

3. 연산자, 제어문

3-1. 연산자의 종류

3-2. 조건문

3-3. 반복문

■ Operator

- 문자 연산자
- 산술 연산자
- 증가 / 감소 연산자
- 대입 연산자
- 논리 연산자
- 비교 연산자

■ Operator

● 문자 연산자 (+ +=)

03-01-문자연산.html

```
console.log('my' + ' cat');  
console.log('1' + 2);  
console.log(`string literals: 1 + 2 = ${1 + 2}`);  
console.log(`string literals: 줄바꿈  
지원`);
```

my cat

12

string literals: 1 + 2 = 3

string literals: 줄바꿈
지원

● 산술 연산자 (+ - * / % **)

03-02-산술연산.html

```
console.log(1 + 1); // 더하기  
console.log(1 - 1); // 빼기  
console.log(1 / 1); // 나누기  
console.log(1 * 1); // 곱하기  
console.log(5 % 2); // 나머지  
console.log(2 ** 3); // 거듭제곱
```

2

0

1

1

1

8

■ Operator

● 증가 연산자 (전위)

03-03-증가연산.html

```
let counter = 2;
const preIncrement = ++counter;
// counter = counter + 1;
// preIncrement = counter;
console.log(`preIncrement: ${preIncrement}`);
console.log(`counter: ${counter}`);
```

preIncrement: 3

counter: 3

● 증가 연산자 (후위)

```
const postIncrement = counter++;
// postIncrement = counter;
// counter = counter + 1;
console.log(`postIncrement: ${postIncrement}`);
console.log(`counter: ${counter}`);
```

postIncrement: 3

counter: 4

■ Operator

● 감소 연산자 (전위)

03-04-감소연산.html

```
let counter = 4;  
const preDecrement = --counter;  
// counter = counter - 1;  
// preDecrement = counter;  
console.log(`preDecrement: ${preDecrement}`);  
console.log(`counter: ${counter}`);
```

preDecrement: 3

counter: 3

● 감소 연산자 (후위)

```
const postDecrement = counter--;  
// postDecrement = counter;  
// counter = counter - 1;  
console.log(`postDecrement: ${postDecrement}`);  
console.log(`counter: ${counter}`);
```

postDecrement: 3

counter: 2

■ Operator

● 대입 연산자 (= += -= *= /= %= **=)

03-05-대입연산.html

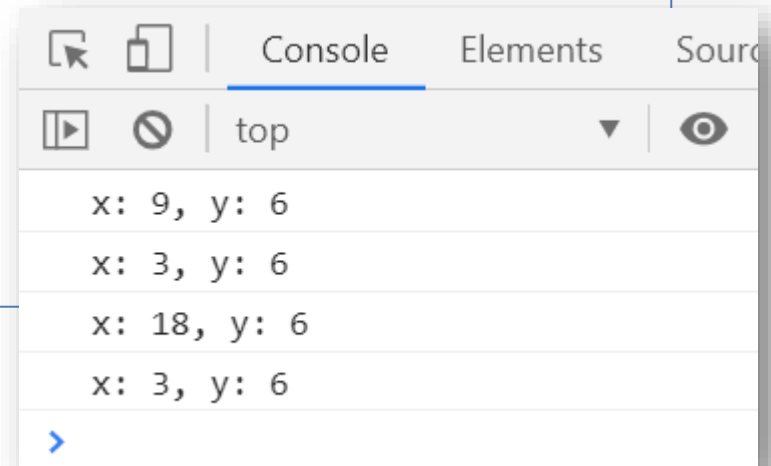
```
let x = 3;
let y = 6;

x += y; // x = x + y;
console.log(`x: ${x}, y: ${y}`);

x -= y; // x = x - y;
console.log(`x: ${x}, y: ${y}`);

x *= y; // x = x * y;
console.log(`x: ${x}, y: ${y}`);

x /= y; // x = x / y;
console.log(`x: ${x}, y: ${y}`);
```



■ Operator

● 논리 연산자 (&& || !)

