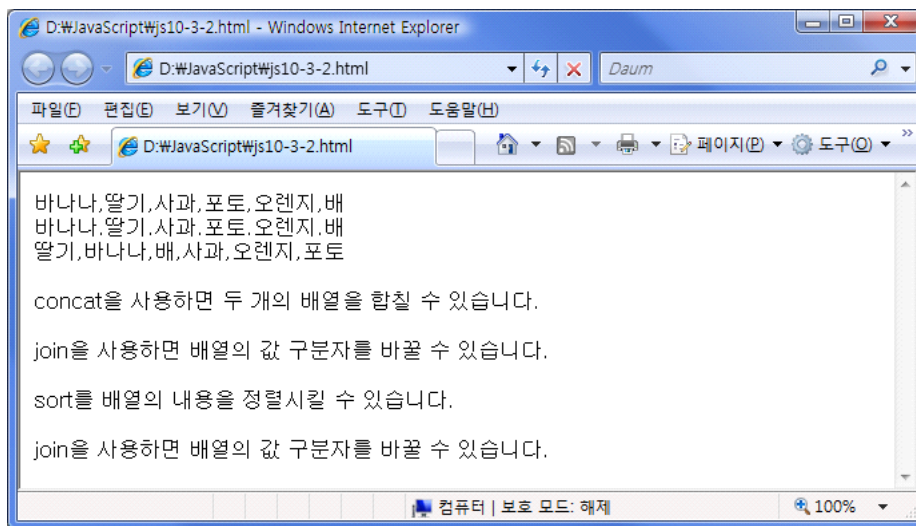
```
1: <!-- js10-3-2.html -->
2:
3: <html>
4: <body>
5:
6: <script type="text/javascript">
7:
8:   var arr = new Array(3);
9:   arr[0] = "바나나";
10:  arr[1] = "딸기";
11:  arr[2] = "사과";
12:
13:  var arr2 = new Array(3);
14:  arr2[0] = "포도";
15:  arr2[1] = "오렌지";
16:  arr2[2] = "배";
17:
18:  arr = arr.concat(arr2)
19:
20:  document.write(arr);
21:  document.write("<br>");
22:
23:  document.write(arr.join("."));
```

```

24: document.write("<br>");
25:
26: document.write(arr.sort());
27: document.write("<br>");
28: </script>
29:
30: <p>concat 사용하면 두 개의 배열을 합칠 수 있습니다.</p>
31: <p>join을 사용하면 배열의 값 구분자를 바꿀 수 있습니다.</p>
32: <p>sort를 배열의 내용을 정렬시킬 수 있습니다.</p>
33: <p>join을 사용하면 배열의 값 구분자를 바꿀 수 있습니다.</p>
34:
35: </body>
36: </html>

```

● 실행 결과



10.4 Boolean

Boolean 객체는 true 또는 false 값을 나타낼 수 있습니다.

```
var myBoolean=new Boolean();
```

Boolean 객체는 초기 값을 갖지 않거나 초기 값을 갖더라도 0, -0, null, "", false, undefined, NaN 등의 값을 초기 값으로 가지면 false로 설정됩니다. 그렇지 않은 경우에는 true로 설정됩니다.

다음은 false로 초기화 되는 Boolean 객체들의 선언입니다.

- var myBoolean=new Boolean();
- var myBoolean=new Boolean(0);
- var myBoolean=new Boolean(null);
- var myBoolean=new Boolean("");
- var myBoolean=new Boolean(false);
- var myBoolean=new Boolean(NaN);

