

for문 반복문(loop) & 함수(fucntion)

```
<title>
  반복문, for문
  for(초깃값; 조건식; 증감식){
    실행문;
  }
</title>
```

◆ JavaScript for 문 기본 정리

1 구조

```
for (초깃값; 조건식; 증감식) {
  실행문;
}
```

각 부분 설명

부분	역할
초깃값	반복문이 시작될 때 초기값을 설정 (주로 카운터 변수)
조건식	반복을 계속할지 판단. 조건식이 <code>true</code> 인 동안 반복
증감식	반복이 끝날 때마다 변수 값을 증가/감소 시킴
실행문	조건이 참일 때 반복 실행할 코드 블록

2 흐름

1. 초깃값 실행
2. 조건식 평가 → **true**이면 실행문 수행, **false**이면 종료

3. 실행문 끝나면 증감식 수행

4. 2~3 반복

3 예제 1 — 1부터 5까지 출력

```
for (var i = 1; i <= 5; i++) {  
  console.log(i);  
}
```

출력 결과 (콘솔)

```
1  
2  
3  
4  
5
```

- 초기값: `i = 1`
- 조건식: `i <= 5` → 참일 때 반복
- 증감식: `i++` → 1씩 증가

4 예제 2 — 10부터 1까지 역순 출력

```
for (var i = 10; i >= 1; i--) {  
  console.log(i);  
}
```

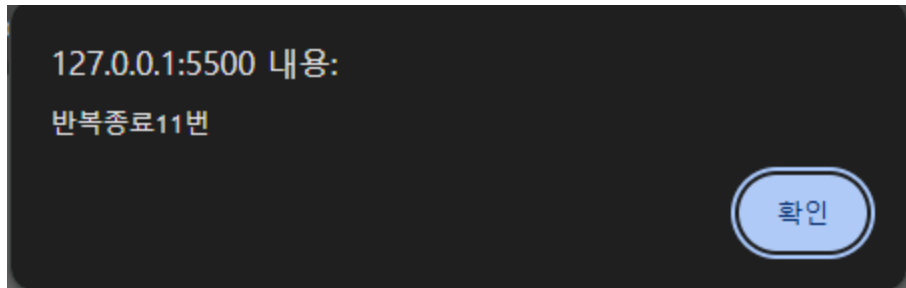
출력 결과

10
9
8
7
6
5
4
3
2
1

- 증감식을 `i--` 로 바꿔서 감소

💀 1. 실습 `i <= 10` 자바스크립트 문구 10번 출력후 10번 출력 이후 경고창
"종료"

자바스크립트: 1
자바스크립트: 2
자바스크립트: 3
자바스크립트: 4
자바스크립트: 5
자바스크립트: 6
자바스크립트: 7
자바스크립트: 8
자바스크립트: 9
자바스크립트: 10



```
for (var i = 1; i <= 10; i++) {  
    document.write("자바스크립트: " + i + "<br>");  
}  
alert("반복종료" + i + "번");
```

💀 2. 실습 for문 이용하여 1부터 10까지의 합을 구하여 [도큐먼트] 출력

1부터 10까지의 합은55 입니다.

```
// [도큐먼트] 1부터 10까지의 합은 ... 입니다.  
var i = 0;  
var num = 0;  
for(i = 1; i <=10; i++){  
    num += i;  
}  
  
document.write( "1부터 10까지의 합은"+ num +" 입니다.");
```

※ 해석 i 는 초깃값 0 을 갖으며, for문 파라미터로 들어와 1 이 되며 10까지 **증감**하며 돌며,
기존 초기 num = 0 에다가 += (복합대입연산하고) num을 for문 밖으로 빼서 document.write 를 출력 하면 된다.