03-06-논리연산.html

```
const value1 = false;
const value2 = 4 < 2;

// true를 찾으면 true 반환하며 종료
console.log(`or: ${value1 || value2 || check()}`);

// false를 찾으면 false 반환하며 종료
console.log(`and: ${value1 && value2 && check()}`);

function check() {
  for(let i = 0; i < 10; i++) {
    console.log('pass');
  }
  return true;
}

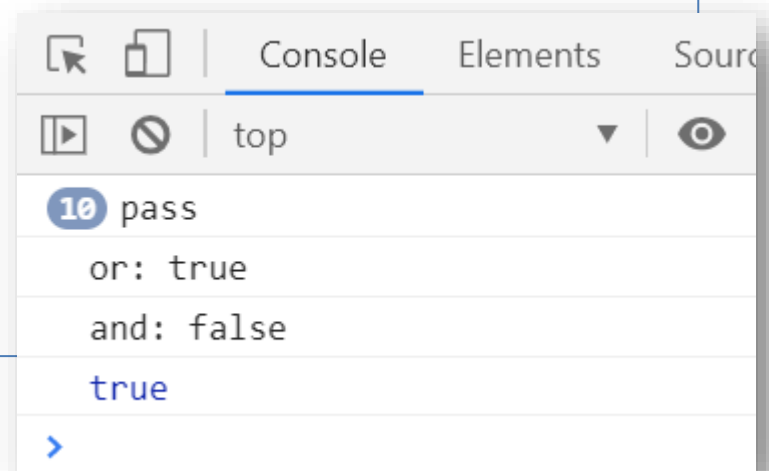
console.log(!value1);
```

논리곱 (&)

A	B	결과
0	0	0
0	1	0
1	0	0
1	1	1

논리합 (|)

A	B	결과
0	0	0
0	1	1
1	0	1
1	1	1



■ Operator

● 비교 연산자 (== === != !== > < >= <=)

03-07-비교연산.html

```
console.log(`10 < 6 -> ${10 < 6}`);  
console.log(`10 <= 6 -> ${10 <= 6}`);  
console.log(`10 > 6 -> ${10 > 6}`);  
console.log(`10 >= 6 -> ${10 >= 6}`);
```

10 < 6 -> false
10 <= 6 -> false
10 > 6 -> true
10 >= 6 -> true

```
const stringFive = '5';  
const numberFive = 5;  
  
// 느슨한 비교 (자료형 자동 변환)  
console.log(stringFive == numberFive);  
console.log(stringFive != numberFive);  
  
// 엄격한 비교  
console.log(stringFive === numberFive);  
console.log(stringFive !== numberFive);
```

true
false
false
true

■ Operator

● 비교 연산자 (== === != !== > < >= <=)

03-07-비교연산.html

```
const dev1 = { name: 'dev' };  
const dev2 = { name: 'dev' };  
const dev3 = dev1;
```

```
console.log(dev1 == dev2);  
console.log(dev1 === dev2);  
console.log(dev1 === dev3);
```

false

false

true

● 삼항 연산자 (? :)

03-08-삼항연산.html

```
let user_id = 'script';  
console.log(user_id === 'script' ? 'yes' : 'no');
```

```
let score = 70;  
console.log(  
  score >= 60 ?  
    (score >= 90 ? 'perfect' : 'good') : 'bad');
```

yes

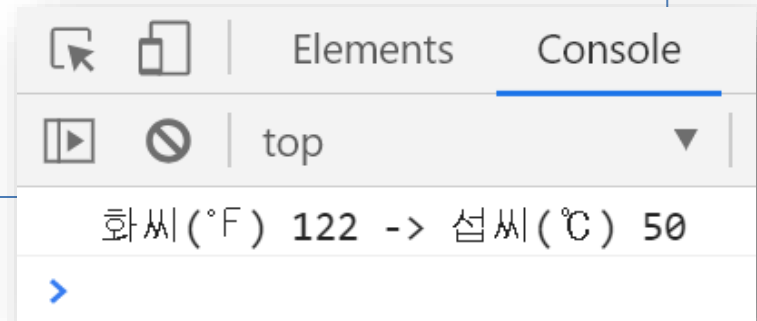
good

■ 연습문제 (03-연습문제1.html)

