

주사위 게임-2

류관희
충북대학교

주사위 게임 규칙

- 플레이어
 - 두 개의 주사위를 던졌을 때 두 주사위 윗면 순자의 합
 - The First Throw(두 주사위의 합)
 - 합: 7 혹은 11 => Win
 - 합: 2, 3, 혹은 12 => Lost
 - 합: 4, 5, 6, 8, 9, 10 => rethrow
 - The Second Throw
 - 합: 첫번째 던진 주사위 합과 같으면 => Win
 - 합: 그렇지 않으면 => Lost

주사위 게임 로직

sum = 두 번 던진 주사위의 합

```
firstturn=true;
```

```
if (firstturn) {
```

```
    switch(sum) {
```

```
        case 7: case 11: document.f.outcome.value = "이겼습니다."; break;
```

```
        case 2: case 3: case 12: document.f.outcome.value = "졌습니다."; break;
```

```
        default: point=sum; firstturn=false; document.f.stage.value="다시 던지세요.";
```

```
                document.f.outcome.value = "  ";
```

```
                document.f.pv.value=point; }
```

```
    } else {
```

```
        switch(sum) {
```

```
            case point: document.f.outcome.value = "이겼습니다.";
```

```
                document.f.stage.value="새로 시작"; document.f.pv.value=point;
```

```
                firstturn=true; break;
```

```
            default: document.f.outcome.value = "졌습니다";
```

```
                document.f.stage.value="새로 시작"; document.f.pv.value=sum;
```

```
                firstturn=true; break;}
```

```
}
```

자바스크립트 – 자바스크립트의 시작

- 자바 스크립트의 삽입 위치는?

- 기본구조

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

```
<!--
```

```
    진짜 자바스크립트 코드
```

```
// ->
```

```
</SCRIPT>
```

- 자바스크립트의 실행시기는?

자바스크립트 – 자바스크립트 삽입과 실행

- 함수형

```
<script language=javascript>  
  function 함수명( )  
  {  
    ~~자바스크립트 소스~~  
  }  
</script>
```

```
<태그명 이벤트핸들러="함수명()">
```

- 링크형

```
<script language=javascript src="js 파일의 전체  
  경로"></script>
```

자바 스크립트

- scripting language is a lightweight programming language.
- JavaScript is programming code that can be inserted into HTML pages.
- JavaScript code can be executed by all modern web browsers.
- JavaScript is easy to learn.

자바스크립트 – 주석달기

- 한 행을 주석문 처리
//주석 처리할 행, 문장
- 두 행 이상에 걸치는 주석문 처리
/* 주석 처리할 영역 */
- HTML 문서의 주석
<!-- 주석 처리할 영역 -->

배열, 연산자, 사용자 정의 함수 – 기본 실행문

- 변수 선언문 : `var i = 10`
- 대입문 : `i = 10`이나 `i = "masan"`
- 조건문
`if(i < 10) document.write("조건만족")`
- 순환문
`for(var i = 0; i < 10; i++){document.write(i)}`

배열, 연산자, 사용자 정의 함수 – 변수

- 변수의 데이터 형(type)
 - Numbers(숫자형), String(문자열형)
 - Boolean(논리형), Null(널)
- 변수의 명명시 주의사항
 - 예약어, 함수명, 객체명, 속성명, 사용 중인 변수 등은 사용할 수 없다.
 - 변수는 영자나 밑줄로만 시작한다.
 - 대소문자를 구별하되, 의미있는 이름을 붙인다.

