

웹프로그래밍 기초

10주차

옥정윤

*참고도서 : 홈페이지 만들기 (이근형 지음/생능출판사)

8. 스크립트 script

8.1 자바 스크립트

8.2 자바 스크립트 사용 방법

8.3 자바 스크립트 기본 문법

8. 스크립트 script

- html : 정적 문서 작성
- **동적문서** : 스크립트, 애플릿, DHTML, XML(eXtensible), CG

(1) Client Script

- 웹 서버의 부담을 줄이고 네트워크의 트래픽도 줄이는 역할
- 스크립트 언어들로는 JavaScript, vbScript 등이 있다.

(2) Server Script

- 클라이언트의 운영체제나 웹 브라우저와는 독립적
- 서버 스크립트로는 PHP, JSP, ASP 등이 있다.

8.2 자바 스크립트 사용 방법

(1) 소스를 공개하는 경우

- <head> 태그 내 또는 <body> 태그 내에 내장
- HTML태그의 속성 값으로 자바스크립트 코드 적용
- HTML태그의 이벤트 핸들러를 자바스크립트 코드로 표현
- 자바스크립트 코드를 URL로 사용

(2) 소스를 공개하지 않는 경우 : 자바 스크립트 파일 내장

(3) 자바스크립트 사용 방법

- A. 내부 자바스크립트
- B. 외부 자바스크립트
- C. 인라인 스크립트

A. 내부 자바스크립트

➤ <head> 태그 내 또는 <body> 태그 내에 내장

➤ <head> 태그는 주로 정의

➤ <body> 태그는 실행

➤ 형식

```
<script language = "javascript" >
```

-
-
-

```
</script>
```

```
<html>
```

```
<head><title> 00 옥정윤 script 태그 넣기 </title></head>
```

```
<body>
```

```
<h1>1. 내부 스크립트 </h1>
```

```
<script language = "JavaScript">
```

```
    document.write("*** 여기는 body 내 자바스크립트***")
```

```
</script >
```

```
</body>
```

```
</html>
```

B. 외부 자바스크립트

- `<head>` 태그 내에 내장
- 웹 문서와 같은 폴더에 저장
- 스크립트 소스 비 공개
- 형식 : head

```
< script language = "javascript" src = "자바스크립트 파일의 URL" >  
  
</script>
```

09_00.js

```
function test1( ) {  
    alert("외부 실행됨_js 파일");  
}
```

```
<head> <!-- 외부 스크립트 -->
```

```
<script language = "JavaScript" src=09_00.js>
```

```
</script >
```

```
</head>
```

```
<body>
```

```
<h1>2. 외부 스크립트 </h1>
```

```
<button type="button" onclick="test1()"> 외부 </button>
```

```
</body>
```


c. 인라인 자바스크립트

- Javascript : 객체
- 태그 내에 직접 사용
- 예

<body>

10_00.html

<h1>3. 인라인 스크립트 </h1>

경고 메시지

<hr color="red">

</body>

8.3 자바 스크립트 기본 문법

(1) 주석처리

- 주석이 한 줄인 경우 : `// 주석내용`
- 주석이 여러 줄인 경우 : `/* 주석내용 */`

`<h1>주석 테스트 </h1>`

`<script language="javascript">`

`document.write("홈페이지 ")` // 메시지 나타내기

// 이 문자열은 나타나지 않는다.

`document.write("body 태그에 자바스크립트 넣기")`

`/* 자바스크립트의 주석 넣기 연습 */`

`</script>`

[단축키]

. 한줄 주석 처리 CTRL+/

. 블록 주석 처리 ALT+SHIFT+A

(2) document.write()

➤ 간단한 메시지 나타내기

예) **document.write**("화면에 출력할 내용")

- 객체를 사용하지 않고 메시지를 넣으면 에러발생
- 만약 따옴표가 겹치면 싱글과 더블로 구분하여 나타내야 한다.
- 비교 : html vs Javascript

(1) html : **** 폰트 테스트 **
**

(2) **<script language = "JavaScript" >**

document.write("")

**document.write("폰트 테스트"+"
")**

</script >

<h1>for 문 연습 </h1>

```
<script language="JavaScript">
```

```
  for (i=1; i<=7; i++){
```

```
    document.write("<font size=",i,"> 어서오세요.</font><br>")
```

```
  }
```

```
</script>
```

(3) 데이터 타입과 문법

A. 식별자 : 변수나 함수이름 등

- 첫 글자는 반드시 영문자나 "_" (밑줄)로 시작하여야 한다.
- 영문자와 숫자 그리고 특수문자의 조합으로 이루어져야 한다.
- 영문자인 경우 대소문자를 구분하여 사용하여야 한다.
- 자바스크립트의 예약어는 사용할 수 없다.
- 가능한 의미 있는 내용으로 만들어 사용한다.