● 화씨(°F)를 섭씨(°C)로 변환하기

- 공식 : $^{\circ}\text{F} = (^{\circ}\text{C} \times \frac{9}{5}) + 32$

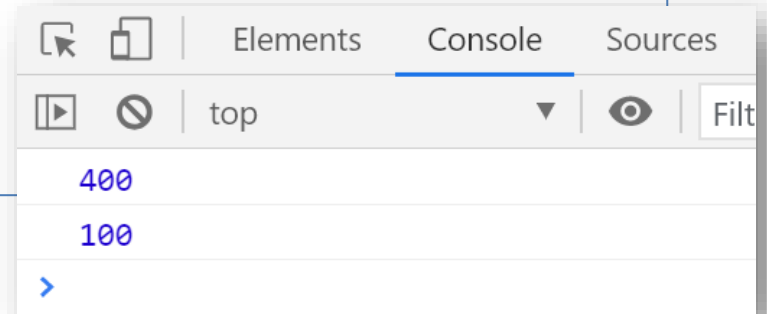
```
let fTemp = 122;  
let cTemp = ?
```



■ 연습문제 (03-연습문제2.html)

- 백의 자리 이하 버리기 (456 → 400, 111 → 100)

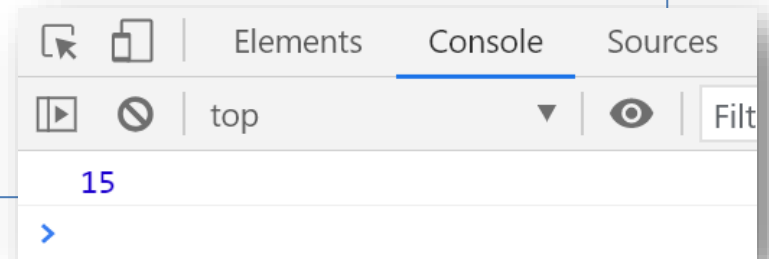
```
let num1 = 456;  
let num2 = 111;
```



■ 연습문제 (03-연습문제3.html)

- 각 자리 숫자들의 합 구하기 (123 → 6, 745 → 16)

```
let num = 12345;
```



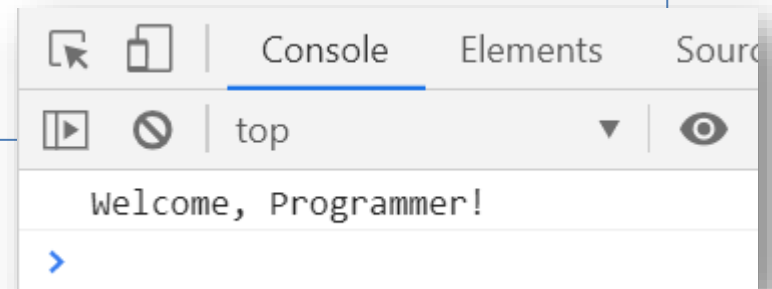
■ 제어문

● if / else if / else

03-09-if.html

```
const job = 'programmer';

if(job === 'programmer') {
  console.log('Welcome, Programmer!');
} else if (job === 'designer') {
  console.log('Good!');
} else {
  console.log('Unknown');
}
```



