



Chapter09

JavaScript 구구단표 만들기

HTML & JavaScript

1. 주석

2. 조건문

3. 반복문

4. 함수

5. 구구단표 만들기

형식 : <BODY 속성=값 속성=값.....> 내용 </BODY>

속성 :

TOPMARGIN, LEFTMARGIN, BACKGROUND, BGCOLOR, TEXT, LINK, VLINK, ALINK

예제 :

```
<BODY BGCOLOR="yellow" TEXT="blue" LINK="red"
      VLINK="green" ALINK="gray">
```

BODY 태그의 속성

```
</BODY>
```

❖ 주석

- 프로그래밍한 코드를 실행하는데 아무런 영향을 주지 않으며 소스 코드 설명 등을 첨가할 때 사용
- 만약 한 줄을 주석 처리하고자 할 때는 `'//'`를 사용하며 여러 줄을 주석 처리하고자 할 때는 `'/*', '*/'`를 사용하여 처음과 끝 부분에 입력

형식 :

```
// 한 줄 주석
```

```
/*
```

```
    여러 줄 주석
```

```
*/
```

예제 :

```
<SCRIPT LANGUAGE="JavaScript">
```

```
<!--
```

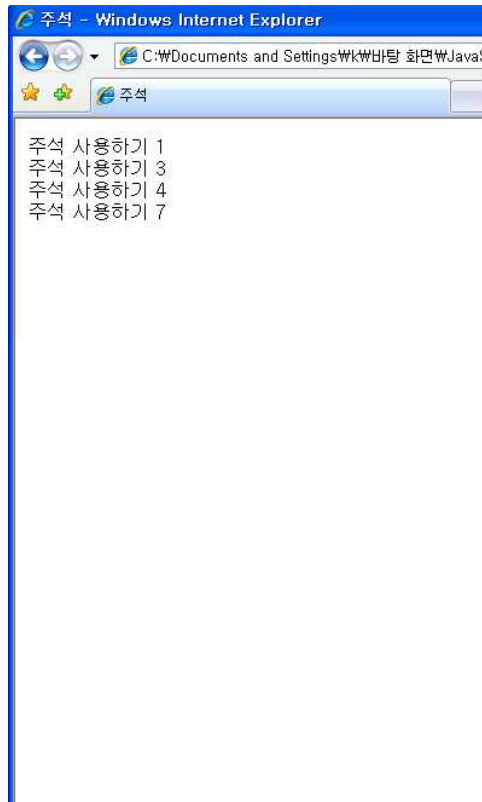
```
    /* 주석문을 사용하는 방법
```

```
    자바스크립트 주석문은 C언어와 같은 방법으로 사용함 */
```

```
    // document.write("주석 사용하기")
```

```
//-->
```

```
</SCRIPT>
```

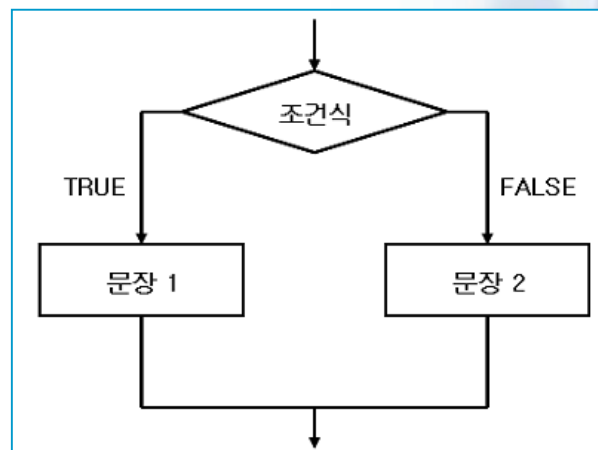


```
document.write("주석 사용하기 1 <BR>");  
//document.write("주석 사용하기 2 <BR>");  
document.write("주석 사용하기 3 <BR>");  
document.write("주석 사용하기 4 <BR>");  
/*document.write("주석 사용하기 5 <BR>");  
document.write("주석 사용하기 6 <BR>");*/  
document.write("주석 사용하기 7");
```

[그림 9-1] Comments의 실행결과

❖ if문

- 지정한 조건식의 참, 거짓 여부에 따라 해당 문장을 실행하는 조건문
- if와 else로 구성하고 else는 생략 가능
- 조건식이 참일 경우에는 if 다음에 오는 명령을 실행하며 거짓일 경우에는 else 다음에 오는 명령을 실행
- 만약 조건에 따라 여러 명령문을 실행하고자 할 때에는 해당하는 명령문에 '{ }'로 묶어 주면 됨



형식 :