복합대입연산자 = 하나의 변수에 데이터 누적하기 위함 `num += i;`

💀 3.실습 <h1>~<h6>순으로 풀스택개발자 [도큐먼트] 1부터 6까지의 출력

풀스택개발자

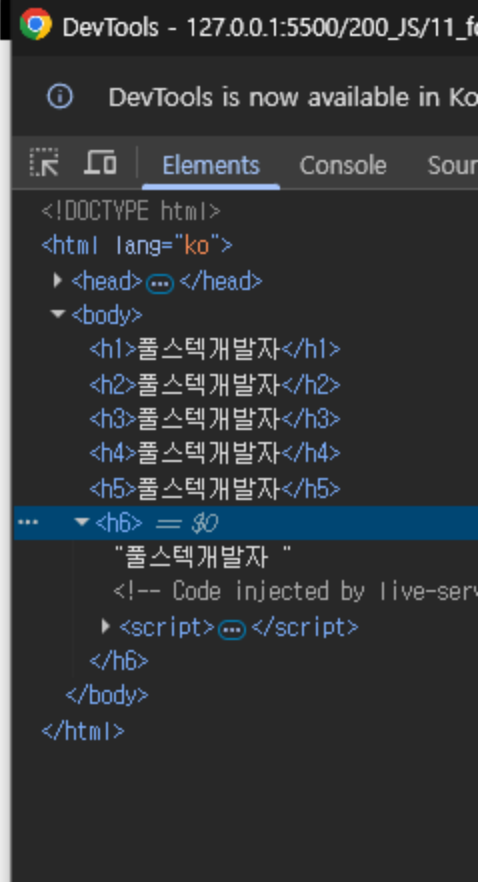
풀스택개발자

풀스택개발자

풀스택개발자

풀스택개발자

풀스택개발자



```
for (var i=1; i<=6; i++) {
  var num = i;
  console.log(num);
  document.write("<h"+ num +">"+"풀스택개발자");
}
```

※ 해석 for문 i는 초깃값 1 갖으며 6번까지 **증감**하며 돌고
var(변수) num = 에다가 i 를 대입, console.log(num) 호출하여 확인,
document.write 로 string+ 출력 하면 된다.

💀 4. 실습 역순으로 <h6>~<h1>순으로 풀스택개발자 [도큐먼트] 출력

풀스택개발자

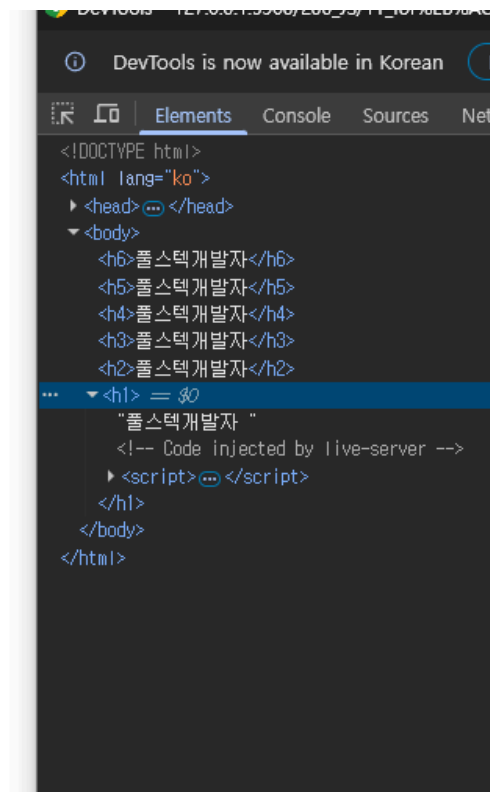
풀스택개발자

풀스택개발자

풀스택개발자

풀스택개발자

풀스택개발자



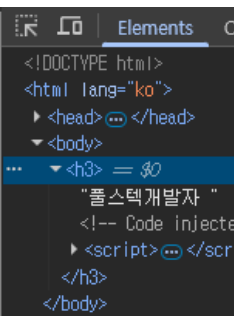
```
for (var i = 6; i >= 1; i--) {
  var num = i;
  document.write("<h" + num + ">" + "풀스택개발자");
}
```

※ 해석 for문에서 i는 초깃값 6 이 되며 1씩 감소하며 돌고
var(변수) num = 에다가 i 를 대입 document.write 로 string+ 출력 하면 된다.