■ 제어문

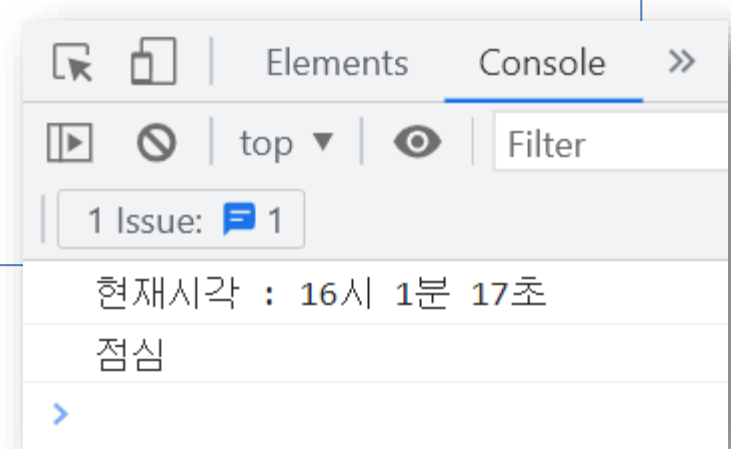
● 현재 시각에 따라 아침 / 점심 / 저녁 출력

03-10-time.html

```
const date = new Date();
let hour = date.getHours();
let min = date.getMinutes();
let sec = date.getSeconds();

console.log("현재시각 : " + hour + "시 " + min + "분 " + sec + "초");

if (hour < 12) {
  console.log("아침");
} else if (hour < 18) {
  console.log("점심");
} else {
  console.log("저녁");
}
```



■ 제어문

● 현재 시각에 따라 배경이미지 변경

03-11-배경이미지.html

```
const date = new Date();
let hour = date.getHours();
let min = date.getMinutes();
let sec = date.getSeconds();

console.log("현재시각 : " + hour + "시 " + min + "분 " + sec + "초");

if (hour < 12) {
  document.body.style.backgroundImage = 'url(images/morning.webp)';
} else if (hour < 18) {
  document.body.style.backgroundImage = 'url(images/afternoon.jpg)';
} else {
  document.body.style.backgroundImage = 'url(images/evening.jpg)';
}
```

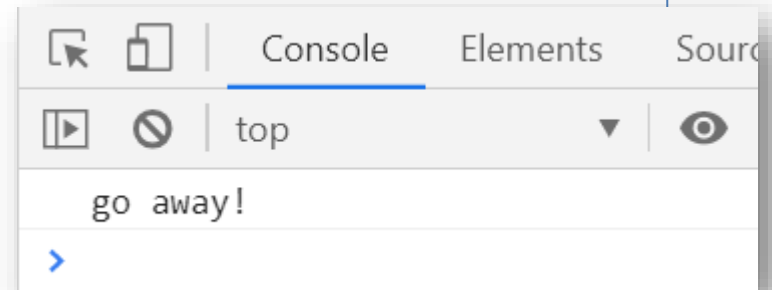


■ 제어문

● switch

03-12-switch.html

```
const browser = 'IE';
switch(browser) {
  case 'IE':
    console.log('go away!');
    break;
  case 'Chrome':
  case 'Firefox':
    console.log('love you!');
    break;
  default:
    console.log('same all!');
}
```



■ 제어문

● 입력된 숫자 홀/짝 구분하기

03-13-switch.html

```
let input = Number(prompt('숫자를 입력해주세요', '숫자'));
switch (input % 2) {
  case 0:
    alert('짝수입니다. ');
    break;
  case 1:
    alert('홀수입니다. ');
    break;
  default:
    alert('숫자만 입력해주세요. ');
}
```

127.0.0.1:5500 내용:

숫자를 입력해주세요

숫자

확인

취소

127.0.0.1:5500 내용:

홀수입니다.