B. 변수

- Var number
- Var ic=10 // 초기 값으로 10 이 저장됨
- Age=20 // 직접 프로그램 내에서 사용

식별자 : 변수나 함수이름 등 예약어는 사용 할 수 없음

[예약어 리스트]

```
abstract arguments  boolean  break byte  
case  catch  char  class*  const  continue  
debugger  default  delete do  double  
else  enum*  eval  export* extends*  
false final  finally float  for function goto  
if implements import* in instanceof int  interface  
let long native  new  null  
package  private  protected  public return  
short static super*  switch  synchronized  
this  throw  throws transient true  
try  typeof  var  void  volatile  
while with  yield
```

C. 상수 : 변하지 않게 지정한 수

- 정수 형, 부동 소수점 형, 문자열 형, 부울 형, null

D. 연산자

- 산술연산자 : +, -, *, /, ++, --, %
- 관계연산자 : <, <=, >, >=, ==, !=
- 조건연산자 : 조건식 ? 구문-1 : 구문-2
 - 결과가 참(true) : -1 / 거짓(false) -2 실행
- 논리연산자 : ! (부정), && (and), || (or)
- 대입연산자 : +=, -=, *=, /=, %=
- 문자열 연산자 : + (문자열 결합)

E. 제어문

- 조건선택문 : If 문, if ~ else 문, 다중 if ~ else 문, switch ~ case 문
- 반복문 : for 문, while 문, do ~ while 문, break 문, continue

F. 함수 : 사용자 정의 함수, 내장 함수

G. 이벤트, 이벤트 핸들러

H. 객체 : 사용자 정의 객체, 내장 객체, 브라우저 객체

G. HTML 이벤트, 이벤트 핸들러

** 이벤트 처리(event processing)를 지원

- 이벤트(event) : 웹 브라우저에서 발생하는 어떤 동작이나 상태변화를 나타내는 것 (HTML 문서 와 사용자 사이의 상호작용에 의해 발생)
- 이벤트 예
 - 사용자가 마우스로 특정 태그 영역을 클릭함(click 이벤트)
 - 사용자가 마우스를 특정 태그 영역 위로 이동시킴(mouseover 이벤트)
 - 특정 <input> 태그의 사용자 입력 값이 변경됨(change 이벤트)
 - 이미지 파일 또는 HTML 문서가 웹 브라우저로의 로드를 마침(load 이벤트)
 - 사용자가 브라우저 창의 크기를 변경함(resize 이벤트)

- 이벤트 처리

- 이벤트 속성

- ✓ 자바스크립트 코드를 속성값으로 갖고, 이벤트가 발생하면 자바스크립트 코드를 실행함

- ✓ 속성이름은 “on이벤트명” 형식임

이벤트 분류	이벤트 속성	이벤트 의미
마우스 관련	onclick	마우스를 클릭함
	onmouseover	마우스가 해당 영역 위에 위치함
키보드 관련	onkeypress	키보드를 누른 상태임
	onkeydown	한 키를 입력함
문서 관련	onload	HTML 문서가 웹 브라우저에 적재됨
	onunload	HTML 문서가 웹 브라우저에서 제거됨
폼(<form>) 관련	onchange	사용자가 입력값을 변경함
	onfocus	사용자가 데이터를 입력할 수 있는 상태임

예) <h1> 8. 나를 눌러보아</h1>

<h1 onclick="document.write('마우스클릭');">나를 눌러보아</h1>

➤ 메시지 대화상자 :

- | | | |
|---------------|-----------------------|----------|
| (1) alert() | : 경고(안내) 메시지 확인 응답 입력 | / 확인 |
| (2) confirm() | : yes 또는 no 의 응답 입력 | / 확인, 취소 |
| (3) prompt() | : 문자열 입력 | / 입력 |