배열, 연산자, 사용자 정의 함수 – 배열 변수 선언법

- 배열(array)은 같은 형, 같은 길이의 데이터를 2개 이상 붙여서 동일한 변수로 처리하는 것
- 기본 형식
var 배열 변수명 = new Array()
배열 변수명[0]=값
배열 변수명[1]=값
배열 변수명[2]=값
- var 배열 변수명 = new Array(배열개수)
- var jumsu = new Array(값1, 값2, 값3)

배열, 연산자, 사용자 정의 함수 – 연산문

- 산술연산문 : +, -, *, /, %
 - 증감연산 : ++, --
- 대입연산문 : =, +=, -=, *=, /=, %=
- 조건 연산자:
 - 변수명=(조건식)? 명령1 : 명령2
- 논리연산문 : &&, ||, !,
 - 관계연산자 : >, <, >=, <=
 - 비교연산 : ==, !=
- 연결연산문 : “happy” + “day”

배열, 연산자, 사용자 정의 함수 – 연산기호의 우선순위

- 산술 > 논리 > 대입

1. ()

2. ! ++ --

3. * / %

4. + -

5. < <= > >=

6. == !=

7. &&

8. ||

9. = += -= *= /= %=

배열, 연산자, 사용자 정의 함수 – 사용자 정의 함수

- FUNCTION은 프로그램의 형식을 완전히 갖추지 않은 부속 프로그램으로, 복잡한 계산을 하거나 자주 사용되는 루틴을 정형화할 때 쓰인다.
- 함수의 정의
 <!--
 function makeWindow(){
 window.open("allim.htm","new","width=200 height=200")}
 //-->
- 함수의 호출
 <body onload="makeWindow()">

배열, 연산자, 사용자 정의 함수 – 사용자 정의 함수의 종류

- 매개변수가 없는 함수

```
Function test(){...}
```

- 매개변수가 있는 함수

```
Function test(name){...}
```

- 리턴 값이 있는 경우

```
Function test(question){  
    Ans=confirm(question)  
    Return ans  
}
```

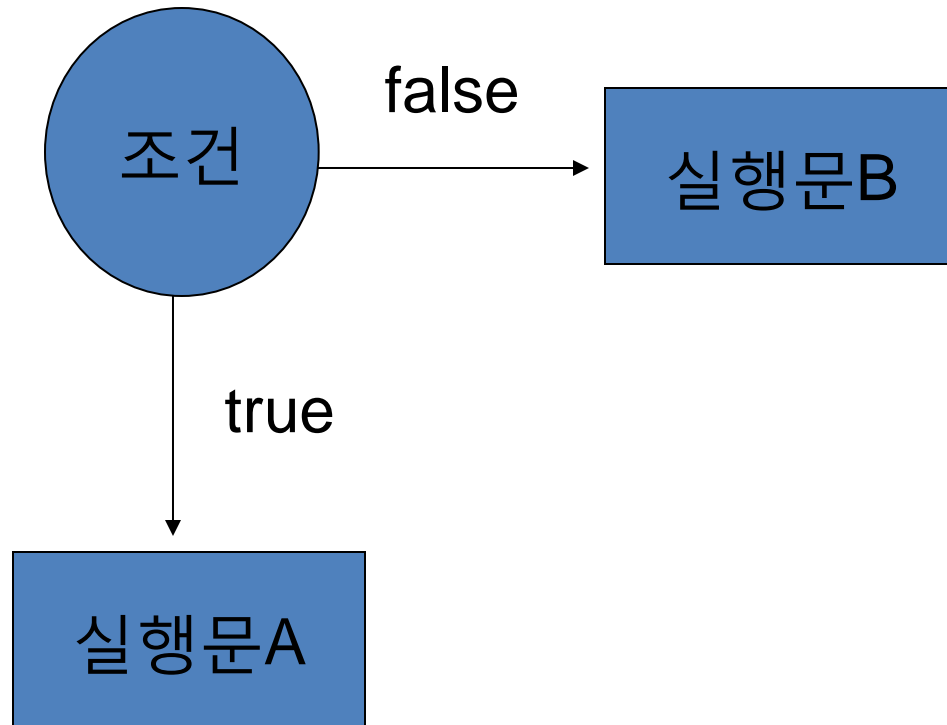
제어문과 내장 함수 – IF 조건문(1)

If(조건)

실행문A

else

실행문B



제어문과 내장 함수 – IF 조건문(2)

