

# 제어문 - 조건문 -

# 조건문

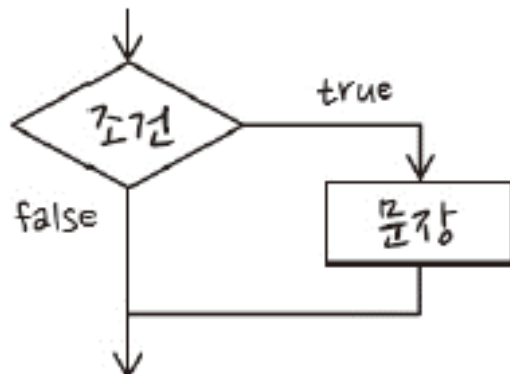
## ❖ if 조건문

- 자바스크립트에서 가장 일반적인 조건문
  - 형태

```
if (불 표현식) {  
    문장  
}
```

- 불리언 표현식이 true면 문장 실행
- false면 문장 무시
  - 조건문에 의해 여러 문장을 실행할 때는 중괄호로 감싸야 함

그림 3-1 if 조건문



# 조건문

---

## ❖ 04\_1\_01\_ifcondition.html

```
<body>
  <script>
    var num = prompt("숫자 하나를 입력하세요", 4);
    if (num % 2 == 0)
      document.write(num + "는 짝수입니다.");
  </script>
</body>
```

# 조건문

---

## ❖ 04\_1\_02\_block.html

```
<body>
  <script>
    var num = prompt("숫자 하나를 입력하세요", 4);
    if (num % 2 == 0) {
      document.write(num + "는 짝수입니다.");
      alert("짝수는 2로 나누어 떨어지는 수입니다.");
    }
  </script>
</body>
```

# 조건문

---

## ❖ 04\_1\_03\_noblock.html

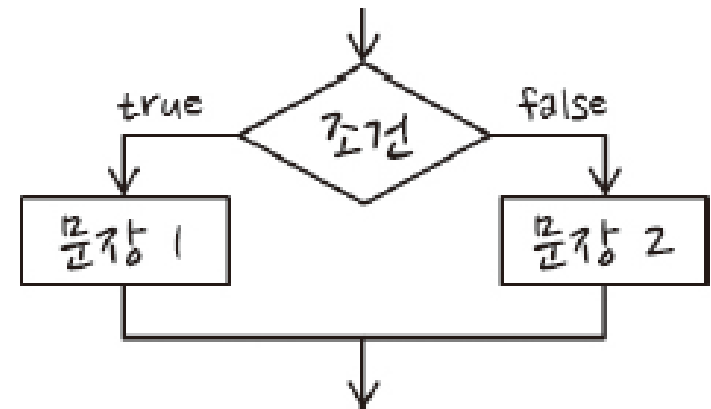
```
<body>
  <script>
    var num = prompt("숫자 하나를 입력하세요", 4);
    if (num % 2 == 0)
      document.write(num + "는 짝수입니다.");
      alert("짝수는 2로 나누어 떨어지는 수입니다.");
  </script>
</body>
```

# 조건문

## ❖ if else 조건문

- 서로 반대되는 조건에 사용하는 조건문
- else 키워드는 if 조건문과 함께 사용하는 키워드
  - if 조건문의 바로 뒤에 붙여 사용
- 조건문의 형태

```
if (불 표현식) {  
    문장 A  
} else {  
    문장 B  
}
```



# 조건문

---

## ❖ 04\_1\_04\_ifelse.html

```
<body>
  <script>
    var num = prompt("숫자 하나를 입력하세요", 4);
    if (num % 2 == 0) {
      document.write(num + "는 짝수입니다.");
    } else {
      document.write(num + "는 홀수입니다.");
    }
  </script>
</body>
```

# 조건문

---

## ❖ 중첩 조건문

- 조건문 안에 조건문을 중첩해 사용하는 형식
- 중첩 조건문의 형태
  - 여러 번 중첩해도 상관 없음

---

```
if (불 표현식) {  
    if (불 표현식) {  
        문장  
    } else {  
        문장  
    }  
} else {  
    if (불 표현식) {  
        문장  
    } else {  
        문장  
    }  
}
```