(1) alert

예) <h1>9. 전구 : alert </h1>

<button onclick=" alert('메세지 ') "> 전구 on/off 방법</button>

(2) confirm

- 확인(Yes, OK) 또는 취소(No, Cancel) 의 응답을 요구하는 메시지를 제공하고 응답을 받아 반환

<h1>10. 전구 : confirm </h1>

```
<script>
```

```
answer = confirm("어두워졌습니다. 전구를 켜까요 ?");
```

```
if (answer == true) {
```

```
    document.write("전구 상태 : On");
```

```
    document.write("<img src='img/on.gif' >");
```

```
}
```

```
else {
```

```
    document.write("전구 상태 : Off");
```

```
    document.write("<img src='img/off.gif' >");
```

```
}
```

```
</script>
```

(3) prompt()

- 대화상자를 통해 사용자에게 문자열의 입력을 요구하고 사용자가 입력한 문자열을 반환함
- 수치값들은 문자열 변수에 저장한 후, 수치형으로 변환해서 사용해야 함
- 문자열 -> 수치형 변환함수들 :

parseInt(), parseFloat(), Number()

```
name = prompt("이름은 ?");
```

```
kor = prompt("국어 시험성적은 ?");
```

```
math = prompt("수학 시험성적은 ?");
```

```
total = Number(kor) + Number(math) ;
```

```
avg = total /2;
```

```
document.write("<table border=1 >");
```

```
document.write("<tr><td>이름</td> <td>" + name + "</td></tr>");
```

```
document.write("<tr><td>국어</td><td>" + kor + "</td></tr>");
```

```
document.write("<tr><td>수학</td><td>" + math + "</td></tr>");
```

```
document.write("<tr><td>총점</td> <td>" + total + "</td></tr>");
```

```
document.write("<tr><td>평균</td> <td>" + avg + "</td></tr>");
```

```
document.write("</table>");
```

8.4 자바 스크립트 기본 문법 (2) : 제어문, 함수

➤ 프로그램 구조

- 순차 구조 : 명령문들을 순차적 나열
 - 선택문 : if, if~else , if~else if, switch
 - 반복문 : while, for, do~while, for~in
 - 그 외 : continue, break
-
- 알고리즘 : 프로그램을 작성하기 위한 생각의 표현, 순서의 정리
 - 알고리즘 안에 포함되어야 하는 내용 : 입력, 출력, 출력을 제공하기 위한 과정
 - 알고리즘의 표현 방법 : 자연어, 의사코드(Pseudo-code), 순서도(flowchart)

➤ 의사코드 예

A. 알바비용 계산

BEGIN

input hours

input rate

pay = hours * rate

print pay

END

B. 삼각형의 면적을 구하기

공식 : 면적 = (밑변 × 높이) / 2

면적(area) : 밑변(base) 10, 높이(height) 20

START

INPUT base : 10

INPUT height : 20

area = base × height

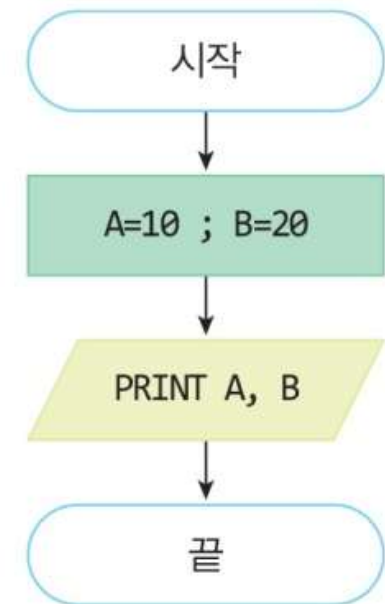
area = area / 2

OUTPUT area

END

➤ 순서도(flow chart)를 이용한 알고리즘 표현 방법

- 일정 기호를 사용하여 문제 해결을 위한 처리 과정을 논리적으로 도식화 → 순서도를 바탕으로 명확하고 논리적인 프로그램 작성
- 직관적이고 이해가 쉬움 (알고리즘 흐름 파악이 용이)
- 표준 기호 이용 : 박스와 연결선으로 표현한 도표(diagram) (설계도 역할)
- 일의 흐름을 나타내기 때문에 '흐름도'라고도 함
- 순서도는 반드시 컴퓨터의 이용을 전제로 하는 것은 아님
 - 개략 순서도 : 전체를 작업 단위로 개괄적으로 작성
 - 상세 순서도 : 실행 가능하도록 줄 단위로 상세하게 작성