// if문의 사용방법 1

```
if (조건식)
    문장
```

```
else
    문장
```

// if문의 사용방법 2

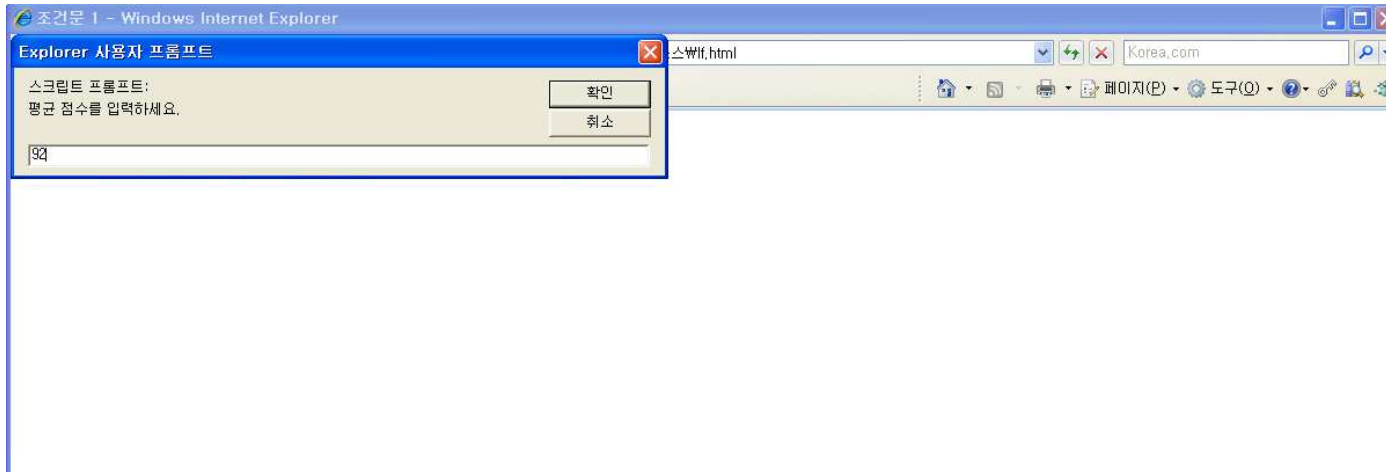
```
if (조건식)
    문장
```

// if문의 사용방법 3

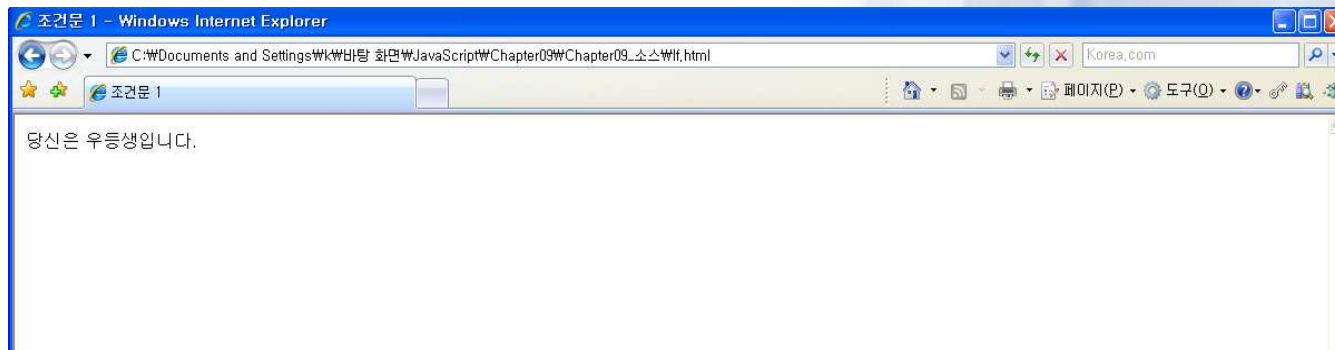
```
if (조건식) {
    문장
    문장
} else {
    문장
    문장
}
```


예제 :

```
<SCRIPT LANGUAGE="JavaScript">
<!--
    if(strLevel >= 90)
        document.write("당신은 우등생입니다.");
    else
        document.write("조금만 더욱 노력하세요.");
//-->
</SCRIPT>
```

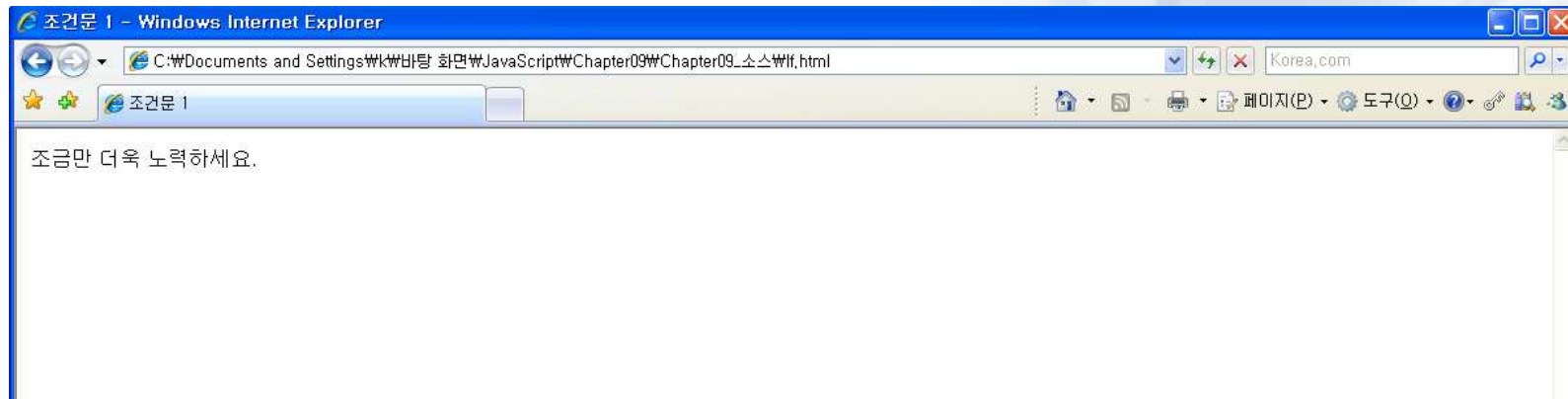


[그림 9-3] If의 실행결과



[그림 9-4] If의 실행결과

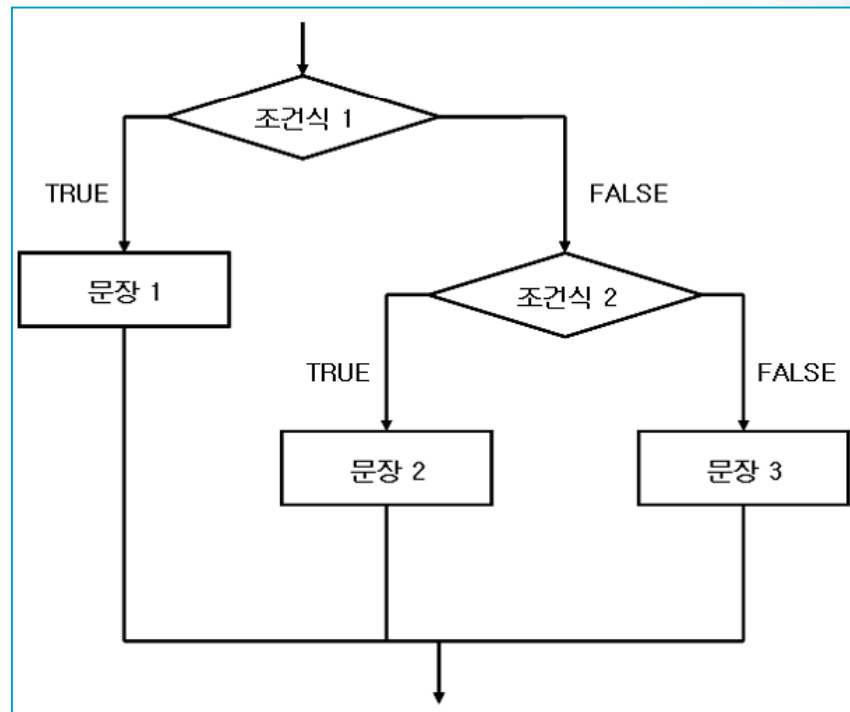
```
var strLevel = prompt("평균 점수를 입력하세요.", "0");  
  
if(strLevel >= 90)  
    document.write("당신은 우등생입니다.");  
else  
    document.write("조금만 더욱 노력하세요.");
```