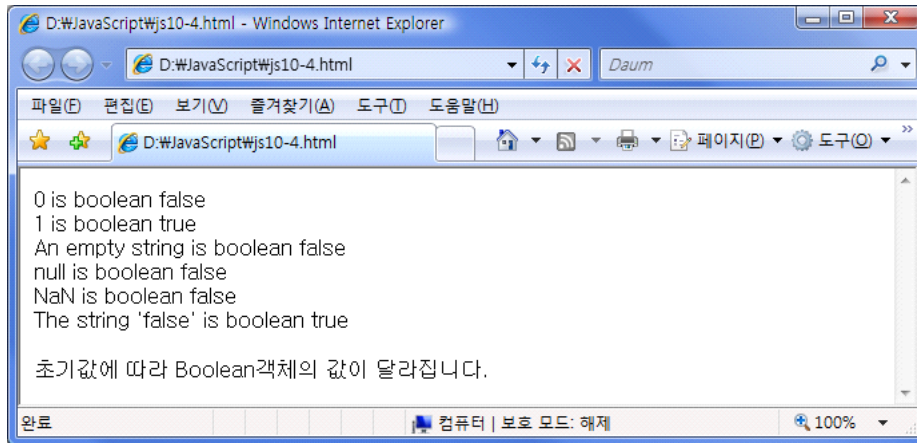
다음은 true로 초기화 되는 Boolean 객체들의 선언입니다.

- var myBoolean=new Boolean(true);
- var myBoolean=new Boolean("true");
- var myBoolean=new Boolean("false");
- var myBoolean=new Boolean("Richard");

● 예제 코드

```
1:  <!-- js10-4.html -->
2:
3:  <html>
4:  <body>
5:
6:  <script type="text/javascript">
7:  var b1=new Boolean( 0);
8:  var b2=new Boolean(1);
9:  var b3=new Boolean("");
10: var b4=new Boolean(null);
11: var b5=new Boolean(NaN);
12: var b6=new Boolean("false");
13:
14: document.write("0 is boolean "+ b1 + "<br />");
15: document.write("1 is boolean "+ b2 + "<br />");
16: document.write("An empty string is boolean "+ b3 + " <br />");
17: document.write("null is boolean "+ b4+ " <br />");
18: document.write("NaN is boolean "+ b5 + "<br />");
19: document.write("The string 'false' is boolean "+ b6 + "<br />");
20: </script>
21:
22: <p>        따라 Boolean객체의 값이 달라집니다.</p>
23:
24: </body>
25: </html>
```

- 실행 결과



10.5 Math

Math 객체는 여러 가지 수학적 상수들과 메소드들을 갖고 있기 때문에 수학적 연산을 쉽게 수행하도록 도와줍니다.

```
var pi_value=Math.PI;
var sqrt_value=Math.sqrt(16);
```

Math 객체는 생성자가 없습니다. 그러므로 new 키워드를 통하여 객체를 생성할 필요가 없습니다. Math 클래스의 모든 속성들과 메소드들은 Math 객체 이름을 통하여 호출할 수 있습니다.

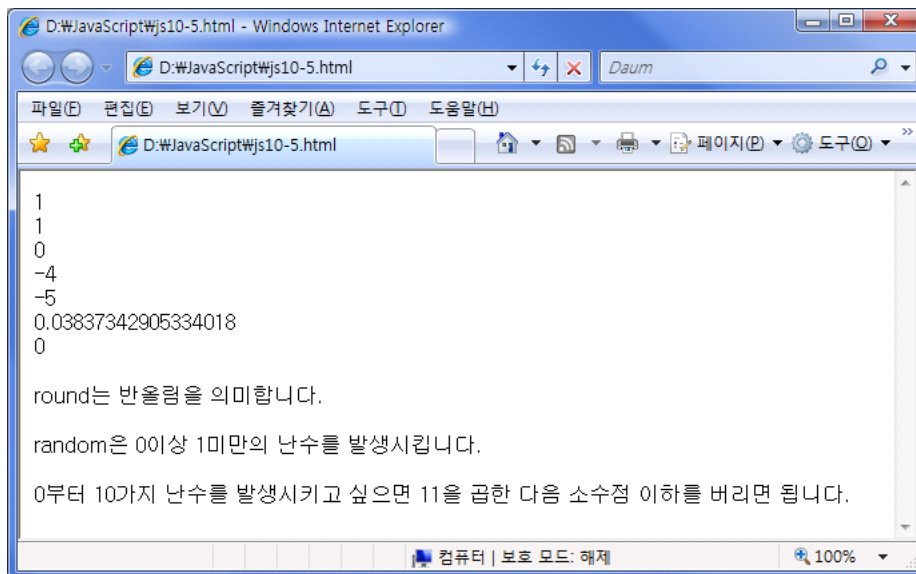
Math객체는 여러 가지 상수 값들이 선언되어 있습니다.