A. 순서도의 기호와 의미

심벌 형태	이름	의미
	시작/끝 (start/stop)	순서도의 시작과 끝을 나타낸다.
	처리 (process)	지정된 동작, 연산, 값이나 데이터의 이동 등 모든 처리과정을 나타낸다.
	판단 (decision)	주어진 조건을 비교한 후 해당되는 조건에 따라 왼쪽 또는 오른쪽으로 흐름이 나누어지게 된다.
	입력/출력 (input/output)	일반적인 입력과 출력을 나타낸다.
	연결자 (connector)	흐름이 다른 곳으로 연결되거나 다른 곳으로부터 연결되는 입구를 나타낸다.
	프린트 (printing)	프린터를 통하여 출력한다.
	흐름선 (flow line)	명령어의 흐름을 나타내며, 화살표 방향의 순서대로 진행됨을 나타낸다,

➤ 예제 1. 순차형(sequence)

➤ 알바비용을 계산하시오.

➤ 의사코드 예

BEGIN

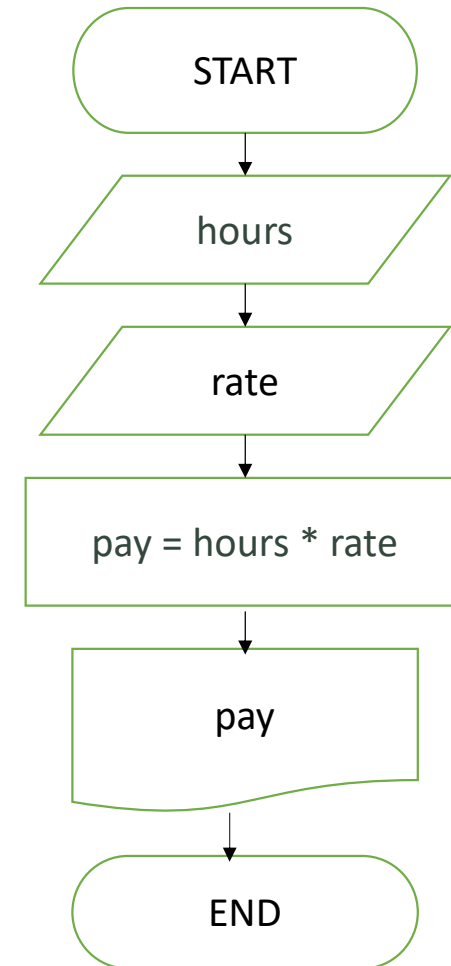
input hours

input rate

pay = hours * rate

print pay

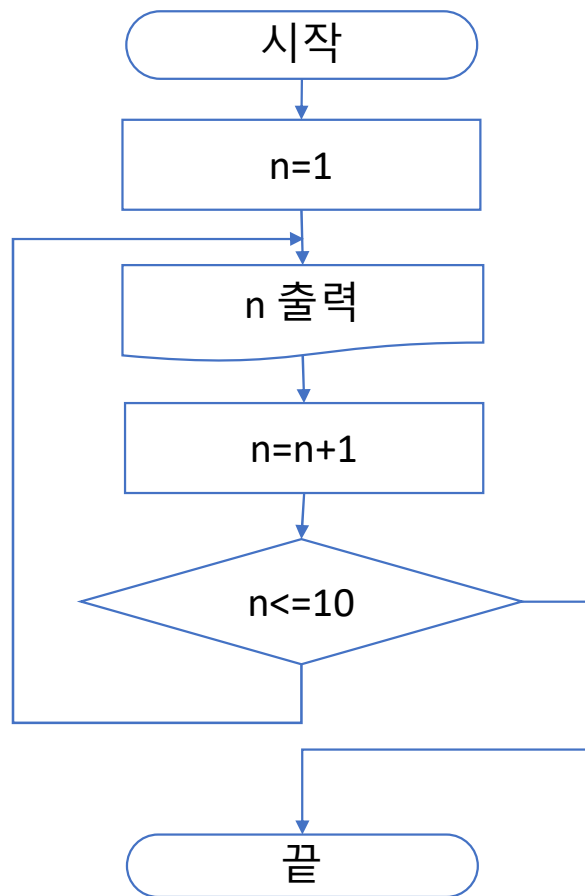
END



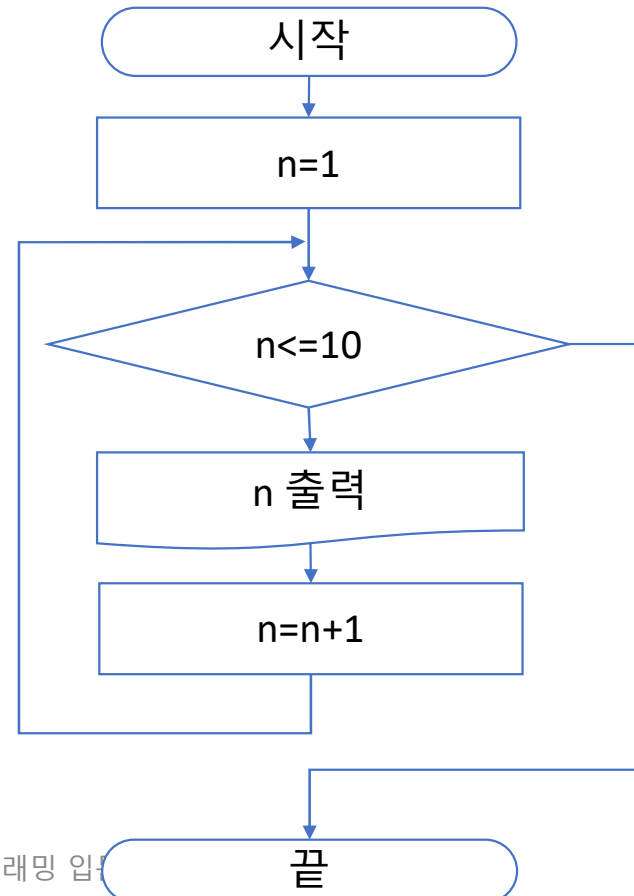
➤ 예제 3. 반복형(iteration)