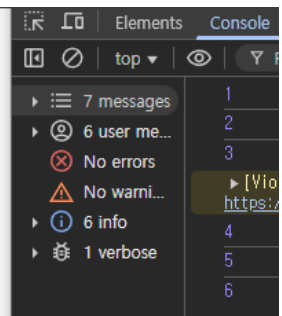
💀 5.실습 <h3> 만 뽑아 쓰기

```
for (var i = 1; i <= 6; i++) {
  var num = i;
  console.log(num);
  if (num == 3) {
    document.write("<h" + num + ">" + "풀스택개발자");
  }
}
```

풀스택개발자



풀스택개발자



※ 해석 for문에서 i는 초깃값 1로 잡고, 6번까지 **증감**하며 돌고
var(변수) num = 에다가 i 를 대입, console.log(num) 호출하여 확인,
document.write 로 string+ 출력 하면 된다.

👤 6. [도큐먼트] 구구단 2단 출력 = 단수x승수 = 단수 * 승수

2단

2x1 = 2
2x2 = 4
2x3 = 6
2x4 = 8
2x5 = 10
2x6 = 12
2x7 = 14
2x8 = 16
2x9 = 18

```
DevTools is now available in Korean
Elements Console Sources
<!DOCTYPE html>
<html lang="ko">
  <head>
  </head>
  <body>
    <h1>2단</h1>
    "2x1 = "
    <strong class="red">2</strong>
    <br>
    "2x2 = "
    <strong class="red">4</strong>
    <br>
    "2x3 = "
    <strong class="red">6</strong>
    <br>
    "2x4 = "
    <strong class="red">8</strong>
    <br>
    "2x5 = "
    <strong class="red">10</strong>
    <br>
    "2x6 = "
    <strong class="red">12</strong>
    <br>
    "2x7 = "
    <strong class="red">14</strong>
    <br>
    "2x8 = "
    <strong class="red">16</strong>
    <br>
    "2x9 = "
    <strong class="red">18</strong>
    <br>
    <!-- Code injected by live-server -->
  </body>
</html>
```



```

var i,dan,result;
dan = 2;
document.write("<h1>" + dan + "단</h1>");

for (i=1; i<= 9; i++) {
    result = (dan * i);
    console.log(result);
    document.write(dan + "x" + i + " = <strong class='red'>" + result + "</strong>" + "<br>");
}

```

※ 해석 i,dan,result 변수 만들고 , dan = 2 ; 초깃값2를 단수 부여
 for문에서 초깃값 1 이 되며 9번까지 증감하며 돌고
 result (변수) = 에다가 dan *i 를 대입, console.log(result) 호출하여 확인,
 document.write 로 string+ 출력 하면 된다.

7. (프롬프트) 구구단 단수를 입력받아 [도큐먼트] 해당 단수 결과물 출력

127.0.0.1:5500 내용:

해당 구구단수를 입력해주세요.

확인
 취소

7단

7x1 = 7

7x2 = 14

7x3 = 21

7x4 = 28

7x5 = 35

7x6 = 42

7x7 = 49

7x8 = 56

7x9 = 63

DevTools is now available

Elements

```

<!DOCTYPE html>
<html lang="ko">
  <head></head>
  <body>
    <h1>7단</h1>
    "7x1 = 7"
    <br>
    "7x2 = 14"
    <br>
    "7x3 = 21"
    <br>
    "7x4 = 28"
    <br>
    "7x5 = 35"
    <br>
    "7x6 = 42"
    <br>
    "7x7 = 49"
    <br>
    "7x8 = 56"
    <br>
    "7x9 = 63"
    <br>
    <!-- Code injected by
    <script></script>
  </body>
</html>

```

```

var dan = Number(prompt("해당 구구단수를 입력해주세요.",7));
var i, result;

document.write("<h1>" + dan + "단</h1>");

for (i = 1; i <= 9; i++) {
    result = (dan * i);
    console.log(result);
    document.write(dan + "x" + i + " = " + result + "" + "<br>");
}

```

※ 해석 i는 for문에서 초기값 1 이 되며 6번까지 증감하며 돌고
var(변수) num = 에다가 i 를 대입, console.log(num) 호출하여 확인,
document.write 로 string+ 출력 하면 된다.