---



# 조건문

---

## ❖ If else if 조건문

- If else if 조건문의 형태
- 중첩 조건문을 if else if 조건문으로 형태로 변경
  - 한쌍의 중괄호를 삭제

---

```
if (불 표현식) {  
    문장  
} else if (불 표현식) {  
    문장  
} else if (불 표현식) {  
    문장  
} else {  
    문장  
}
```

---

# 조건문

---

## ❖ 04\_1\_05\_ifelseif.html

```
<body>
  <script>
    var num = prompt("숫자 하나를 입력하세요", 2);
    if (num == null) {
      document.write("숫자를 입력해 주십시오.");
    } else {
      if (num > 0) {
        document.write(num + "는 양수입니다.");
      } else if (num < 0) {
        document.write(num + "는 음수입니다.");
      } else {
        document.write(num + "은 틀림없이 0입니다.");
      }
    }
  </script>
</body>
```

# 조건문

---

## ❖ switch 조건문의 기본 형태

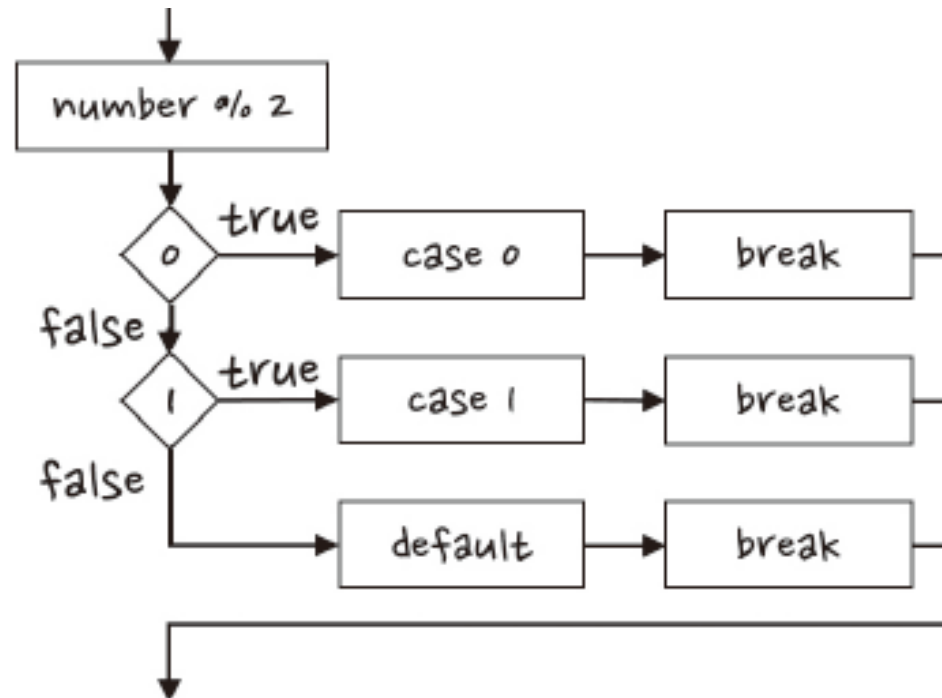
- default 부분은 생략 가능

```
switch(비교값) {  
  case 값:  
    문장  
    break;  
  case 값:  
    문장  
    break;  
  default:  
    문장  
    break;  
}
```

# 조건문

## ❖ break 키워드의 의미

- switch 조건문이나 반복문을 빠져 나가려고 사용하는 키워드
- switch 조건문의 괄호 안에는 비교할 값을 입력
  - 입력한 값을 기준으로 특정 코드 실행
  - 입력한 표현식과 case 키워드 옆의 표현식이 같음
    - case 키워드 바로 다음에 오는 문장 실행



# 조건문

---

## ❖ 04\_1\_06\_switch.html

```
<body>
  <script>
    var menu = prompt("통화는 1번, 문자는 2번, 영상통화는 3번", 1);
    switch (Number(menu)) {
      case 1:
        document.write("전화를 겁니다.");
        break;
      case 2:
        document.write("문자를 보냅니다.");
        break;
      case 3:
        document.write("영상 통화를 연결합니다.");
        break;
      default:
        document.write("잘못 입력하셨습니다.");
        break;
    }
  </script>
</body>
```