- 변수 n을 선언하고 1부터 10까지 출력하는 순서도를 작성하시오.

선처리 후검사



선검사 후처리



(1) if 문 : 단일 선택 구조

➤ 형식) **if (조건식)**

{

문장-1 ;

}

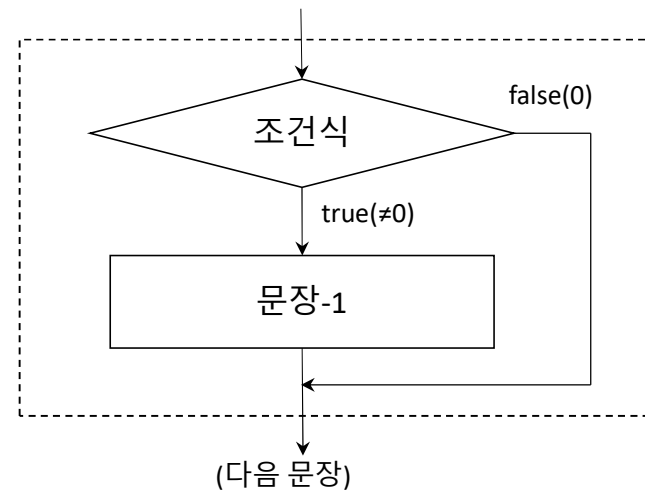
10_00.html #11

```
if (score<=70) {
```

```
    document.write("추가 과제를 부여합니다.");
```

```
}
```

```
document.write("수업을 종료합니다!");
```



(2) if ~ else 문 : 양자택일 구조

➤ 형식)

if (조건식)

{

문장-1 ;

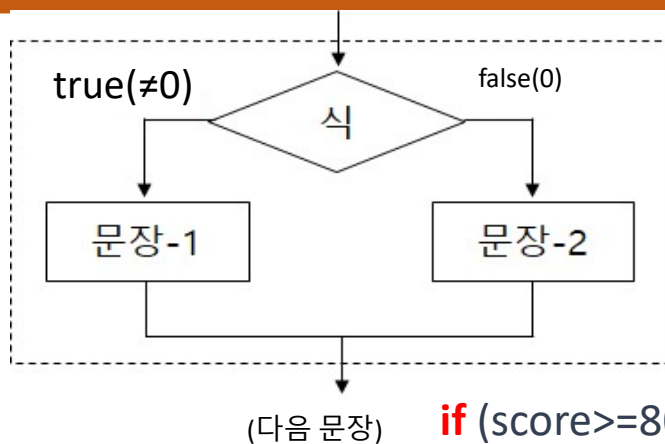
... }

else

{

문장-2;

... }



10_00.html #12

if (score>=80) {

document.write("합격 입니다");

document.write("추가과제는 없습니다.
");

} else {

document.write("불합격 입니다");

document.write("추가과제가 있습니다.
");

}

document.write("수업을 종료합니다!");