◆ JavaScript 이중 for 문 (중첩 반복문)

1 구조

```

for (초깃값1; 조건식1; 증감식1) {
    // 바깥 루프(외부 반복)
    for (초깃값2; 조건식2; 증감식2) {
        // 안쪽 루프(내부 반복)
        실행문;
    }
}

```

- 외부 반복문이 한 번 돌 때마다

한눈에 보기 예제

- 내부 반복문이 전부 다 실행됨

2 흐름

1. 외부 반복문 시작 → 초깃값 1 설정
2. 조건식 1 확인 → true면 내부 반복문 시작
3. 내부 반복문
 - 초깃값 2 설정
 - 조건식 2 확인 → true면 실행문 수행
 - 증감식 2 실행 → 조건식 2 체크 반복
4. 내부 반복 종료 → 외부 반복 증감식 실행
5. 외부 조건식 1 체크 → 반복

3 예제 1 — 2단 구구단 출력

```
for (var dan = 2; dan <= 3; dan++) { // 외부 반복: 2단 ~ 3단
  console.log(dan + "단");
  for (var i = 1; i <= 9; i++) { // 내부 반복: 1~9 승수
    console.log(dan + " x " + i + " = " + (dan * i));
  }
}
```

출력 결과

```
2단
2 x 1 = 2
2 x 2 = 4
...
2 x 9 = 18
```

3단
3 x 1 = 3
3 x 2 = 6
...
3 x 9 = 27

- 외부 루프 `dan` 한 번 돌 때마다
- 내부 루프 `i` 는 1~9까지 완전히 반복됨

4 예제 2 — 1~5까지의 구구단 삼각형 출력

```
for (var i = 1; i <= 5; i++) {    // 행
    var row = "";
    for (var j = 1; j <= i; j++) { // 열
        row += j + " ";
    }
    console.log(row);
}
```

출력 결과

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

- 외부 반복문 → 행
- 내부 반복문 → 열
- 삼각형 형태 출력 가능

```
// for() 중첩문
var i,j; // 변수 i 와 j 가 있다.

for(i=2; i<=9; i++){ // 초기 i는 2를 갖고, for문을 실행하며 9까지 조건을갖고
    document.write("<h1>" + i + "단</h1>");
    for(j=1;j<=9;j++){ // 초기 j는 1를 갖고, for문을 실행하며 9까지 조건을갖고
        document.write( i + "x" + j + "=" + i*j + "<br>"); // i * j 를 하여 write 출력
    }
}
```

※ 해석 i , j 변수 선언
 첫번째 for문에다가 i = 2 초깃값 9까지 증감
 write(출력)
 두번째 for문에다가 j = 1 초깃값 9까지 증감
 document.write 로 i * j + string+ 출력 하면 된다.

◆ JavaScript 함수 (function) = 캐치볼

1 함수와 입력값

함수는 특정 작업을 수행하는 코드 묶음이라고 보면된다.

- 함수는 한 번 만들어서 필요할 때마다 호출하고, 값을 넣어주면 처리 결과를 돌려주는 역할을 한다.

여기서 값을 함수에 넣는 것이 바로 매개변수와 인자 개념과 연결된다.

2 매개변수(Parameter)

- 함수가 입력을 받을 때 사용하는 변수
- 함수를 정의할 때 괄호 안에 적는 것

예제:

```
function add(a, b) { // a, b가 매개변수
  return a + b;
}
```

- **a**와 **b**가 **매개변수(parameter)**
- 함수 안에서 **a + b** 처럼 **변수처럼 사용 가능**
- 함수가 실행될 때, 실제 숫자가 들어와야 계산 가능

비유:

- **매개변수 = 함수의 손**
- 공(값)을 받을 준비가 되어 있는 손

3 인자(Argument)

- 함수를 호출할 때 실제로 전달하는 값
- 함수 안에서는 이 값이 매개변수에 들어감

예제:

```
add(3, 5); // 3과 5가 인자(argument)
```

- **3** → 매개변수 **a**에 들어감
- **5** → 매개변수 **b**에 들어감
- 함수 안에서 **a + b = 3 + 5 = 8** 계산 후 결과 반환

비유:

- **인자 = 던지는 공**
- 함수 손(매개변수)이 공(인자)을 받아 처리 → 결과 반환

4 한눈에 비교

구분	설명	예시
매개변수(P)	함수 정의 시 입력을 받는 변수	<code>function add(a, b)</code>
인자(A)	함수 호출 시 실제로 전달하는 값	<code>add(3, 5)</code>

5 그림으로 이해 (캐치볼 비유)