[그림 9-5] If의 실행결과

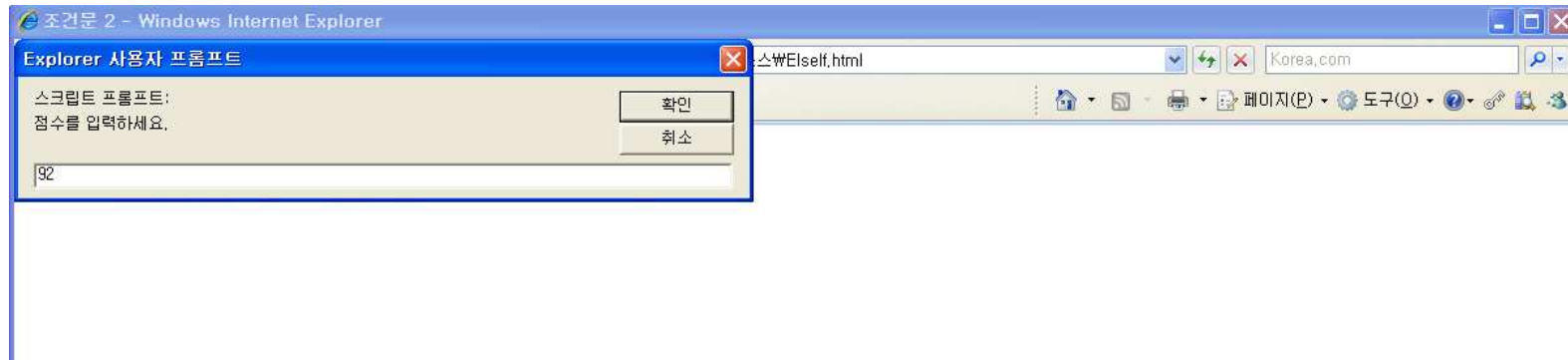
❖ else if문

- 조건식을 2개 이상 사용하여 3개 이상의 분기로 나누어 실행하고자 할 때 사용
- else if를 사용하여 추가적인 조건식을 입력



형식 :
if (조건식)
 문장
else if (조건식)
 문장
else
 문장

예제 :
<SCRIPT LANGUAGE="JavaScript">
<!--
 if(strLevel >= 90)
 document.write("당신의 학점은 A입니다.");
 else if(strLevel >= 80)
 document.write("당신의 학점은 B입니다.");
 else
 document.write("당신의 학점은 C입니다.");
//-->
</SCRIPT>

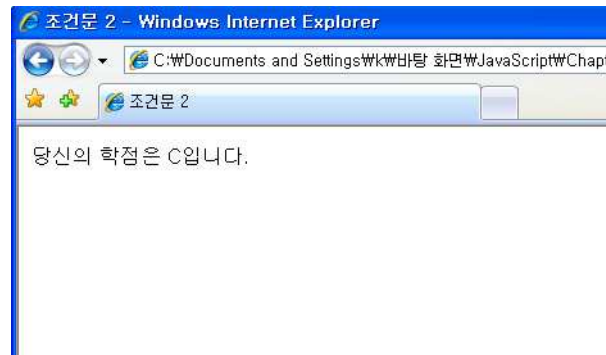


[그림 9-7] ElseIf의 실행결과



[그림 9-8] ElseIf의 실행결과

```
if(strLevel >= 90)
    document.write("당신의 학점은 A입니다.");
else if(strLevel >= 80)
    document.write("당신의 학점은 B입니다.");
else if(strLevel >= 70)
    document.write("당신의 학점은 C입니다.");
else if(strLevel >= 60)
    document.write("당신의 학점은 D입니다.");
else
    document.write("당신의 학점은 F입니다..");
```



[그림 9-9] ElseIf의 실행결과

❖ switch문

- 하나의 조건식만을 판단
- 그 결과 값을 **case**문의 값과 비교해 해당 문장을 실행하는 구조를 가지고 있음
- **default**는 **case**로 지정된 값 이외의 경우에 대한 명령을 실행할 경우 사용

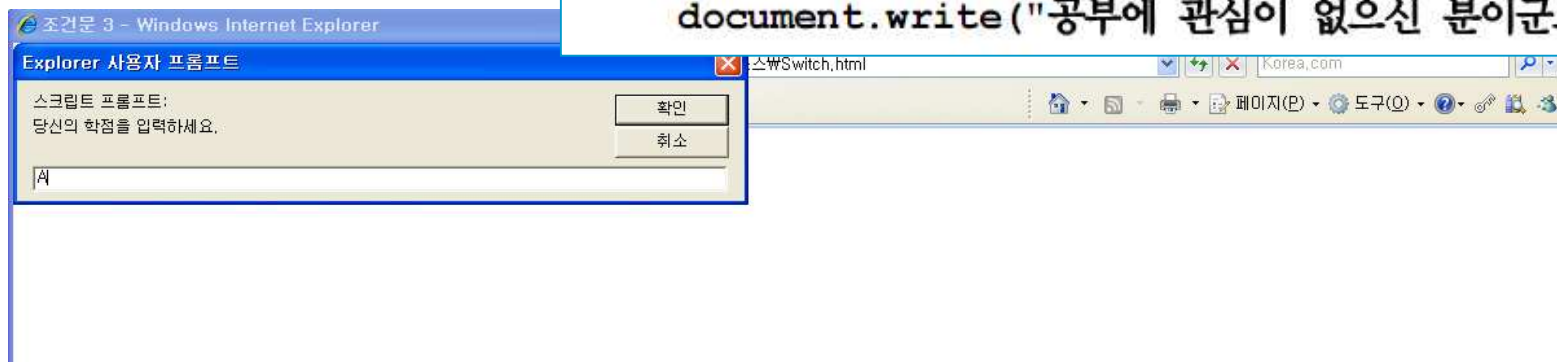
형식 :

```
switch (변수) {  
    case 값 1 :  
        문장  
        break;  
    case 값 2 :  
        문장  
        break;  
    default :  
        문장  
}
```