확인

■ 제어문

● 클라이언트의 웹브라우저 종류 구분하기 (1 / 2)

03-14-userAgent.html

```
<img src="" style="width:300px;">
<script src="js/03-14-userAgent.js"></script>
<script>
  const client = getClient();
  const img = document.querySelector('img');
  switch(client) {
    case 'AndroidPhone':
      img.src = 'images/android.png';
      break;
    case 'iPhone':
      img.src = 'images/iphone.svg';
      break;
    case 'iPad':
      img.src = 'images/ipad.jpg';
      break;
    case 'Chrome':
      img.src = 'images/chrome.png';
      break;
    default:
      img.src = 'images/etc.webp';
  }
</script>
```

■ 제어문

● 클라이언트의 웹브라우저 종류 구분하기 (2 / 2)

03-14-userAgent.js

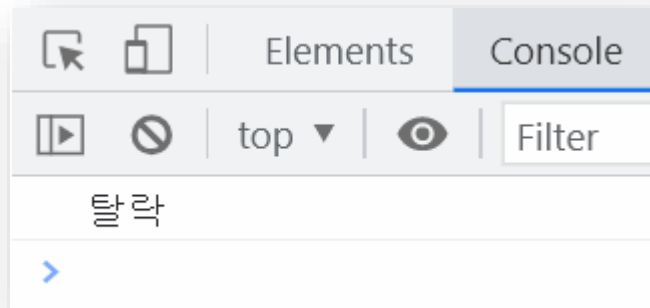
```
function getClient() {  
  const userAgent = navigator.userAgent;  
  console.log(userAgent);  
  
  let os = '';  
  
  if(userAgent.indexOf('Android') > -1) {  
    os = 'AndroidPhone';  
  } else if(userAgent.indexOf('iPhone') > -1) {  
    os = 'iPhone';  
  } else if(userAgent.indexOf('iPad') > -1) {  
    os = 'iPad';  
  } else if(userAgent.indexOf('Chrome') > -1) {  
    os = 'Chrome';  
  } else {  
    os = 'ETC'  
  }  
  
  return os;  
}
```

■ 연습문제 (03-연습문제4.html)

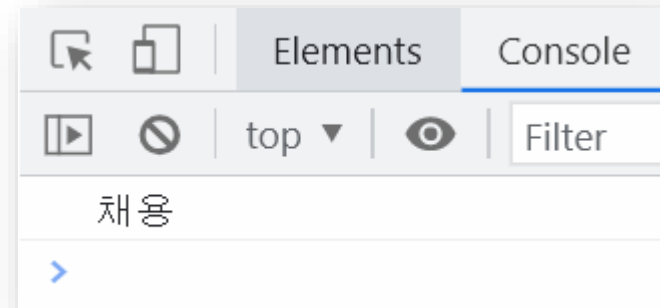
- 중첩 if를 수정하여 하나의 if만 사용하도록 수정하기 (논리연산자 사용)

```
let score = 50;
let hasPortfolio = true;
let isPass = true;
if(score >= 60) {
  if(hasPortfolio === true || isPass === true) {
    console.log('채용');
  } else {
    console.log('탈락');
  }
} else {
  console.log('탈락');
}
```

score : **50**, hasPortfolio : **true**, isPass : **true**



score : **60**, hasPortfolio : **true**, isPass : **true**



■ 연습문제 (03-연습문제5.html)