상수	값
Math.E	E(약 2.718)
Math.PI	PI(약 3.14159)
Math.SQRT2	2의 제곱근(약 1.414)
Math.SQRT1_2	1/2의 제곱근(약 0.707)
Math.LN2	2의 자연로그(약 0.693)
Math.LN10	10의 자연로그(약 2.302)
Math.LOG2E	E의 로그(base 2)(약 1.442)
Math.LOG10E	E의 로그(base 10)(약 0.434)

● 예제 코드

```
1: <!-- js10-5.html -->
2:
3: <html>
4: <body>
5:
6: <script type="text/javascript">
7:
8: document.write(Math.round(0.60) + "<br />");
9: document.write(Math.round(0.50) + "<br />");
10: document.write(Math.round(0.49) + "<br />");
11: document.write(Math.round(-4.40) + "<br />");
12: document.write(Math.round(-4.60) + "<br />");
13:
14: num = Math.random();
15: document.write(num+ "<br />");
16:
17: document.write(Math.floor(num*11));
18:
19: </script>
20: <p>round 반올림을 의미합니다.</p>
21: <p>random은 0이상 1미만의 난수를 발생시킵니다.</p>
22: <p>0부터 10가지 난수를 발생시키고 싶으면 11을 곱한 다음 소수점 이하를 버리면 됩니다.</p>
23:
24: </body>
25: </html>
```

● 실행 결과



10.6 RegExp

RegExp(regular expression) 문자열 내에서 찾고 싶은 패턴 문자열을 정의할 때 사용합니다.

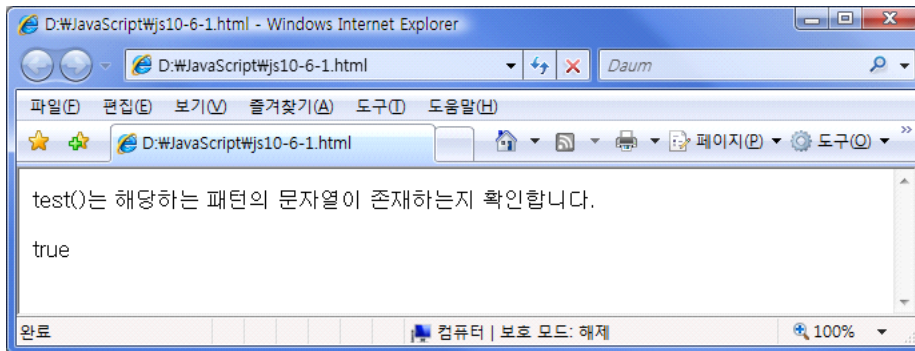
```
var patt1=new RegExp("e");
```

RegExp 객체는 test(), exec(), compile() 3개의 메소드를 가지고 있습니다.

● 예제 코드(test())

```
65: <!-- js10-6-1.html -->
66:
67: <html>
68: <body>
69: <p>test() 해당하는 패턴의 문자열이 존재하는지 확인합니다.</p>
70:
71: <script type="text/javascript">
72: var patt1=new RegExp("co");
73:
74: document.write(patt1.test("Dok-Do belongs to my country, Republic of Korea"));
75: </script>
76:
77: </body>
78: </html>
```