- 형식1

```
if(조건)  
    명령문
```

- 형식2

```
if(조건){  
    명령문1  
    명령문2  
}  
else{  
    명령문1  
    명령문2  
}
```

- 중첩 IF문

```
if(조건)  
    명령문  
else if(조건)  
    명령문  
else if(조건)  
    명령문  
else(조건)  
    명령문
```


제어문과 내장 함수 – SWITCH 문

```
switch(표현식){  
    case value1:  
        명령문1;  
        Break;  
  
    case value2:  
        명령문;  
        Break;  
  
    .....  
  
    default  
        명령문n  
  
}
```

제어문과 내장 함수 – FOR 문 - 반복문(1)

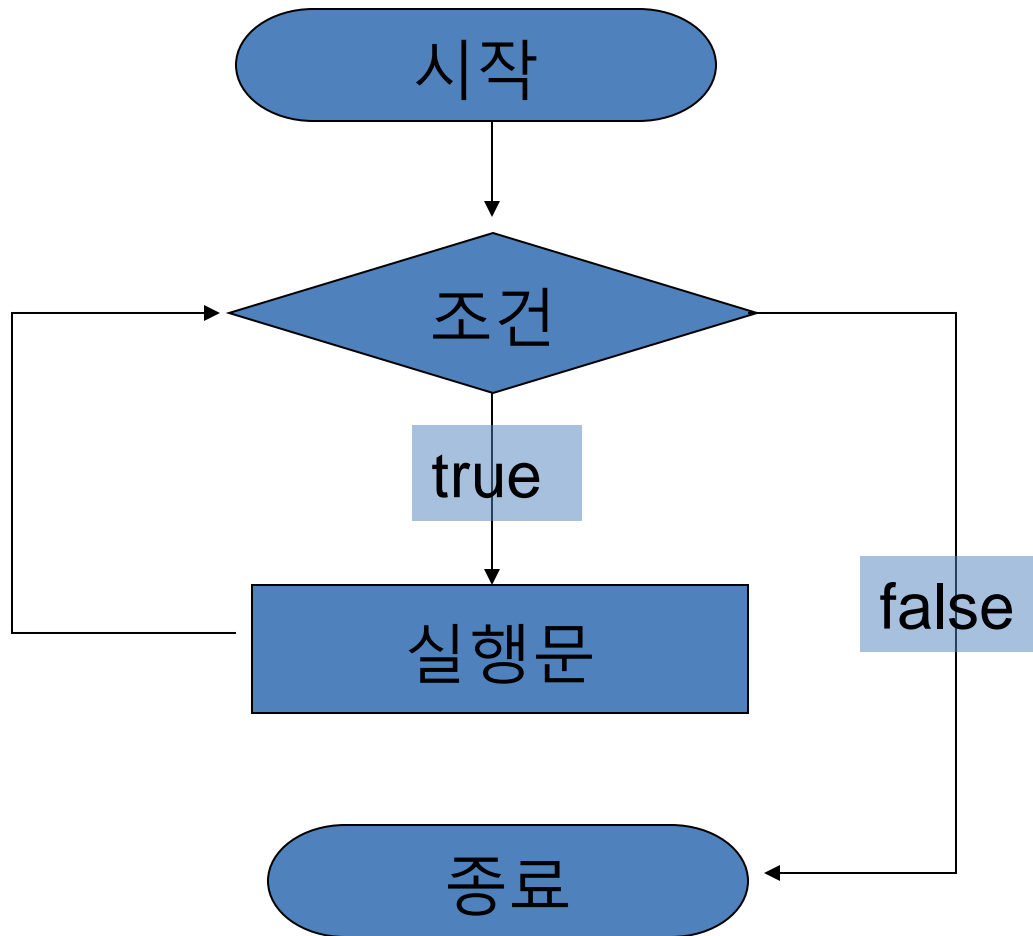
- 기본 형식

```
for(초기 값;조건부;증감식){  
    코드부  
}
```