- 가위바위보 게임의 결과를 출력 예시와 같이 출력하기

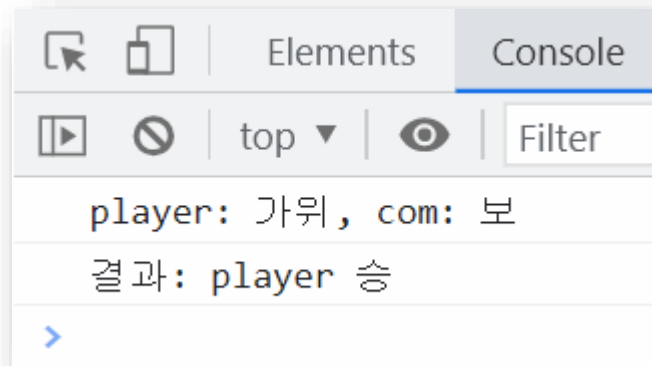
```
const map = ['가위', '바위', '보'];
const player = 0;
const com = parseInt(Math.random() * 3);
console.log(`player: ${map[player]}, com: ${map[com]}`);

let result = '';

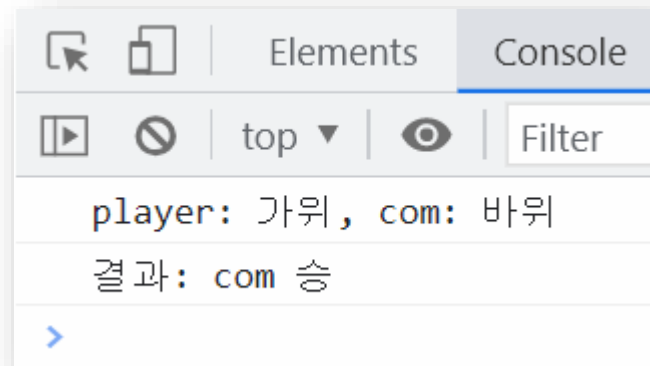
/* 코드 */

console.log(`결과: ${result}`);
```

출력 예시 1)



출력 예시 2)



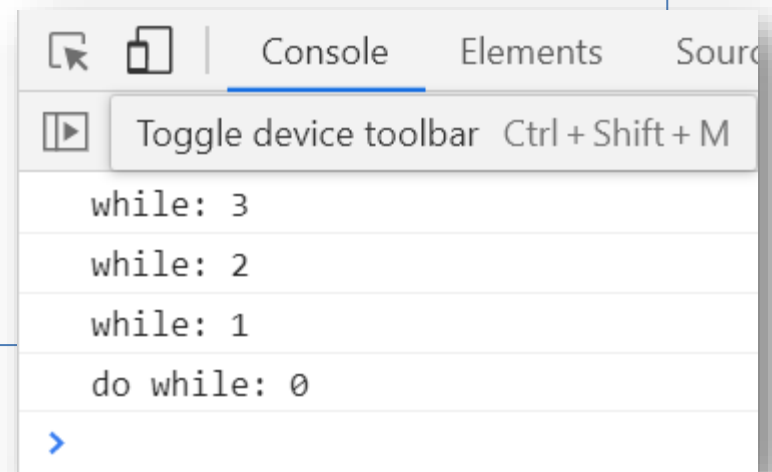
■ 제어문

● while

03-15-while.html

```
let i = 3;
while(i > 0) {
  console.log(`while: ${i}`);
  i--;
}

do {
  console.log(`do while: ${i}`);
  i--;
} while(i > 0);
```



■ 제어문

● 1 에서 10 사이의 랜덤 숫자 맞추기

03-16-while.html

```
const input = parseInt( Math.random() * 10 + 1 );
let isContinue = true;
while (isContinue) {
  let number = prompt("숫자를 입력하세요.");
  if (input == number) {
    alert("정답");
    isContinue = false;
  } else {
    if (input > number) {
      alert("입력된 숫자가 더 적음");
    } else {
      alert("입력된 숫자가 더 큼");
    }
  }
}
```

127.0.0.1:5500 내용:

숫자를 입력하세요.

확인

취소

127.0.0.1:5500 내용:

입력된 숫자가 더 적음

확인

127.0.0.1:5500 내용:

정답

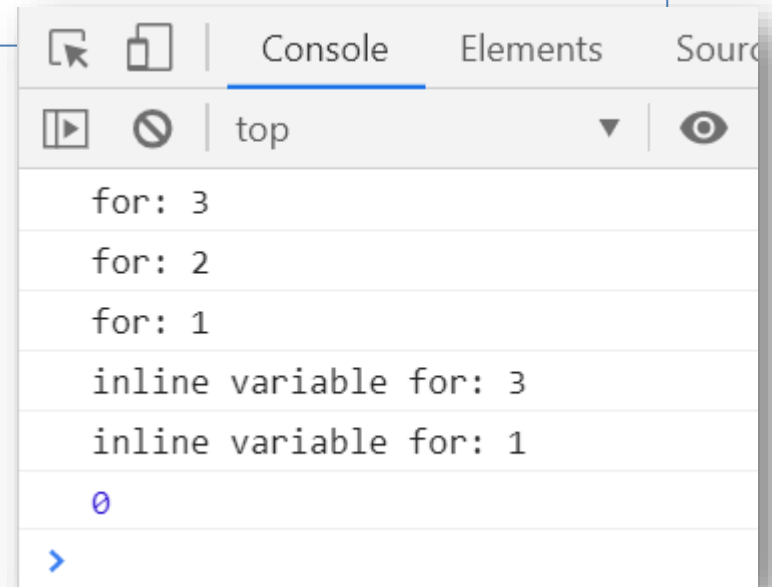
확인

■ 제어문

● for

03-17-for.html

```
let i;  
for(i = 3; i > 0; i--) { // for(begin; condition; step)  
  console.log(`for: ${i}`);  
}  
  
for(let i = 3; i > 0; i = i - 2) {  
  console.log(`inline variable for: ${i}`);  
}
```

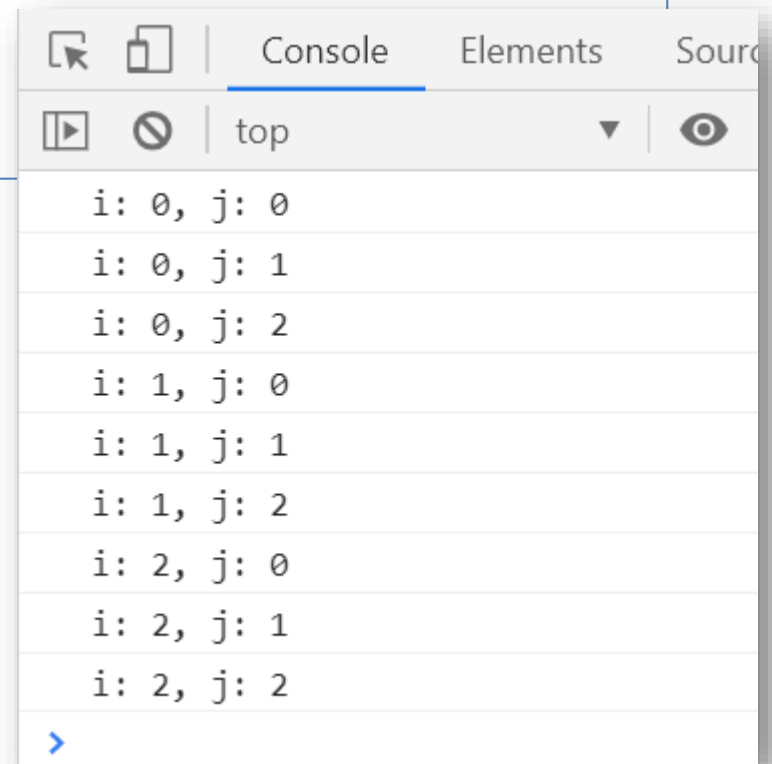


■ 제어문

● for (중첩)

03-18-for.html

```
for(let i = 0; i < 3; i++) {  
  
    for (let j = 0; j < 3; j++) {  
        console.log(`i: ${i}, j: ${j}`);  
    }  
  
}
```