예제 :

```
switch(strLevel)  
{  
    case "A":  
        document.write("당신은  
                        우등생입니다.");  
        break;  
    case "B":  
        document.write("조금만 더  
                        노력하세요.");  
        break;  
    case "C":  
        document.write("더욱 많은 노력이  
                        필요합니다.");  
        break;  
    default:  
        document.write("공부에 관심이 없으신  
                        분이군요.");  
}
```

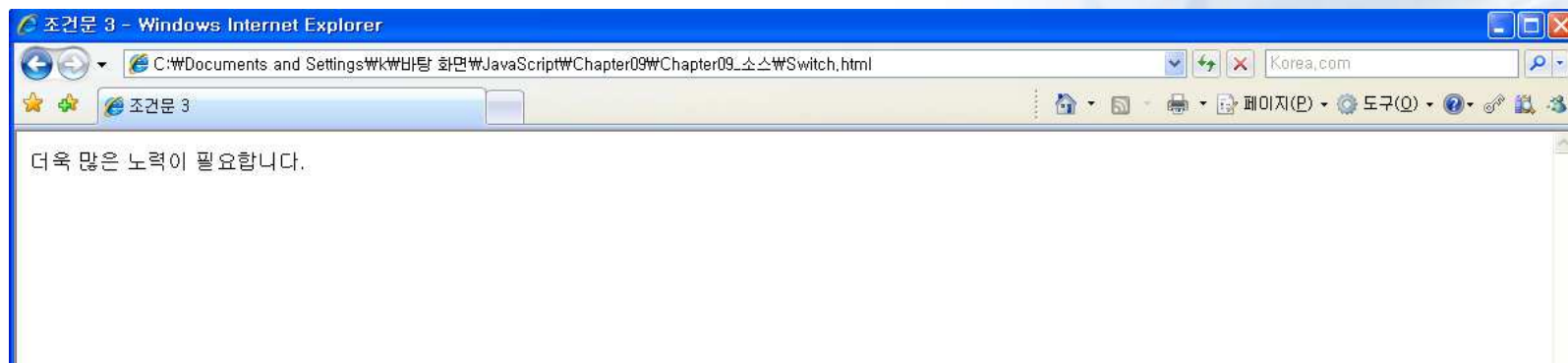
```
case "A":
    document.write("당신은 우등생입니다.");
break;
case "B":
    document.write("조금만 더 노력하세요.");
break;
case "C":
    document.write("더욱 많은 노력이 필요합니다.");
break;
default:
    document.write("공부에 관심이 없으신 분이군요.");
```



[그림 9-10] Switch의 실행결과



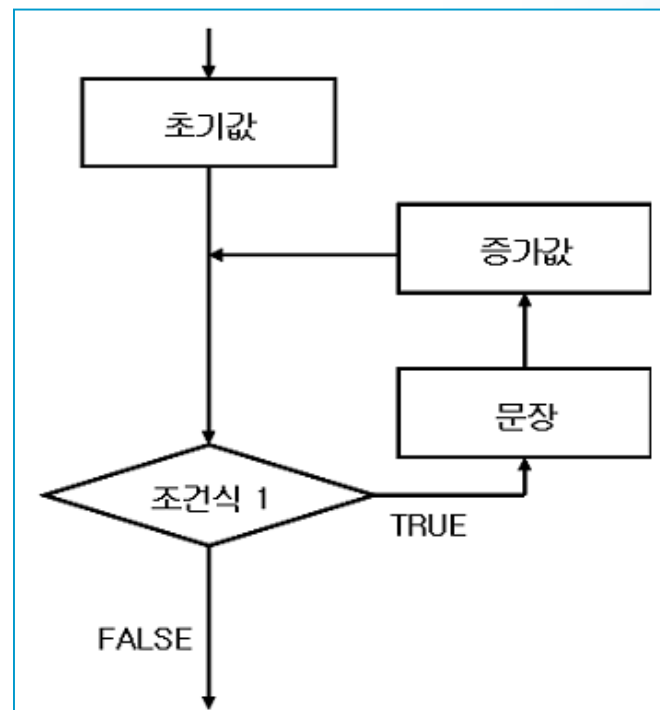
[그림 9-11] Switch의 실행결과



[그림 9-12] Switch의 실행결과

❖ for문

- 지정한 회수만큼 { } 안의 명령을 반복 수행
- 기본적으로 3개의 값을 지정하게 되는데, 그것은 초기값, 종료 조건, 증감 식으로 구성됨

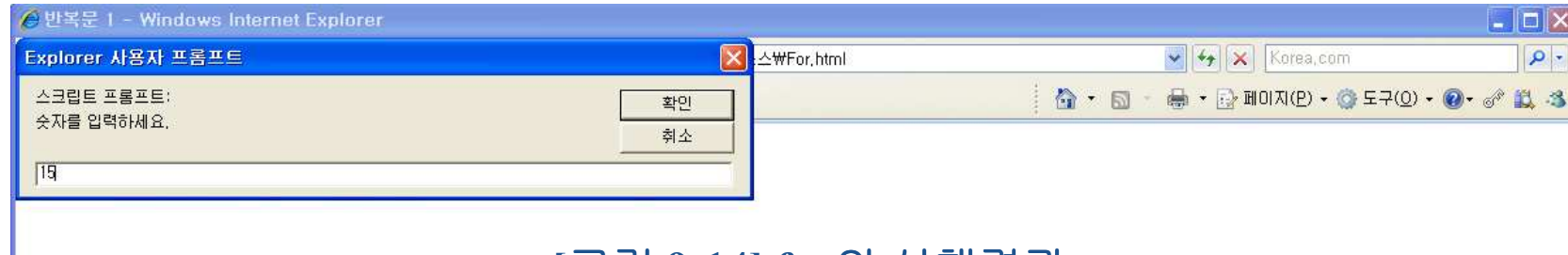


형식 :