공 던지기 → 함수 손 → 공 처리 → 다시 돌려주기
 3,5 a,b a+b 8

- 공 = 인자(argument)
- 손 = 매개변수(parameter)
- 결과 = return 값

6 한눈에 보기 예제

```
function addTen(num) { // num = 매개변수
  return num + 10;
}
```

```
addTen(5); // 5 = 인자 → 결과 15
```

- 매개변수: `num`
- 인자: `5`
- 결과: `15`

선언적 함수 (Named Function / 선언문)

```
function test() {
  alert("선언적함수 : 자바스크립트 개념정리");
}
```

```
test(); // 호출
```

특징

1. 함수 이름 있음 → `test`
2. 호이스팅(Hoisting) 가능
 - 선언된 위치보다 위에서 호출해도 실행 가능

```
test(); // ✅ 호출 가능  
function test() { alert("OK"); }
```

3. 호출 방법
 - 그냥 함수 이름 + 괄호 → `test()`
4. 여러 번 호출 가능

2 익명 함수 (Anonymous Function / 함수 표현식)

```
var noName = function() {  
    alert("익명함수: 자바스크립트 수업");  
};  
noName(); // 호출
```


1-1. 익명 (이름이 정해지지않은) 함수 만들기

```
var count = 0;
var noName = function () {
    count++;
    document.write("익명함수 : " + count + "<br>");
}
```

```
var count = 0;
var noName = function () {
    count++;
    document.write("익명함수 : " + count + "<br>");
}
```

(형식) var = 참조명 = function() {실행문;}

변수에 noName이라는 함수를 선언 = function () {
count++ // 증감
document.write를 출력 "익명함수" count 변수 호출
}

1-2 . 선언적 (이름이 지정된) 함수 만들기

```
/* 1-2. 선언적 함수 만들기 */  
function pub() {  
    count++;  
    document.write("선언적 함수 : " + count + "<br>");  
}  
pub();
```

(형식) var = 참조명 = function() {실행문;}

함수 이름은 pub () 호출한다. 안에 술식은 익명함수와 동일

💀 1. (프롬프트) 숫자를 받아, 입력받은 숫자 + 10을 한 결과값을 반환하는 함수 제작

```
var user = Number(prompt("숫자를 입력하세요", 1)); //초기값 1  
var add = function (num) {  
    return num + 10; // 입력값 + 10 반환  
}  
alert("결과:" + add(user));
```

```
var user = Number(prompt("숫자를 입력하세요", 1)); //초기값 1  
var add = function (num) {  
    return num + 10; // 입력값 + 10 반환
```

```
}
alert("결과:" + add(user));
```

👤 2. 매개 변수 활용하여 인자값을 파라미터로 함수에 넣어 연동하기

1 코드 보기

```
function ezFn(name, area) { // ①
  ezFn2(name);           // ②
  ezFn3(area);           // ③
}

function ezFn2(name) {
  document.write("안녕하세요 " + name + "입니다.<br>")
}

function ezFn3(area) {
  document.write("사는곳은 " + area + " 입니다.<br>")
}

// 호출
ezFn("김현수", "노원구"); // ④
```

코드에서 매개변수 확인

함수	매개변수	설명
ezFn	name, area	외부에서 전달되는 이름과 지역값을 받음
ezFn2	name	ezFn에서 전달된 이름을 받아서 출력
ezFn3	area	ezFn에서 전달된 지역을 받아서 출력

3 호출 시 인자(argument)

```
ezFn("김현수", "노원구");
```

- "김현수" → `name` 매개변수에 들어감
- "노원구" → `area` 매개변수에 들어감

💡 쉽게 말하면:

- **매개변수** = 함수 안에서 공을 받을 준비가 되어 있는 손
- **인자** = 실제 던지는 공
- `ezFn` 호출 시 "김현수" 라는 공을 `name` 손에 전달 → "안녕하세요 김현수입니다." 출력

4 함수 내부에서 매개변수 활용

```
function ezFn(name, area) {  
  ezFn2(name); // name 매개변수 사용  
  ezFn3(area); // area 매개변수 사용  
}
```

- ezFn2와 ezFn3도 각각 매개변수가 있음
- ezFn에서 받은 값을 다시 다른 함수에 전달할 수도 있음 → 유연하게 값 전달 가능

◆ JavaScript 함수 (function) + click(클릭이벤트)