- 예제

```
for(a=1;a<11;a++){  
    document.write(a+ "*" +a+ "=" +a*a+"<br>")  
}
```

제어문과 내장 함수 – FOR 문 - 반복문(2)



제어문과 내장 함수 – WHILE 문

- 기본 형식

```
while(조건){  
    명령문  
}
```

- 예제

```
a=1  
while(a<11){  
    document.write(a+ "*" +a+ "=" +a*a+"<br>")  
    A++  
}
```

제어문과 내장 함수 – DO WHILE 문

- 기본 형식

```
do{  
    명령문  
} While(조건)
```

- 예제

```
a=1  
do{  
    document.write(a+ "*" +a+ "=" +a*a+"<br>")  
    A++  
} while(a<11)
```

제어문과 내장 함수 – BREAK, CONTINUE 문

- Break : 제어문 종료
- Continue : 제어문 반복
- 예제

A=10

```
While(true){
```

```
    a—
```

```
    if(a>10) continue
```

```
    if(a==0) break
```

```
    document.write(a+ "*" +a+ "=" +a*a+"<br>")
```

```
}
```

제어문과 내장 함수 – RETURN 문

- 함수에서 특정 값을 리턴 값으로 보내고 싶을 때 사용
- 예제

```
Function square(a){  
    return a*a  
}
```

```
For(a=1;a<11;a++){  
    document.write(a+ "*" +a+ "=" +a*a+"<br>")  
}
```

제어문과 내장 함수 – FOR IN– 객체 조작문

- 객체가 가지는 속성 정보를 알려준다.
- 만약 객체의 모든 속성이 5개라면 5번 반복된다.
- 자바 스크립트는 완성된 언어가 아니므로 버전업 되면서 새로 추가된 객체의 속성 정보를 알 수 있다.
- `for (variable in 객체)`
 - {
 - 수행할 작업
 - }

제어문과 내장 함수 – WITH – 객체 조작문

- 하나의 객체에 대해 여러 가지 속성들을 한꺼번에 조작할 때 사용한다.
- with (객체)
{
 조작 내용
}
- with (document)
{
 bgcolor = "white";
 fgcolor = "red";
}

자바 스크립트

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>My First JavaScript</h1>
```

```
<p>
```

JavaScript can react to events. Like the click of a button:

```
</p>
```

```
<button type="button" onclick="alert('Welcome!')">Click Me!</button>
```

```
</body>
```

```
</html>
```

My First JavaScript

JavaScript can react to events. Like the click of a button:



주사위 게임

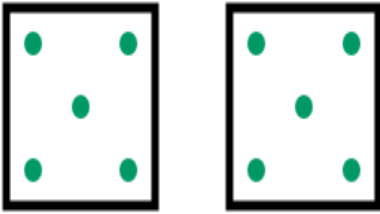
```
<!doctype html>
<html lang="kr">
<head>
  <meta charset="UTF-8">
  <title>주사위 게임</title>
  <script>
    function throwdice() {
      <body>
        <canvas id="canvas" width = "400" height="300">
          이 브라우저는 HTML5의 Canvas 요소를 지원하지 않습니다.
        </canvas>
        <br/>

        <button onClick="throwdice();"> 주사위 던지기 </button>
        <form name="f" id="f">
          판: <input name="stage" value="첫 번째 던지기"/>
          포인트: <input name="pv" value="" />
          결과: <input name="outcome" value="" />
        </form>

      </body>
    }
  </script>
</head>

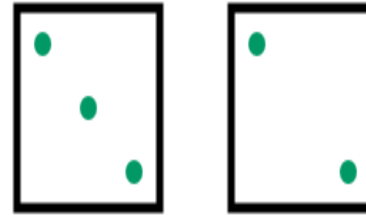
</html>
```

주사위 게임



주사위던지기

판: 다시 던지세요. 포인트: 10 결과:



주사위던지기

판: 새로 시작 포인트: 5 결과: 졌습니다.