■ 제어문

● 별 출력하기

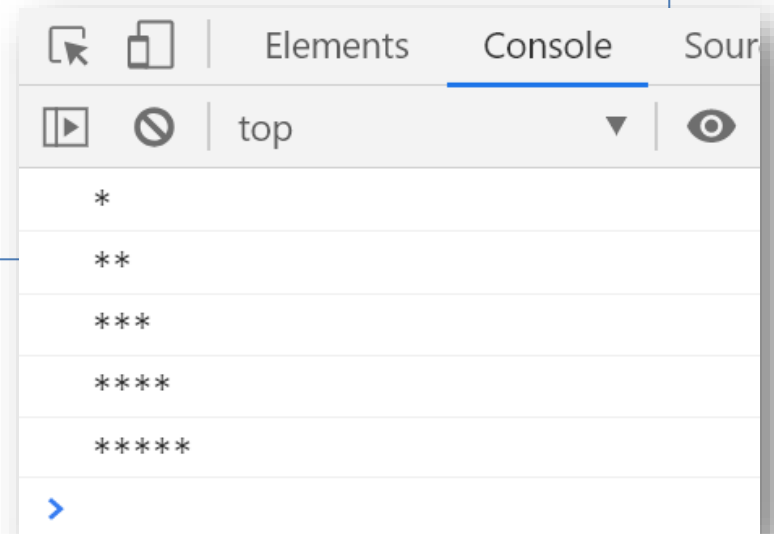
03-19-별출력.html

```
let count = 5;
for (let i = 1; i <= count; i++) {

    let temp = '';

    for (let j = 1; j <= i; j++) {
        temp += '*';
    }

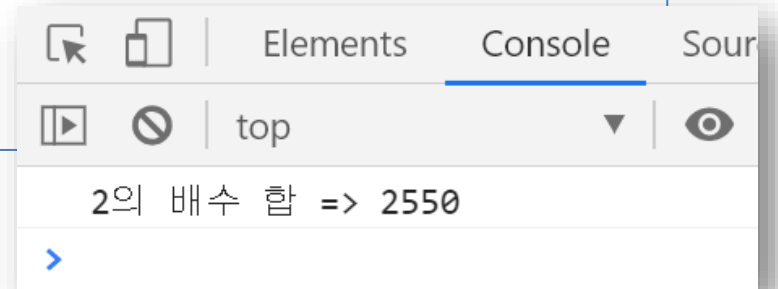
    console.log(temp);
}
```



■ 연습문제 (03-연습문제6.html)

- 1 에서 100 사이의 수 중 2의 배수 합 구하기

```
let total = 0;
```



■ 무작위 번호 맞추기

1. 중복되지 않는 3개의 번호 추출

03-20-무작위번호.html

```
const selected = [];  
  
while(selected.length < 3) {  
  const num = parseInt(Math.random() * 45) + 1;  
  if(selected.indexOf(num) == -1) {  
    selected.push(num);  
  }  
}
```

▶ (3) [28, 21, 24]

>

▶ (3) [1, 16, 8]

>

■ 무작위 번호 맞추기

2. 추출된 번호 정렬하기

03-20-무작위번호.html

```
selected.sort((a, b) => a - b);
```

▶ (3) [28, 21, 24]

>



▶ (3) [21, 24, 28]

>

▶ (3) [1, 16, 8]

>



▶ (3) [1, 8, 16]

>

■ 무작위 번호 맞추기

3. 제시된 번호와 비교하기

03-20-무작위번호.html

```
const lotto = [1, 8, 16];  
if(lotto == selected.toString()) {  
  console.log('맞히기 성공');  
}
```

▶ (3) [21, 24, 28]

맞히기 성공



▶ (3) [1, 8, 16]

맞히기 성공



■ 연습문제 (03-연습문제7.html)

- "무작위 번호 맞추기"의 코드를 활용하여

제시된 번호를 맞힐 때까지 시도한 횟수를 결과와 같이 출력하기

(중첩 반복문 사용)

```
const lotto = [1, 8, 16];  
let cnt = 0;  
  
/* 코드 */  
  
console.log(`1등 맞추기 시도 횟수: ${cnt}`);
```

결과

▶ (3) [19, 32, 39]

▶ (3) [14, 23, 42]

▶ (3) [28, 33, 45]

▶ (3) [9, 19, 33]

▶ (3) [1, 8, 16]

맞히기 성공

시도 횟수: 18488