

[10차시 교안]

<자바스크립트 기초2>

1. 제어문

자바스크립트 문장은 순차적으로 수행

제어문 : 실행되는 순서에 영향을 주는 문장

(1) 조건문

가. if 문 : 조건이 참일 때만 어떤 코드를 실행하고 싶을 때 사용

나. if... else 문 : 조건이 참이면 어떤 코드를 실행하고 조건이 거짓이면 다른 코드를 실행하고 싶을 때 사용

다. switch 문 : 많은 코드 중에서 하나를 선택하여 실행하고 싶은 경우 사용

(2) 반복문

가. while : 지정된 조건이 참이면 반복 실행

나. for : 정해진 횟수 동안 코드를 반복 실행

2. 조건문

조건에 따라 결정을 내려야 하는 경우 사용

(1) if문

형식	if(조건식) 문장;
설명	조건식이 참인 경우에만 문장 실행
<pre>if (time<12) //조건식 { greeting="Good Morning!"; // 조건식이 참인 경우 실행 }</pre>	

(2) if...else문

조건이 참이 아닌 경우에 실행할 코드를 넣고 싶은 경우

형식	<code>if(조건식)</code> <code>문장1;</code> <code>else</code> <code>문장2;</code>
설명	만약 조건식이 참이면 문장1 실행 그렇지 않으면 문장2 실행

```
if (time<12)
{
    msg="Good Morning!";
}
else{
    msg="Good Afternoon!";
}
```

(3) 연속적인 if문

조건이 다중으로 분기되어야 하는 경우(학점 등)

```
<script>
    var msg = "";
    var time = new Date().getHours();           // 현재 시간 반환
    if (time < 12) {                             // 12시 이전이면
        msg = "Good Morning";
    }
    else if (time < 18) {                         // 오후 6시 이전이면
        msg = "Good Afternoon";
    }
    else {                                       // 그렇지 않으면(오후 6시 이후이면)
        msg = "Good evening";
    }
    alert(msg);
</script>
```

(4) switch문

- ① if문과 비슷
- ② 조건에 따라 프로그램의 흐름을 분기 시키기 위해 사용
- ③ 분기 가능한 실행 경로가 여러 개인 경우 유리

형식	<pre>switch(제어식) { case c1 : // 제어식의 값이 c1이면 실행 문장; break; case c2 : // 제어식의 값이 c2이면 실행 문장; break; ... default : // 위의부분에서 일치하는 값이 없으면 실행 문장; break; }</pre>
----	--

- ④ 사용자로부터 점수를 입력받아 학점을 출력하는 프로그램

```
<script>
var grade = prompt("성적을 입력하시오:", "A-F사이의 문자로");
switch (grade) {
    case 'A': alert("잘했어요!");
              break;
    case 'B': alert("좋은 점수군요");
              break;
    case 'C': alert("괜찮은 점수군요");
              break;
    case 'D': alert("좀더 노력하세요");
              break;
    case 'F': alert("다음 학기 수강하세요");
              break;
    default: alert("알 수 없는 학점입니다.")
}
</script>
```

(5) 숫자 게임 예제

프로그램이 가지고 있는 정수를 사용자가 알아맞히는 게임

- ① 사용자가 숫자를 제시하면 프로그램이 저장한 정수와 비교하여 제시된 정수가 더 높은지 낮은지 알려줌
- ② 정수 범위는 1부터 100 사이로 한정
- ③ 게임이 끝나면 몇 번 만에 맞추었는지 함께 출력

```

<body>

  <h2>숫자 맞추기 게임</h2>

  이 게임은 컴퓨터가 생성한 숫자를 맞추는 게임입니다. 숫자는 1부터 100
  사이에 있습니다.

  <form>

    숫자: <input type="text" id="user" size="5">

           <input type="button" value="확인" onclick="guess();">

    추측횟수: <input type="text" id="guesses" size="5">

    힌트: <input type="text" id="result" size="16">

  </form>

</body>

</html>