1 코드 구조

```
ezFn("김현수", "노원구");

function ezFn(name, area) {
  ezFn2(name);
  ezFn3(area);
}

function ezFn2(name) {
  document.write("안녕하세요 " + name + "입니다.<br>")
}

function ezFn3(area) {
  document.write("사는곳은 " + area + " 입니다.<br>")
}
```

```
<button onclick="ezFn2('김현수')">버튼1</button>
<button onclick="ezFn3('노원구')">버튼2</button>
```

2 버튼과 onclick 이벤트

1. `<button>` 태그 → 사용자 클릭
2. `onclick="함수(값)"` → 버튼 클릭 시 함수 호출

예를 들어:

```
<button onclick="ezFn2('김현수')">버튼1</button>
```

- 버튼1 클릭 → `ezFn2` 함수 실행
- `'김현수'` 가 **인자(argument)**로 함수에 전달
- 함수 안에서 매개변수 `name` 에 할당 하며 → `"안녕하세요 김현수입니다.
"` 출력

버튼2

```
<button onclick="ezFn3('노원구')">버튼2</button>
```

- 버튼2 클릭 → ezFn3 실행
- '노원구' 가 매개변수 area 에 전달 → "사는곳은 노원구 입니다.
" 출력

3 매개변수 + 버튼 동작 흐름

1. 매개변수 = 함수 내부에서 값을 받을 준비된 변수
2. 인자 = 버튼 클릭 시 실제 전달되는 값

버튼 클릭 → 인자 전달 → 매개변수에 값 저장 → 함수 실행 → 출력

- 버튼1 → ezFn2('김현수')
 - 매개변수 name = '김현수'
 - 결과: "안녕하세요 김현수입니다."
- 버튼2 → ezFn3('노원구')
 - 매개변수 area = '노원구'
 - 결과: "사는곳은 노원구 입니다."

4 기존 ezFn 호출과 버튼 호출 비교

```
ezFn("김현수", "노원구");
```

- 이 코드는 페이지 로딩 시 자동 실행
- 내부에서 ezFn2와 ezFn3 호출 → 두 문장 출력

```
<button onclick="ezFn2('김현수')">버튼1</button>  
<button onclick="ezFn3('노원구')">버튼2</button>
```

- 버튼 클릭 시 사용자가 원하는 시점에 출력

- 매개변수와 인자 활용: 함수 재사용 가능

5 요약

개념	예시	설명
매개변수(Parameter)	<code>name</code> , <code>area</code>	함수 안에서 값을 받는 변수
인자(Argument)	<code>'김현수'</code> , <code>'노원구'</code>	함수 호출 시 전달하는 실제 값
버튼 클릭 이벤트	<code>onclick="ezFn2('김현수')"</code>	클릭 시 함수 실행 + 인자 전달

◆ JavaScript 함수 - 배열(Array)

배열은 모든 데이터타입을 담을수 있다.

배열은 같은 변수명에 데이터가 여러개 있는것 (나누어 ,열맞추다)

메모리열 0,1,2,3,4,5 = index; (0 ~4)

`var color = ["red","green","blue","black","pink"];` // 5개의 문자열 (1~5)

`console.log(color[2]);` // 결과값 blue {메모리열 2번째[2] , 문자열 3번째 }

```
var color = ["red", "green", "blue", "black", "pink"];
```

```
var i = 0; //초깃값
```

```
//console.log(color[2]);
```

```
function bgCh() {
```

```
    var body = document.body;
```

```
    body.style.backgroundColor = color[i];
```

```
    i++;
```

```
    if (i >= color.length) {
```

```
        //만약( 변수 i가 color 배열에 담긴 길이or갯수(length)보다 같거나 높을때까지 조건  
선언
```

```
        // 로직은 i가 0,1,2,3,4 까지 갔을때
```

```
i = 0; // 다시 i를 0으로 만들어준다
}
}

```

밑에 글에서 좀더 자세히 정리 해보겠다.

```
var color = ["red", "green", "blue", "black", "pink"];
```

- 배열 선언
- `color` 라는 변수에 5개의 문자열 색상 저장
- `color[0]` = "red", `color[1]` = "green", ..., `color[4]` = "pink"

```
var i = 0; //초깃값
```