## 조건문

---

### ❖ 04\_1\_07\_switchstring.html

```
<body>
  <script>
    var menu = prompt("먹고 싶은 음식을 입력하세요.", "짜장면");
    switch (menu) {
      case "짜장면":
        document.write("수타면으로 준비해 드리겠습니다.");
        break;
      case "짬뽕":
        document.write("시원한 홍합 짬뽕입니다.");
        break;
      case "탕수육":
        document.write("소고기 탕수육입니다.");
        break;
      default:
        document.write("군만두나 드세요.");
        break;
    }
  </script>
</body>
```

## 조건문

### ❖ 04\_1\_08\_fallthrough.html

```
<body>
<script>
  var yoil = prompt("요일을 입력하세요", "월");
  switch (yoil) {
    case "월":
      document.write("일주일의 시작입니다.");
      break;
    case "화":
    case "수":
    case "목":
      document.write("열심히 일해야 할 때입니다.");
      break;
    case "금":
      document.write("불타는 금요일 보내세요.");
      break;
    case "토":
    case "일":
      document.write("편안한 주말 보내세요.");
      break;
  }
</script>
```

# 조건문

---

## ❖ 논리 연산자가 가지고 있는 특성을 조건문으로 사용

- 논리합 연산자를 사용한 표현식은 뒤에 어떠한 값이 들어가도 항상 참
  - `true || ○○○`

```
function fn(a) {  
  a = a || 10;  
  
  :  
}
```



# 조건문

---

## ❖ 논리 연산자를 이용한 짧은 조건문

- 논리합 연산자  
(조건식) || (조건식이 거짓일 때 실행할 문장)
- 논리곱 연산자  
(조건식) && (조건식이 참일 때 실행할 문장)

# 제어문

## - 반복문 -

## 반복문

---

### ❖ 배열이란?

- 여러 개의 변수를 한꺼번에 다룰 수 있는 자료형
- 객체 중 하나
- 대괄호([ ])로 생성/ 대괄호 사용 후 안에 쉼표로 구분해 자료를 입력

```
> var array = [273, 32, 103, 57, 52];  
undefined  
> array  
[ 273, 32, 103, 57, 52 ]
```

## 반복문

---

### ❖ 배열 요소

- 다른 언어에서는 동일한 데이터 타입에 대해서 배열 생성 가능
- 자바스크립트는 어떠한 종류의 자료형도 배열 요소가 될 수 있음
  - 자바스크립트의 모든 자료형을 넣을 수 있음

```
> var array = [  
    273,          // Number  
    'String'      // String  
    ,true,        // Boolean  
    function() {}, // Function  
    {},           // Object  
    [273, 103]    // Array  
];  
  
> array  
[ 273, 'String', true, [Function], {}, [ 273, 103 ] ]
```

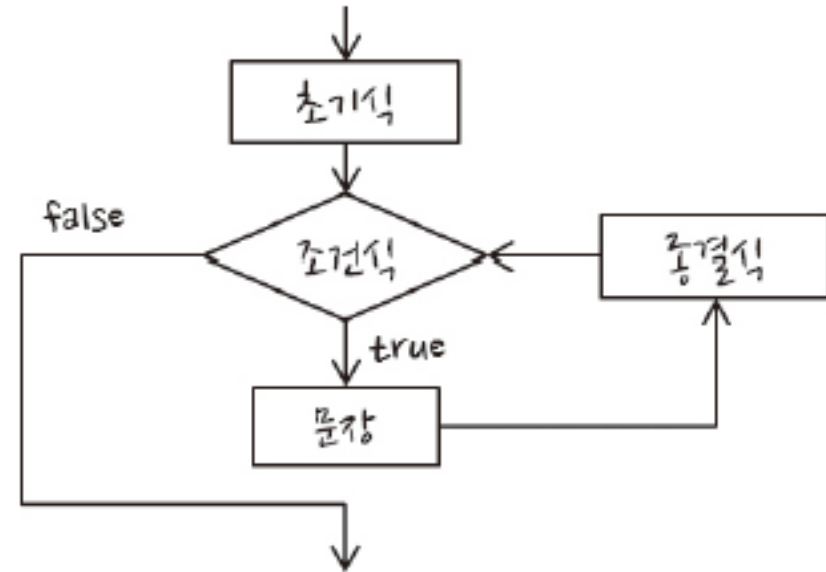
## 반복문

### ❖ for 반복문

- 조건(while 반복문)보다 횟수에 비중을 둘 때 사용 하는 반복문
- 초기식과 종결식이 있음

```
for (초기식; 조건식; 종결식) {  
    문장  
}
```