```

```

<html>
<head>
  <title></title>
  <script>
    var computerNumber = 53; // 정답
    var nGuesses = 0;        // 추측 횟수
    function guess() {
      var result = "";        // 결과 메시지

      var number = parseInt(document.getElementById("user").value);
      nGuesses++;             // 추측 횟수 증가

      if (number == computerNumber) result = "성공입니다.";
      else if (number < computerNumber) result = "낮습니다.";
      else result = "높습니다.";

      document.getElementById("result").value = result;
      document.getElementById("guesses").value = nGuesses;
      return true;
    }
  </script>
</head>

```

3. 반복문

같은 처리 과정을 여러 번 되풀이하는 것

성적처리 : 30명의 학생의 과목 점수에 대한 평균을 내는 경우

종류

while - 지정된 조건이 참이면 반복 실행

for - 정해진 횟수 동안 코드를 반복 실행

"Hello World!"를 5번 출력하는 코드 비교

```
document.write("Hello World! <br>");
document.write("Hello World! <br>");
document.write("Hello World! <br>");
document.write("Hello World! <br>");
document.write("Hello World! <br>");
```

```
For(var i=0; i<5; i++)
{
    document.write("Hello World! <br>");
}
```

(1) for문

① 정해진 횟수만큼 반복하여 실행하는 구조

② 초기식은 반복 루프를 시작하기 전에 한번만 실행. 변수값 초기화 용도

```
<script>
    var i = 0;    //루프 제어변수
    for (i = 0; i < 10; i++) {
        document.write("카운터 : " + i + "<br>");
    }
</script>
```

③ 초기식에 여러 개의 문장을 콤마로 연결 가능

```
for(var i=0, len=10; i < len ; i++)
```

④ 반복문을 사용해 <h1>요소부터 <h6>요소까지의 화면 출력 프로그램

```
<script>
    for (i = 1; i <= 6; i++) {
        document.write("<h" + i + ">header" + i)
        document.write("</h" + i + ">")
    }
</script>
```

⑤ 섭씨 온도를 화씨 온도로 변환하는 예제

```
<html>
<head>
  <title>온도 변환기</title>
</head>
<body>
  <table border="3">
    <tr>
      <td>섭씨 온도</td>
      <td>화씨 온도</td>
    </tr>
    <script>
      for (celsius = 0; celsius <= 10; celsius = celsius + 1) {
        document.write("<tr><td>" + celsius + "</td><td>"
          + ((celsius * 9.0 / 5) + 32) + "</td></tr>");
      }
    </script>
  </table>
</body>
</html>
```

⑥ 중첩 반복문 예제

- a. 하나의 for루프 안에 다른 for루프가 내장될 수 있음
- b. 중첩된 for 루프를 이용하여 구구단표를 화면에 표시하는 프로그램

```
<script>
  document.write("<h1>구구단표</h1>");
  document.write("<table border=2 width=50%>");

  for (var i = 1; i <= 9; i++) {
    document.write("<tr>");
    document.write("<td>" + i + "</td>");
    for (var j = 2; j <= 9; j++) {
      document.write("<td>" + i * j + "</td>");
    }
    document.write("</tr>");
  }
  document.write("</table>");
</script>
```

(2) while문

- ① 주어진 조건이 만족되는 동안 문장을 반복 실행하는 제어문

형식	While(조건식) 문장;
설명	조건식이 참이면 문장을 반복 실행

- ② 변수 i의 값을 0부터 9까지 하나씩 증가시키면서 브라우저의 화면에 출력하는 예제

```
<script>
  var i = 0;
  while (i < 10) {
    document.write("카운터 : " + i + "<br />");
    i++;
  }
</script>
```

(3) do/while 문

- ① while문과 비슷하나 반복 조건을 루프의 처음이 아니라 루프의 끝에서 검사
② 일단 문장을 한 번 실행하고 나서 조건 검사하고 싶을 때 사용 : 최소 한번은 실행

```
<script>
  var i = 0;
  do {
    document.write("카운터 : " + i + "<br />");
    i++;
  } while (i < 10);
</script>
```