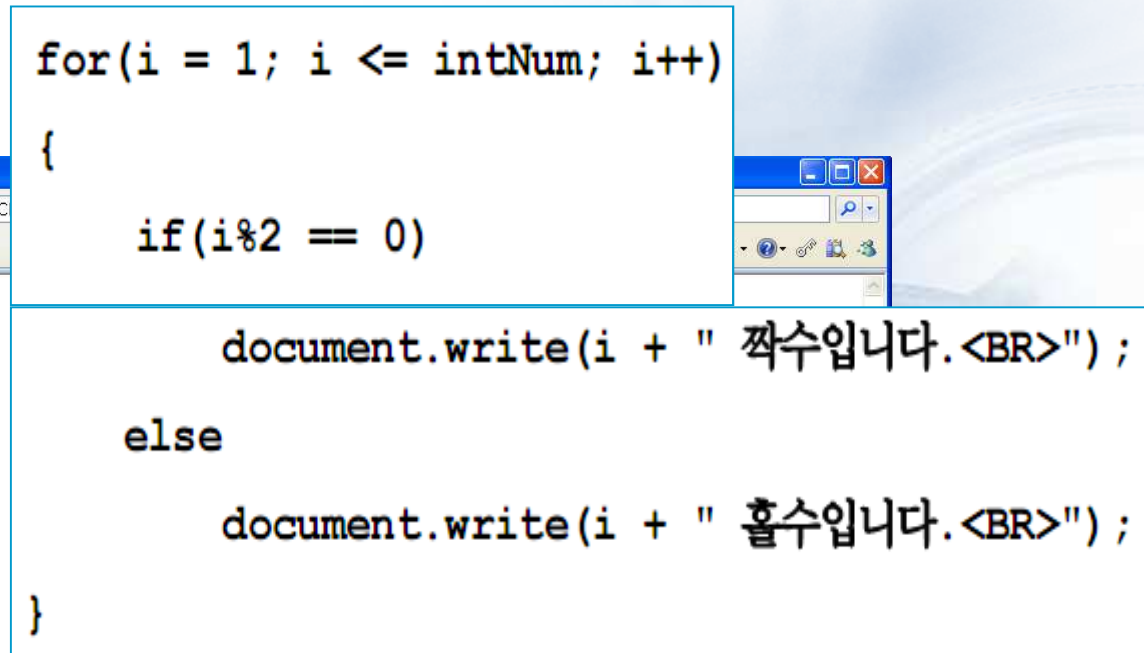
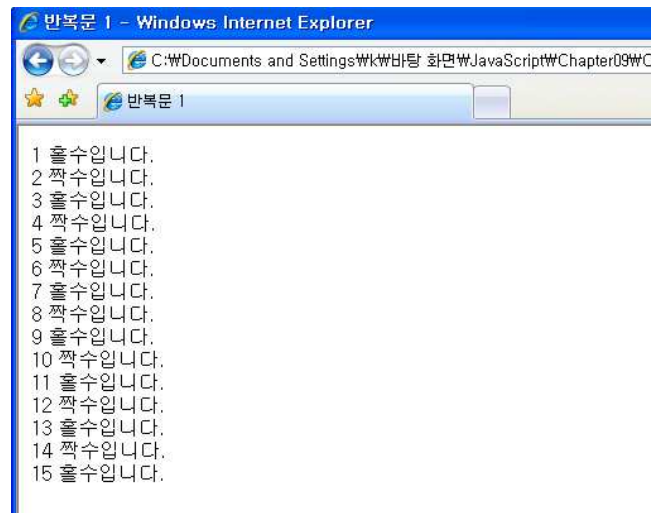
```
for (초기값; 최종값; 증감) {  
    문장;  
    문장;  
    .....  
}
```

예제 :

```
<SCRIPT LANGUAGE="JavaScript">  
<!--  
    for (var i = 1; i <= 10; i++){  
        document.write("회원님의 방문횟수는" + i + "입니다<BR>");  
    }  
//-->  
</SCRIPT>
```



[그림 9-14] for의 실행결과



[그림 9-15] for의 실행결과

❖ while문

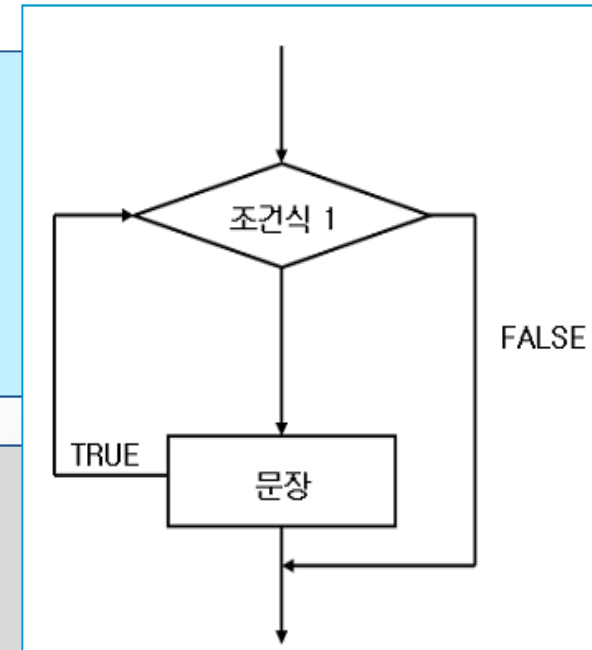
- 사용하는 이유는 for문과 비슷하지만 반복 조건의 지정하는 방법이 다름
- while문의 옆에 () 안에는 조건식을 입력하며 반복 실행되는 명령은 { }안에 넣음
- 조건식에 지정한 변수의 값의 증감식은 { }안에 넣음
- 조건식이 참인 동안에만 반복적으로 { }안의 명령을 실행
- 만약 증감식을 혹시나 넣지 않는다면 while문 안의 명령들이 무한반복 실행되어 무한 루프에 빠지게 됨