- 초기식 실행
- 조건식 비교/ 조건이 거짓이면 반복문 종료
- 문장을 실행
- 종결식 실행
- 다시 '조건식 비교' 이후를 반복



# 반복문

---

## ❖ for 반복문 사용

- 초기문에 선언하는 변수는 간단한 한 글자로 만들
- 단순 for 반복문 : `for (var i = 0; i < length; i++ )`와 같은 형태
- 역 for 반복문 : 배열 요소를 역으로 출력

---

```
for (var i = 0; i < length; i++) {  
    문장  
}
```

---

### ❖ 04\_2\_01\_forloop.html

```
<body>
  <script>
    for (var i = 0; i < 5; i++) {
      document.write("좋은 아침입니다.<br>");
    }
  </script>
</body>
```

### ❖ 04\_2\_01\_forloop2.html

```
<body>
  <script>
    var arScore = [88, 78, 96, 54, 23];
    for (var st = 0; st < 5; st++) {
      document.write(st + "번째 학생의 성적 : " + arScore[st] +
" <br>");
    }
  </script>
</body>
```



### ❖ 04\_2\_01\_forloop3.html

```
<body>
  <script>
    var sum = 0;
    for (var i = 1; i <= 100; i++) {
      sum = sum + i;
    }
    document.write("1~100까지의 합 = " + sum);
  </script>
</body>
```

### ❖ 04\_2\_01\_forloop4.html

```
<body>
  <script>
    var arScore = [88, 78, 96, 54, 23];
    var sum = 0;
    for (var st = 0; st < arScore.length; st++) {
      sum += arScore[st];
    }
    document.write("총점 : " + sum +
                  ", 평균 : " + sum/arScore.length);
  </script>
</body>
```

## 반복문

---

### ❖ for in 반복문의 형태

- for in 반복문은 단순 for 반복문과 같은 기능

---

```
for (var i in array) {  
  
}
```

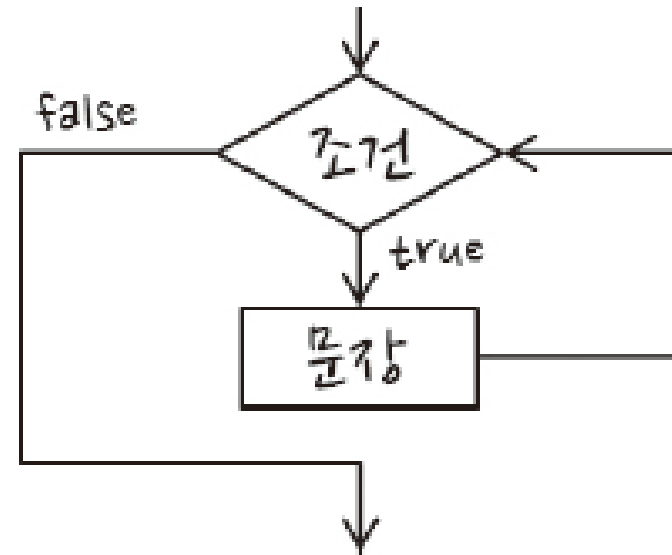
---

# 반복문

## ❖ while 반복문

- 가장 간단한 반복문
- if 조건문 형태가 비슷

```
while (불 표현식) {  
    문장  
}
```



# 반복문

---

## ❖ while 반복문

- while 반복문 종료
- 숫자 증가
- 내부적으로 변화

```
var value=0;

while(value<5){
    value++;
}
```

### ❖ 04\_2\_02\_while.html

```
<body>
  <script>
    var size = 1024;
    var upload = 0;
    while (upload <= size) {
      upload += 200;
      document.write(upload + "M 업로드중....<br>");
    }
    document.write("업로드를 완료하였습니다");
  </script>
</body>
```