(4) for/in 반복문

- ① 객체 안의 속성들에 대하여 어떤 처리를 반복할 수 있는 구조
- ② 객체 : 속성과 함수를 모아놓은 집합

```
<script>
    var myCar = { make: "BMW", model: "X5", year: 2013 };
    var txt="";

    for (var x in myCar) {
        txt += myCar[x] + " ";
    }

    document.write(txt);
</script>
```

(5) break 문

반복 루프를 벗어나기 위해 사용

반복 루프 안에서 break 문이 실행되면 반복문 실행 종료

```
<script>
    var msg = "";
    for (var i = 0; i < 10; i++) {
        if (i == 3) break;
        msg += i + "<br>";
    }

    document.write(msg);
</script>
```


(6) continue 문

- ① 현재 실행하고 있는 반복 과정의 나머지를 생략하고 다음 반복을 시작하게 함
- ② 0부터 10까지의 정수 중에서 3만 제외하고 출력하는 프로그램

```
<script>
    var msg = "";
    for (var i = 0; i < 10; i++) {
        if (i == 3) continue;
        msg += i + "<br>";
    }
    document.write(msg);
</script>
```

4. 배열

(1) 정의

- ① 많은 값을 한꺼번에 저장할 수 있는 저장 장소
- ② 데이터 집단에 하나의 이름을 부여한 변수 집합
각각의 데이터에 숫자로 된 번호(첨자)를 붙여서 접근하는 방법

(2) 필요성

- ① 많은 값을 저장할 수 있는 공간이 필요할 때 배열 사용
 - a. 학생이 10명의 평균을 계산할 때, 성적을 저장하기 위해서 10개의 변수 필요
 - b. 만약 학생이 30명이라면 30개, 100명이라면 100개
- ② 서로 관련된 데이터를 차례로 접근하여서 처리 가능
 - a. 관련 데이터가 서로 다른 이름을 사용하고 있다면 각 변수의 이름 기억
 - b. 배열을 사용하면 하나의 이름을 공유하여 사용 가능

(3) 배열 생성 방법

가. 리터럴로 배열 생성(array 객체로 배열 생성)

- ① 구체적인 값을 가지고 배열 생성
- ② 대괄호 [] 사용
- ③ 배열에 저장된 값은 정수 인덱스(0부터 시작)를 가지고 접근 가능
첫번째 요소 fruit[0]

```
var fruits = ["apple", "banana", "peach"];
```

```
document.write(fruits[0] + "<br>");  
document.write(fruits[1] + "<br>");  
document.write(fruits[2] + "<br>");
```

나. Array 객체로 배열 생성

① 비어있는 배열 생성

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

② 배열이 값 저장 시 인덱스 사용(인덱스는 0부터 시작)

```
Fruits[0] = "Apple";
```

③ 선언과 동시에 초기화

```
var fruits = new Array("apple", "banana", "orange");
```

(4) 배열의 특징

하나의 배열에 여러 가지 종류의 객체를 혼합해서 저장 가능

```
var comp = new Array();  
comp[0] = "Apple";      //문자열  
comp[1] = new Date();   //객체  
comp[2] = 3;            //정수
```

(5) 객체 Array의 속성과 메서드

length 속성 : 배열의 크기

배열 요소를 반복 처리할 때 사용

```
for( i = 0 ; i < fruits.length ; i++) {  
    .....  
}
```

(6) 연관 배열

- ① 키(key, 단순 문자열)를 이용해 값을 저장하였다가 추출 가능
인덱스 대신 사용 가능(입력 양식 필드에서 값 가져올 때 편리)

```
<html>  
<body>  
  <form name="myForm">  
    필드1<input type="text" name="a0"><br />  
    필드2<input type="text" name="a1"><br />  
    필드3<input type="text" name="a2"><br />  
    <input type="button" value="초기화" onclick="init();">  
  </form>  
  <script>  
    function init() {  
      for (var i = 0; i < 3; i++) {  
        document.myForm["a" + i].value = i;  
      }  
    }  
  </script>  
</body>  
</html>
```