형식 :

```
while (조건식) {  
    문장;  
    증감식;  
}
```

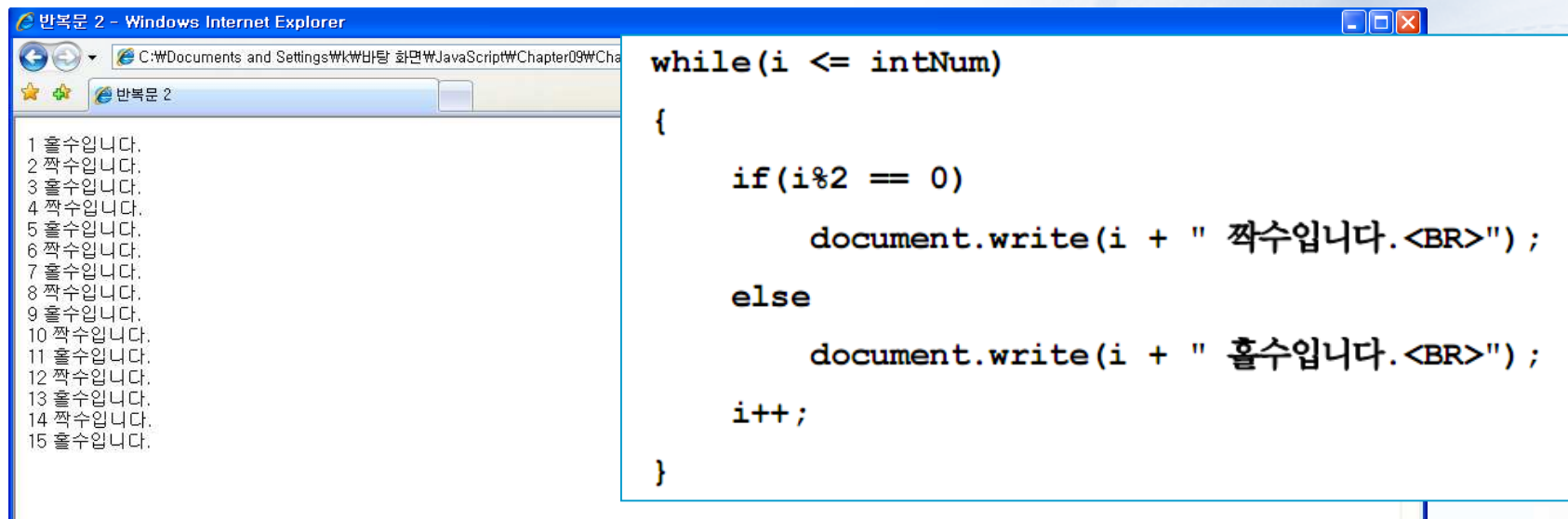
예제 :

```
<SCRIPT LANGUAGE="JavaScript">  
<!--  
    var intCount = 1;  
    while(intCount < 10){  
        document.write("회원님은" + intCount + "번째 방문입니다.<BR>");  
        intCount++;  
    }  
//-->
```





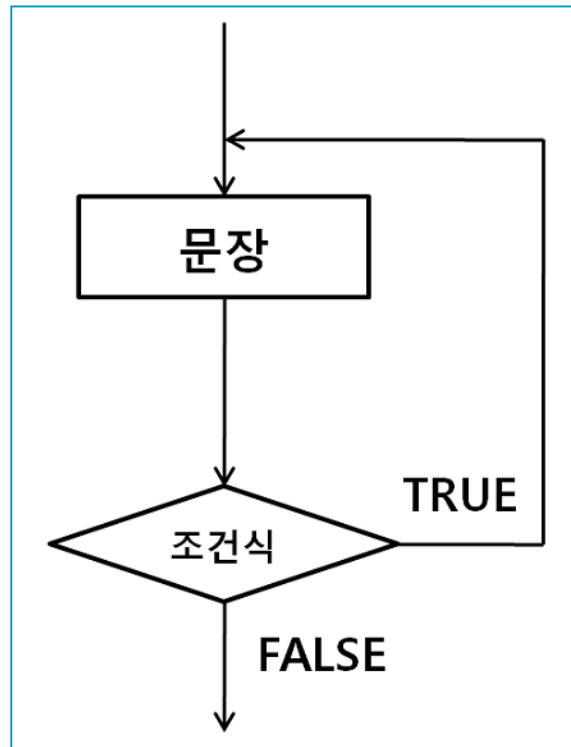
[그림 9-17] While의 실행결과



[그림 9-18] While의 실행결과

❖ do while문

- 명령문들을 먼저 실행한 후 조건식이 참인지 거짓인지 판단
- 명령문은 조건에 상관없이 최소 1번 실행

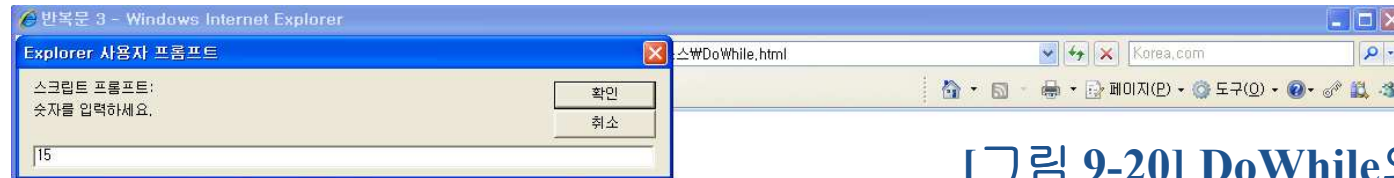


형식 :