### ❖ 04\_2\_03\_whilesum.html

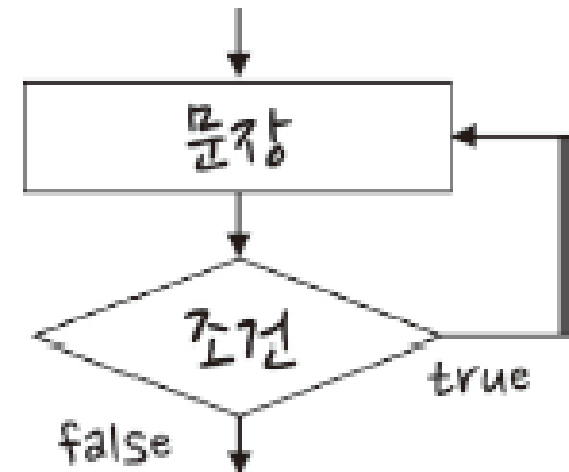
```
<body>
  <script>
    var sum = 0;
    var i = 1;
    while (i <= 100) {
      sum = sum + i;
      i++;
    }
    document.write("1~100까지의 합 = " + sum);
  </script>
</body>
```

## 반복문

### ❖ do While 반복문

- 조건의 참 거짓 여부와 상관없이 내부의 문장을 최소한 한 번은 실행해야 하는 경우 사용
- while 반복문과 형태가 비슷
- do while 반복문의 기본 형태

```
do {  
    문장  
} while (불 표현식);
```





### ❖ 04\_2\_04\_dowhile.html

```
<body>
  <script>
    var sum = 0;
    var num;
    do {
      num = prompt("숫자를 입력하세요(끝낼 때는 0)", "2");
      sum += Number(num);
    } while (num != 0);
    document.write("입력한 모든 숫자의 합 = " + sum);
  </script>
</body>
```

## 반복문

---

### ❖ 다중 루프

### ❖ 04\_2\_05\_nestarray.html

```
<body>
  <script>
    var ar = [[0, 1, 2, 3], [4, 5, 6], [7, 8]];
    for (var i = 0; i < ar.length; i++) {
      for (var j = 0; j < ar[i].length; j++) {
        document.write("ar[" + i + "][" + j + "] =" + ar[i][j] +
          "<br>");
      }
      document.write("<br>");
    }
  </script>
</body>
```

### ❖ 04\_2\_06\_gugudan.html

```
<body>
  <script>
    for (var i = 1; i <= 9; i++) {
      for (var j = 1; j <= 9; j++) {
        document.write(i + " * " + j + " = " + i*j + "<br>");
      }
      document.write("<br>");
    }
  </script>
</body>
```

### ❖ 04\_2\_07\_triangle.html

```
<body>
  <script>
    for (var y = 1; y < 10; y++) {
      for (var x = 0; x < y; x++) {
        document.write("*");
      }
      document.write("<br>");
    }
  </script>
</body>
```

### ❖ 04\_2\_08\_twocontrol.html

```
<body>
  <script>
    for (var i = 0, j = 5; i < 5; i++, j += 2) {
      document.write("i = " + i + ", j = " + j + "<br>");
    }
  </script>
</body>
```

### ❖ 04\_2\_08\_twocontrol2.html

```
<body>
  <script>
    var i = 0, j = 5;
    while (i < 5) {
      document.write("i = " + i + ", j = " + j + "<br>");
      i++;
      j+=2;
    }
  </script>
</body>
```

# 반복문

---

## ❖ break 키워드

- switch 조건문이나 반복문을 벗어날 때 사용
- 다음 반복문은 조건이 항상 참으로 무한 반복
- 무한루프는 break 키워드 사용으로 탈출

---

```
while (true) {  
  
}
```

---



### ❖ 04\_2\_09\_break.html

```
<body>
  <script>
    var arScore = [88, 78, 96, -54, 23];
    for (var st = 0; st < 5; st++) {
      if (arScore[st] < 0 || arScore[st] > 100) break;
      document.write(st + "번째 학생의 성적 : " + arScore[st] +
" <br>");
    }
  </script>
</body>
```

# 반복문

---

## ❖ continue 키워드

- 현재 반복을 멈추고 다음 반복을 진행

```
for(var i=0; i < 5; i++) {  
    continue;  
    console.log(i);  
}
```

### ❖ 04\_2\_10\_continue.html

```
<body>
  <script>
    var arScore = [88, 78, 96, null, 23];
    for (var st = 0; st < 5; st++) {
      if (arScore[st] == null) continue;
      document.write(st + "번째 학생의 성적 : " + arScore[st] +
                     "<br>");
    }
  </script>
</body>
```