● 실행 결과

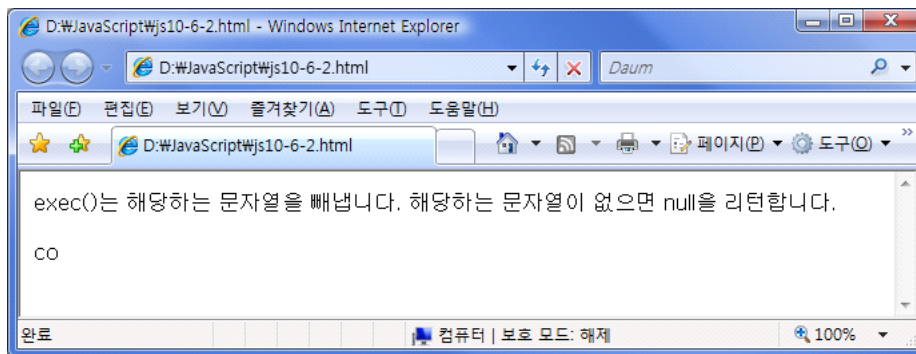


● 예제 코드(exec())

```
1: <!-- js10-6-2.html -->
2:
3: <html>
4: <body>
5: <p>exec()는 해당하는 문자열을 빼냅니다. 해당하는 문자열이 없으면 null을 리턴합니다.</p>
6:
```

```
7: <script type="text/javascript">
8: var patt1=new RegExp("co");
9:
10: document.write(patt1.exec("Dok-Do belongs to my country, Republic of Korea"));
11: </script>
12:
13: </body>
14: </html>
```

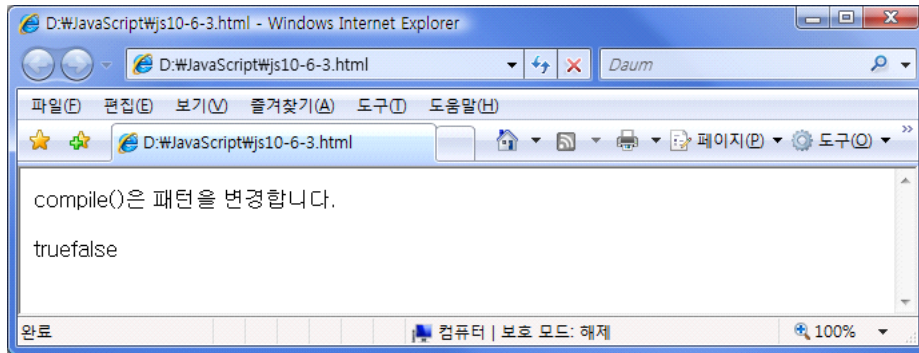
● 실행 결과



● 예제 코드 (compile())

```
1: <!-- js10-6-3.html -->
2:
3: <html>
4: <body>
5: <p>compile() 패턴을 변경합니다.</p>
6:
7: <script type="text/javascript">
8: var patt1=new RegExp("co");
9: document.write(patt1.test("Dok-Do belongs to my country, Republic of Korea"));
10:
11: patt1.compile("d");
12:
13: document.write(patt1.test("Dok-Do belongs to my country, Republic of Korea"));
14: </script>
15:
16: </body>
17: </html>
```

● 실행 결과



10.7 HTML DOM

앞에서 언급된 JavaScript 내장객체들 외에 HTML DOM 객체들도 자바스크립트에서 사용할 수 있습니다.

10.7.1 JavaScript

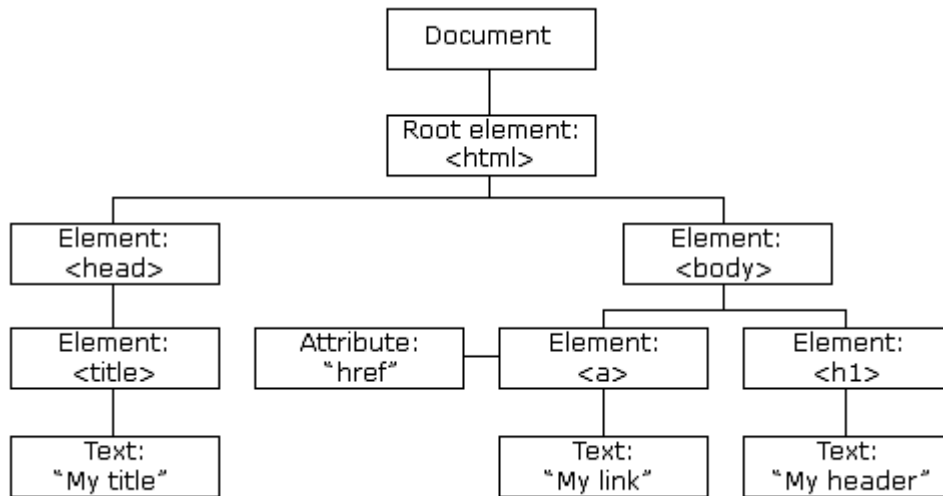
다음은 자바스크립트 내장 객체들입니다. 각각의 속성, 메소드, 그리고 이벤트들을 알아두면 유용하게 사용할 수 있습니다.