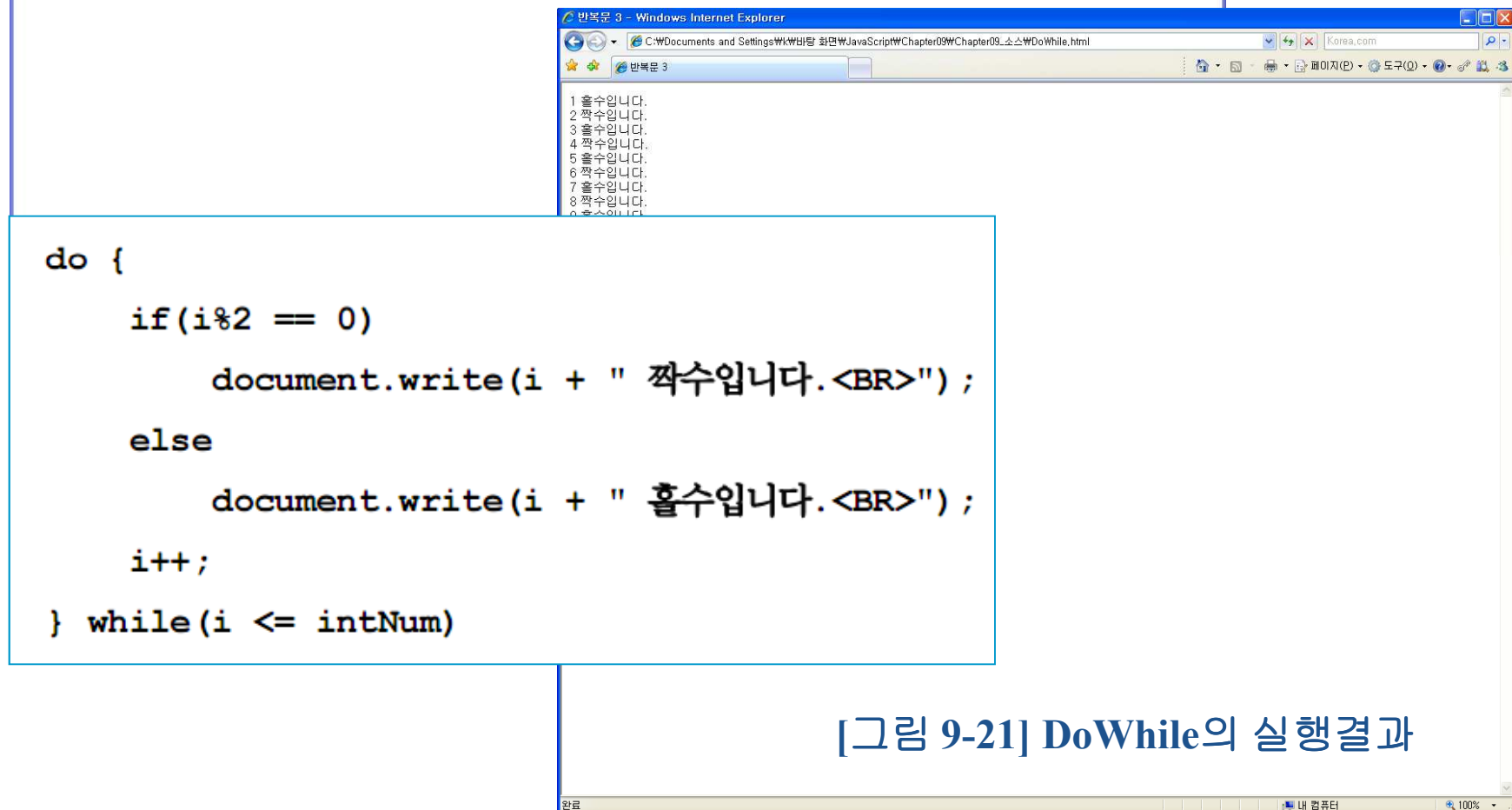
```
do {  
    문장;  
    문장;  
    .....  
    증감식;  
} while (조건식)
```

예제 :

```
<SCRIPT LANGUAGE="JavaScript">  
<!--  
    var intCount = 1;  
    do {  
        document.write("회원님은" + intCount + "번째 방문입니다.<BR>");  
        intCount ++;  
    } while (intCount < 10)  
    //-->  
</SCRIPT>
```



[그림 9-20] DoWhile의 실행결과



[그림 9-21] DoWhile의 실행결과

❖ 함수

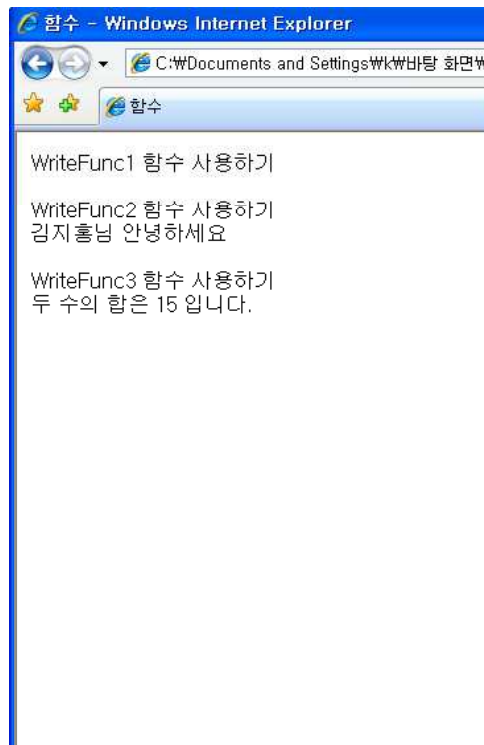
- 반복적으로 실행되는 명령들을 미리 정의하여 필요할 때마다 호출하여 사용
- 프로그램 코드도 구조화 되고, 코드 길이도 짧아지며, 유지 보수도 쉬워짐
- <HEAD> 부분에 작성
- 함수에 인수를 정의하여 인수에 따른 다른 결과를 나타낼 수도 있으며 **return**을 사용하여 함수의 결과 값을 함수를 호출한 곳으로 돌려줌
- **function**을 사용을 사용하여 정의

형식 :

```
function 함수이름( 인수1, 인수2, ...)  
{  
    문장  
    문장  
    ....  
}
```

예제 :