## 반복문

---

### ❖ break 라벨

### ❖ 04\_2\_11\_breaklabel.html

```
<body>
  <script>
    for (var i = 0; i < 3; i++) {
      for (var j = 0; j < 5; j++) {
        document.write("i = " + i + ", j = " + j + "<br>");
        if (i == 1 && j == 2) break;
      }
      document.write("<br>");
    }
  </script>
</body>
```

### ❖ 04\_2\_11\_breaklabel2.html

```
<body>
  <script>
    iloop:
    for (var i = 0; i < 3; i++) {
      jloop:
      for (var j = 0; j < 5; j++) {
        document.write("i = " + i + ", j = " + j + "<br>");
        if (i == 1 && j == 2) break iloop;
      }
      document.write("<br>");
    }
  </script>
</body>
```

## 반복문

---

### ❖ 무한 루프

```
for (;;) { }
```

```
while (true) { }
```

### ❖ 04\_2\_12\_infinite.html

```
<body>
  <script>
    var grade;
    for (;;) {
      grade = Number(prompt("학년을 입력하세요(1~6).", "1"));
      if (grade >= 1 && grade <= 6) break;
      alert("잘못 입력하셨습니다. 다시 입력해 주세요");
    }
    document.write(grade + "학년을 입력했습니다.");
  </script>
</body>
```



### ❖ 04\_2\_13\_mincommon.html

```
<body>
  <script>
    var num1 = Number(prompt("첫 번째 숫자를 입력하세요.", "6"));
    var num2 = Number(prompt("두 번째 숫자를 입력하세요.", "8"));

    var common = 1;
    for (;;) {
      if (common % num1 == 0 && common % num2 == 0) {
        break;
      }
      common++;
    }
    document.write("최소 공배수 = " + common);
  </script>
</body>
```

# 제어문

## - 예외처리 -

# 예외처리

---

## ❖ 예외처리

```
try {  
    명령;  
} catch(exception) {  
    예외 발생시의 명령  
} finally {  
    정리 코드  
}
```

# 예외처리

---

## ❖ 04\_3\_01\_exception.html

```
<body>
  <script>
    document.write("실행을 시작합니다.<br>");
    document.message("메시지입니다.<br>");
    document.write("실행을 완료하였습니다.<br>");
  </script>
</body>
```

## 예외처리

---

### ❖ 04\_3\_02\_ifexception.html

```
<body>
  <script>
    document.write("실행을 시작합니다.<br>");
    document.message("메시지입니다.<br>");
    document.write("a<body>
  <script>
    document.write("실행을 시작합니다.<br>");
    if (document.message) {
      document.message("메시지입니다.<br>");
    }
    document.write("실행을 완료하였습니다.<br>");
  </script>
</body>실행을 완료하였습니다.<br>");
  </script>
</body>
```

## 예외처리

---

### ❖ 04\_3\_03\_trycatch.html

```
<body>
  <script>
    document.write("실행을 시작합니다.<br>");
    try {
      document.message("메시지입니다.<br>");
    } catch(exception) {
      alert("예외가 발생했습니다.");
    }
    document.write("실행을 완료하였습니다.<br>");
  </script>
</body>
```

## 예외처리

---

### ❖ 예외 객체

속성	설명
name	예외의 이름
message	예외에 대한 짧은 메시지
description	예외에 대한 긴 설명

# 예외처리

---

## ❖ 04\_3\_04\_exceptionobject.html

```
<body>
  <script>
    document.write("실행을 시작합니다.<br>");
    try {
      document.message("메시지입니다.<br>");
    } catch(exception) {
      alert("이름 : " + exception.name +
        "\n메시지 : " + exception.message +
        "\n설명 : " + exception.description);
    }
    document.write("실행을 완료하였습니다.<br>");
  </script>
</body>
```



# 예외처리

---

## ❖ 04\_3\_05\_throw.html

```
<body>
  <script>
    // 라이브러리 내부
    function func() {
      if (true) throw "예외가 발생했습니다";
    }

    // 라이브러리 사용
    try {
      func();
    } catch(exception) {
      alert("이름 : " + exception.name +
        "\n메시지 : " + exception.message +
        "\n설명 : " + exception.description);
    }
    document.write("실행을 완료하였습니다.<br>");
  </script>
</body>
```