(3) 다중 if 문 : 다중 선택 구조

➤ 형식 : **if (조건식-1)** { 문장-1; ... }

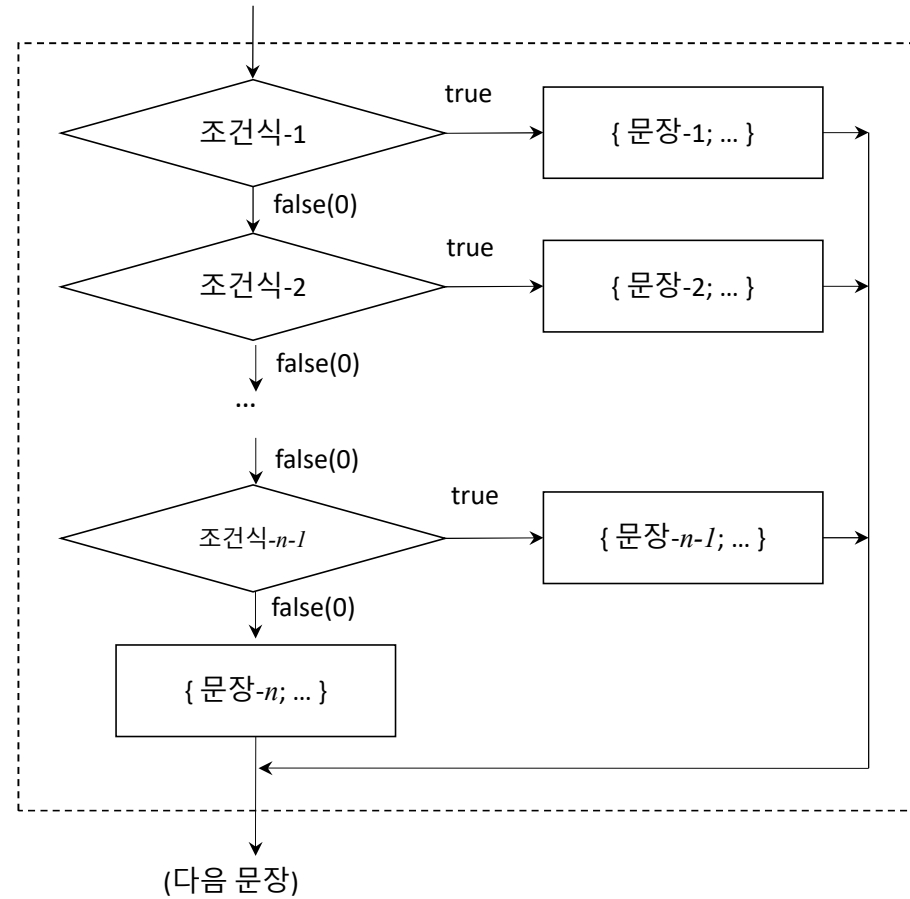
else if (조건식-2) { 문장-2; ... }

else if (조건식-3) { 문장-3; ... }

...

else if (조건식-n-1) { 문장-n-1; ... }

else { 문장-n; ... }



시험성적(score)이

90점 이상이면, ❶ 학점(grade)은 A 이다.

80~89점 사이 이면, ❷ 학점(grade)은 B 이다.

70~79점 사이 이면, ❸ 학점(grade)은 C 이다.

60~69점 사이 이면, ❹ 학점(grade)은 D 이다.

그 외, ❺ 학점(grade)은 F 이다.

```
{
  if (score>=90) grade="A";
  if (score<90 && score>=80) grade="B";
  if (score<80 && score>=70) grade="C";
  if (score<70 && score>=60) grade="D";
  else grade="F";
}
```

```
if (score >=90) grade="A";
else if (score>=80) grade="B";
else if (score>=70) grade="C";
else if (score>=60) grade="D";
else grade="F";
```


(4) switch 문 : 다중 선택 구조

➤ 형식 : // break문은 생략 할 수 있음

```
switch(수식) {
```

```
    case 값1 : 문장-1; ... ; break;
```

```
    case 값2 : 문장-2; ... ; break;
```

```
    ...
```

```
    case 값n-1 : 문장-n-1; ... ; break;
```