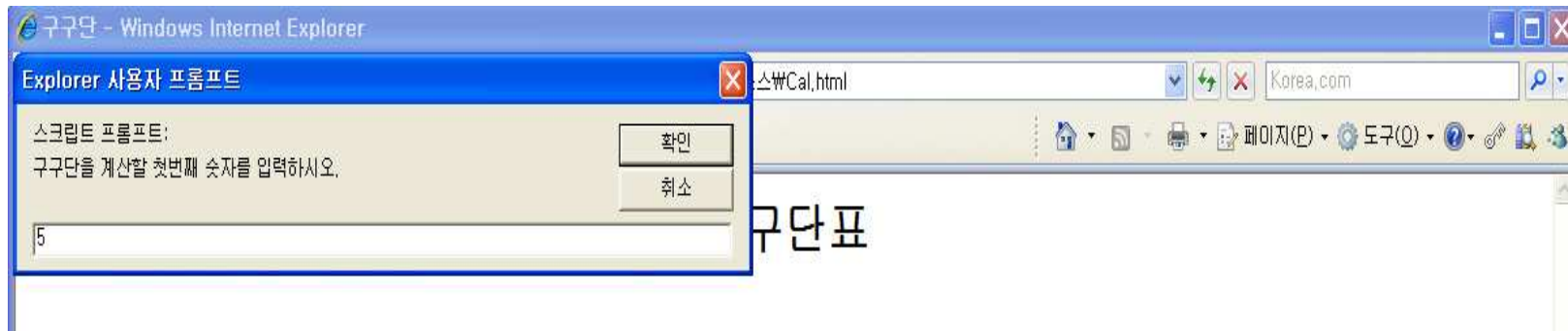
```
function WriteFunc1()  
{  
    document.write("WriteFunc1 함수 사용하기 <BR><BR>");  
}  
function WriteFunc3(intNum1, intNum2)  
{  
    var intSum = intNum1 + intNum2;  
    return intSum;  
}
```



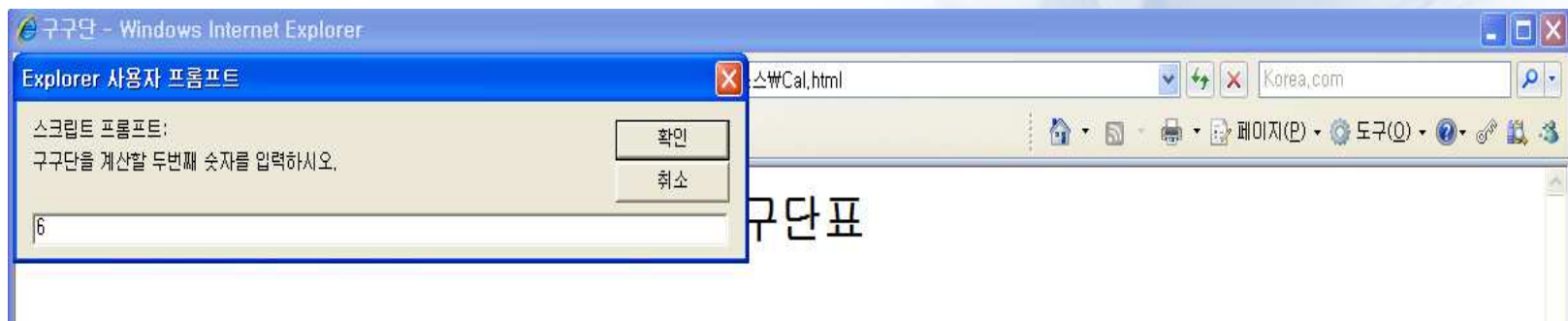
```
function WriteFunc1()
{
    document.write("WriteFunc1 함수 사용하기 <BR><BR>");
}

function WriteFunc2(strName)
{
    document.write("WriteFunc2 함수 사용하기 <BR>");
    document.write(strName + "님 안녕하세요 <BR><BR>");
}
```

[그림 9-22] Func의 실행결과



[그림 9-23] Cal의 실행결과



[그림 9-24] Cal의 실행결과

구구단 - Windows Internet Explorer

C:\Documents and Settings\Wk\바탕 화면\JavaScript\Chapter09\Chapter09_소스\Cal.html Korea.com

구구단

구구단표

1단	2단	3단	4단	5단	6단	7단	8단	9단
1 × 1 = 1	1 × 2 = 2	1 × 3 = 3	1 × 4 = 4	1 × 5 = 5	1 × 6 = 6	1 × 7 = 7	1 × 8 = 8	1 × 9 = 9
2 × 1 = 2	2 × 2 = 4	2 × 3 = 6	2 × 4 = 8	2 × 5 = 10	2 × 6 = 12	2 × 7 = 14	2 × 8 = 16	2 × 9 = 18
3 × 1 = 3	3 × 2 = 6	3 × 3 = 9	3 × 4 = 12	3 × 5 = 15	3 × 6 = 18	3 × 7 = 21	3 × 8 = 24	3 × 9 = 27
4 × 1 = 4	4 × 2 = 8	4 × 3 = 12	4 × 4 = 16	4 × 5 = 20	4 × 6 = 24	4 × 7 = 28	4 × 8 = 32	4 × 9 = 36
5 × 1 = 5	5 × 2 = 10	5 × 3 = 15	5 × 4 = 20	5 × 5 = 25	5 × 6 = 30	5 × 7 = 35	5 × 8 = 40	5 × 9 = 45
6 × 1 = 6	6 × 2 = 12	6 × 3 = 18	6 × 4 = 24	6 × 5 = 30	6 × 6 = 36	6 × 7 = 42	6 × 8 = 48	6 × 9 = 54
7 × 1 = 7	7 × 2 = 14	7 × 3 = 21	7 × 4 = 28	7 × 5 = 35	7 × 6 = 42	7 × 7 = 49	7 × 8 = 56	7 × 9 = 63
8 × 1 = 8	8 × 2 = 16	8 × 3 = 24	8 × 4 = 32	8 × 5 = 40	8 × 6 = 48	8 × 7 = 56	8 × 8 = 64	8 × 9 = 72
9 × 1 = 9	9 × 2 = 18	9 × 3 = 27	9 × 4 = 36	9 × 5 = 45	9 × 6 = 54	9 × 7 = 63	9 × 8 = 72	9 × 9 = 81

[그림 9-25] Cal의 실행결과