객체	설명
Window	자바스크립트 객체 구조 중 최상위 객체임, 브라우저 윈도우를 표현함.
Navigator	클라이언트 브라우저에 관한 정보를 포함
Screen	클라이언트에 디스플레이 되는 스크린에 대한 정보
History	브라우저 윈도우에서 방문한 URL정보
Location	현재 URL에 대한 정보

10.7.2 The HTML DOM

HTML DOM(Document Object Model for HTML)은 HTML객체들을 표현하기 위한 W3C의 표준입니다. 모든 HTML 요소들(텍스트와 속성 포함)은 DOM객체를 통해서 접근할 수 있습니다. 콘텐츠들은 수정되거나, 삭제되거나 또는 새로운 요소가 추가될 수도 있습니다.

DOM은 HTML 문서를 트리 구조로 표현합니다. 트리구조의 각 노드들은 태그이름, 속성, 그리고 텍스트가 됩니다.



HTML DOM은 플랫폼과 언어에 독립적입니다. 즉, JavaScript에서도 사용할 수 있으며, VBScript에서도 동일하게 사용할 수 있습니다. 다음은 HTML DOM을 나열한 것입니다.

객체	설명
Document	HTML 문서
Anchor	<a>
Area	image-map에서 <area>
Base	<base>
Body	<body>
Button	<button>
Event	event의 상태
Form	<form>
Frame	<frame>
Frameset	<frameset>
Iframe	<iframe>
Image	
Input button	HTML 폼 안의 button
Input checkbox	HTML 폼 안의 checkbox
Input file	HTML 폼 안의 fileupload
Input hidden	HTML 폼 안의 hidden field

Input password	HTML 폼 안의 password field
Input radio	HTML 폼 안의 radio button
Input reset	HTML 폼 안의 reset button
Input submit	HTML 폼 안의 submit button
Input text	HTML 폼 안의 text-input field
Link	<link>
Meta	<meta>
Option	<option>
Select	HTML 폼 안의 selection list
Style	style 문장
Table	<table>
TableData	<td>
TableRow	<tr>
Textarea	<textarea>

JavaScript Quiz

1. 사용하기 위한 태그는?
☐ a) <scripting>
☐ b) <javascript>
☐ c) <script>
☐ d) <js>

2. "Hello World"나타내기 위한 자바스크립트 구문은?
☐ a) document.write("Hello World")
☐ b) ("Hello World")
☐ c) "Hello World"
☐ d) response.write("Hello World")

3. 자바스크립트를 삽입하는 바른 위치는?
☐ a) <head> 태그 또는 <body> 태그
☐ b) <body> 태그
☐ c) <head> 태그
☐ d) <html> 태그 뒤

4. 외부 자바스크립트 파일 "xxx.js"을 참조하기 위한 코드 중 바른 것은?
☐ a) <script link="xxx.js">
☐ b) <script src="xxx.js">
☐ c) <script href="xxx.js">
☐ d) <script name="xxx.js">

5. 외부 파일로 작성되어지는 자바스크립트는 <script> 태그를 포함합니다.
☐ True
☐ False

6. "Hello World"를 경고 창으로 나타내는 코드는?
☐ a) alertBox("Hello World")
☐ b) msgBox("Hello World")
☐ c) alertBox="Hello World"
☐ d) alert("Hello World")

7. 함수를 작성하는 방법 중 옳은 것은?