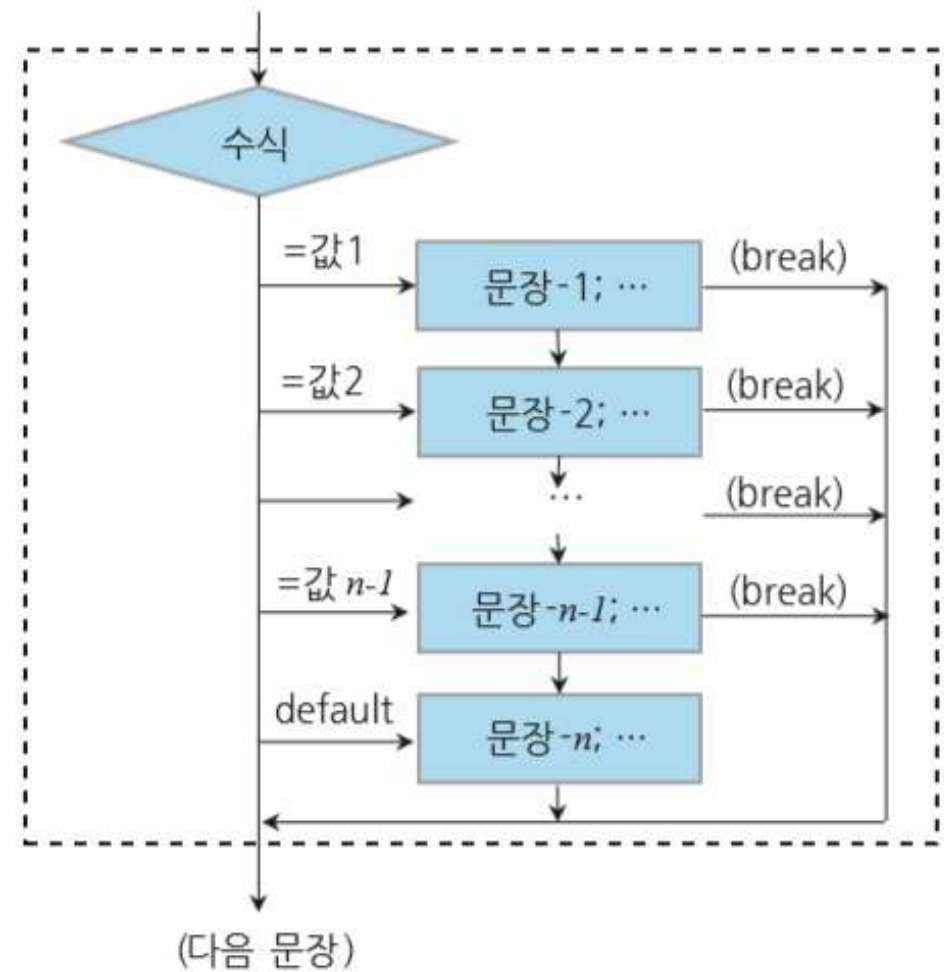
```
    default : 문장-n ; ... ;
```

```
}
```

➤ break 문

➤ 현재 실행중인 문장을 종료하고 다음 문장으로 이동함

➤ 선택문과 반복문에서 사용함



```
switch ( parseInt(score/10) ) {  
    case 10: grade="A"; break;  
    case 9: grade="A"; break;  
    case 8: grade="B"; break;  
    case 7: grade="C"; break;  
    case 6: grade="D"; break;  
    default: grade="F";  
}
```

<h1>14. 학점판정 </h1>

<script>

score = prompt("성적을 입력하세요")

switch (**parseInt(score/10)**) {

case 10: alert("성적은 A입니다"); break;

case 9: alert("성적은 A입니다"); break;

case 8: alert("성적은 B입니다"); break;

case 7: alert("성적은 C입니다"); break;

case 6: alert("성적은 D입니다"); break;

default: alert("성적은 F입니다");

}

</script>

(5) for 문

➤ 형식 :

```
for ( 초기값 ; 종료값 ; 증감값 ) {  
    반복루프;  
}
```

10_00.html #15

➤ 예) 문장 document.write("welcome to javascript!");의 10회 반복

```
for ( i=1 ; i<=10 ; i=i+1 ) {  
    document.write("welcometojavascript!");  
}
```