- ☐ a) function myFunction()
- ☐ b) fonction.myFunction()
- ☐ c) function=myFunction()
- ☐ d) function:myFunction()

8. "myFunction" 함수를 호출하는 방법은?

- ☐ a) myFunction.exec()
- ☐ b) myFunction()
- ☐ c) call myFunction()
- ☐ d) call function myFunction

9. 만일 "i" 값이 5와 같으면 어떤 문장을 실행하기 위한 조건문을 바르게 작성한 것은?

- ☐ a) if i=5 then
- ☐ b) if i=5
- ☐ c) if (i==5)
- ☐ d) if i==5 then

10. "i" 값이 5와 같지 않으면 어떤 문장을 실행시키기 위한 문장 중 바르게 작성한 것은?

- ☐ a) if (i != 5)
- ☐ b) if I != 5 then
- ☐ c) if (i <> 5)
- ☐ d) if <>5

11. 자바스크립트에서 반복문의 개수와 종류?

- ☐ a) 1개. "for" 문
- ☐ b) 2개. "for" 문, "while" 문
- ☐ c) 3개. "for" 문, "while" 문, "switch" 문
- ☐ d) 4개. "for" 문, "while" 문, "do...while" 문, "loop...until" 문

12. "for" 문이 바르게 작성된 것은?

- ☐ a) for (i = 0; i <= 5)
- ☐ b) for i = 1 to 5
- ☐ c) for (i = 0; i <= 5; i++)
- ☐ d) for (i <= 5; i++)

13. 자바스크립트 주석을 바르게 작성한 것은?

- ☐ a) #This is a comment

- ☐ b) `//This is a comment`
- ☐ c) `<!--This is a comment-->`
- ☐ d) `'This is a comment`

14. 라인에 걸쳐서 주석을 넣기 위한 방법은?

- ☐ a) `//This comment has
more than one line//`
- ☐ b) `/*This comment has
more than one line*/`
- ☐ c) `<!--This comment has
more than one line-->`
- ☐ d) `#This comment has
more than one line#`

15. 자바스크립트 배열을 바르게 사용한 것은?

- ☐ a) `var txt = new Array="월","화","수"`
- ☐ b) `var txt = new Array(1:"월",2:"화",3:"수")`
- ☐ c) `var txt = new Array:1=("월")2=("화")3=("수")`
- ☐ d) `var txt = new Array("월","화","수")`

16. 숫자 3.75를 가장 가까운 정수로 만들기 위한 방법은?

- ☐ a) `Math.round(3.75)`
- ☐ b) `rnd(3.75)`
- ☐ c) `Math.rnd(3.75)`
- ☐ d) `round(3.75)`

17. 숫자 3과 7중 큰 수를 찾기 위한 방법은?

- ☐ a) `Math.max(3,7)`
- ☐ b) `top(3,7)`
- ☐ c) `ceil(3,7)`
- ☐ d) `Math.ceil(3,7)`

18. "window2"라고 불리는 새로운 윈도우 창을 열기 위한 방법은?

- ☐ a) `window.open("http://www.javaspecialist.co.kr","window2")`
- ☐ b) `open.new("http://www.javaspecialist.co.kr","window2")`
- ☐ c) `new("http://www.javaspecialist.co.kr","window2")`
- ☐ d) `new.window("http://www.javaspecialist.co.kr","window2")`

19. 상태바에 메시지를 보이게 하기 위한 방법은?

- ☐ a) `statusbar = "put your message here"`
- ☐ b) `window.status("put your message here")`
- ☐ c) `window.status = "put your message here"`
- ☐ d) `status("put your message here")`

20. 클라이언트 브라우저의 이름을 알기 위한 방법은?

- ☐ a) `client.navName`
- ☐ b) `navigator.appName`
- ☐ c) `browser.name`
- ☐ d) `explorer.name`