웹프로그래밍 기초

```
welcometojavascript!  
welcometojavascript!  
welcometojavascript!  
welcometojavascript!  
welcometojavascript!  
welcometojavascript!  
welcometojavascript!  
welcometojavascript!  
welcometojavascript!  
welcometojavascript!  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
0까지의 5의 배수의 합 : 0  
5까지의 5의 배수의 합 : 5  
10까지의 5의 배수의 합 : 15  
15까지의 5의 배수의 합 : 30  
20까지의 5의 배수의 합 : 50  
25까지의 5의 배수의 합 : 75  
30까지의 5의 배수의 합 : 105  
35까지의 5의 배수의 합 : 140  
40까지의 5의 배수의 합 : 180  
45까지의 5의 배수의 합 : 225  
50까지의 5의 배수의 합 : 275  
55까지의 5의 배수의 합 : 330  
60까지의 5의 배수의 합 : 390  
65까지의 5의 배수의 합 : 455  
70까지의 5의 배수의 합 : 525  
75까지의 5의 배수의 합 : 600  
80까지의 5의 배수의 합 : 680  
85까지의 5의 배수의 합 : 765  
90까지의 5의 배수의 합 : 855  
95까지의 5의 배수의 합 : 950  
100까지의 5의 배수의 합 : 1050
```

(6) while 문

➤ 형식

```
while (조건식) {  
반복루프;  
}
```

10_00.html #16

➤ 예) 문장 document.write("welcome to javascript!");의 10회 반복

```
i=1;
```

```
while (i<=10) {
```

```
    document.write("welcometojavascript!");
```

```
    i=i+1;
```

```
}
```

```
welcometojavascript!  
welcometojavascript!  
welcometojavascript!  
welcometojavascript!  
welcometojavascript!  
welcometojavascript!  
welcometojavascript!  
welcometojavascript!  
welcometojavascript!  
welcometojavascript!  
0까지의 5의 배수의 합 : 0  
5까지의 5의 배수의 합 : 5  
10까지의 5의 배수의 합 : 15  
15까지의 5의 배수의 합 : 30  
20까지의 5의 배수의 합 : 50  
25까지의 5의 배수의 합 : 75  
30까지의 5의 배수의 합 : 105  
35까지의 5의 배수의 합 : 140  
40까지의 5의 배수의 합 : 180  
45까지의 5의 배수의 합 : 225  
50까지의 5의 배수의 합 : 275  
55까지의 5의 배수의 합 : 330  
60까지의 5의 배수의 합 : 390  
65까지의 5의 배수의 합 : 455  
70까지의 5의 배수의 합 : 525  
75까지의 5의 배수의 합 : 600  
80까지의 5의 배수의 합 : 680  
85까지의 5의 배수의 합 : 765  
90까지의 5의 배수의 합 : 855  
95까지의 5의 배수의 합 : 950  
100까지의 5의 배수의 합 : 1050
```

(7) do~while 문

➤ 형식

```
do {  
    반복루프;  
} while (조건식);
```

10_00.html #17

<h1>17. do-while 반복 </h1>

<script>

sum=0, i=0;

do{

sum=sum+i;

document.write(i,"까지의 5의 배수의 합 :", sum, "
");

i=i+5

} while(i<=100) ;


</script>

0까지의 5의 배수의 합 : 0
5까지의 5의 배수의 합 : 5
10까지의 5의 배수의 합 : 15
15까지의 5의 배수의 합 : 30
20까지의 5의 배수의 합 : 50
25까지의 5의 배수의 합 : 75
30까지의 5의 배수의 합 : 105
35까지의 5의 배수의 합 : 140
40까지의 5의 배수의 합 : 180
45까지의 5의 배수의 합 : 225
50까지의 5의 배수의 합 : 275
55까지의 5의 배수의 합 : 330
60까지의 5의 배수의 합 : 390
65까지의 5의 배수의 합 : 455
70까지의 5의 배수의 합 : 525
75까지의 5의 배수의 합 : 600
80까지의 5의 배수의 합 : 680
85까지의 5의 배수의 합 : 765
90까지의 5의 배수의 합 : 855
95까지의 5의 배수의 합 : 950
100까지의 5의 배수의 합 : 1050


(8) continue 문

- 반복루프에서 나머지 문장들의 실행은 생략하고, 반복루프 다음의 실행 위치로 이동함
- 반복문에서 사용함
- continue 문을 for, while, do~while 문에서 사용하면, 다음 실행 위치가 각각 다름

```
for ( i=1; i<= 100; i++ ) {  
    ...  
    if (조건식) continue;  
    ...  
}
```



```
while ( i<=100 ) {  
    ...  
    if (조건식) continue;  
    ...  
}
```



```
do {  
    ...  
    if ( 조건식 ) continue;  
    ...  
} while ( i<=100 );
```