- 인덱스 초기값 설정
- 배열의 현재 위치를 추적하기 위한 변수
- 처음에는 `0` → 빨강 색상부터 시작

```
function bgCh() {
  var body = document.body;

```

- 함수 선언: `bgCh()`
- 클릭 이벤트나 호출 시 실행됨
- `document.body` → HTML 문서의 `<body>` 요소 선택

```
body.style.backgroundColor = color[i];
```

- 배경색 변경
- 현재 인덱스 `i` 가 가리키는 색상을 `<body>` 의 `backgroundColor` 속성에 적용
- 예: `i = 0` → 빨강, `i = 1` → 초록


```
i++;
```

- 다음 색상으로 이동
- 클릭 시 배열의 다음 인덱스로 증가
- 즉, 다음 번 클릭 때 다른 색상을 적용하도록 준비

```
if (i >= color.length) {  
    i = 0;  
}
```

- 유효성 검사 / 순환 로직
- `i` 가 배열 길이보다 같거나 커지면 → 배열 끝에 도달
- `i = 0` → 다시 처음 색상으로 초기화 → **배경색이 무한 순환**

`if (i >= color.length)` 의 의미는

"만약 `i` 값이 `color` 배열의 길이(`length`)보다 크거나 같다면" 이라는 뜻

- `color.length` → 5
- `i` 값이 하나씩 증가하다가
 - `i = 0` → red
 - `i = 1` → green
 - `i = 2` → blue
 - `i = 3` → black
 - `i = 4` → pink
 - `i = 5` → ❌ (배열 인덱스 없음)

그래서 `i`가 5 이상이 되면 색상이 없으니까, 다시 0으로 초기화

 코드 흐름

```
if (i >= color.length) {
    i = 0; //
```

즉, 배열의 끝에 도달했을 때 처음으로 돌아가라 는 제어문인 것이다. .

클릭이벤트 함수 와 매개변수 활용

```
<title>함수사용, 매개변수(Parameter)</title>
<!-- 3. 이벤트 호출 -->
<script>

// 1. 실행 함수
function ezFn() { // 김현수, 노원구
    var name = prompt("이름을 입력하세요", "김현수");
    var area = prompt("사는곳을 입력하세요", "노원구");
    ezFn2(name);
    ezFn3(area);
}

// 2. 실행 함수
function ezFn2(name) { // 이름1
    var chName = document.getElementById("name");
    chName.textContent = name;
}

// 3. 실행 함수
function ezFn3(area) { // 구청1
    var chAarea = document.getElementById("area");
    chAarea.textContent = area;
}
</script>
</head>

<body>

<!-- 1. 초기는 html로 구조 잡고 -->
<div id="output">
    <p>안녕하세요 <span id="name">홍길동</span>입니다.</p>
    <p>사는곳은 <span id="area">강남구</span>입니다.</p>
</div>#output

<!-- 2. 클릭시 발생한 이벤트 해당 함수 호출 (변경할 매개변수 , 초깃값) 넣기 -->
<button onclick="ezFn2('이름1')">버튼1</button>
<button onclick="ezFn3('구청1')">버튼2</button>
<button onclick="ezFn()">전체 변경 버튼</button>
```

안녕하세요 홍길동입니다.

사는곳은 강남구입니다.

버튼1 버튼2 전체 변경 버튼

초기 HTML BODY 안에 OUTPUT

안녕하세요 이름1입니다.

사는곳은 강남구입니다.

버튼1 버튼2 전체 변경 버튼

버튼 1을 누르면 홍길동 > 이름1 로 변경

안녕하세요 이름1입니다.

사는곳은 구청1입니다.

버튼1 버튼2 전체 변경 버튼

버튼 2를 누르면 강남구 > 구청1 로 변경

안녕하세요 김현수입니다.

사는곳은 노원구입니다.

버튼1 버튼2 전체 변경 버튼

전체 변경버튼 누르면 프롬프트 호출 및 함수
대입

함수 종합 실습

```
<title>함수 종합실습</title>
<script>
  /*
  // 그냥함수 호출
  (function () {
    alert("자바스크립트 수업");
  })();
```

```

*/
/*실습1. 익명함수를 만들고, 로딩시 (경고창)"자바스크립트 수업"*/
var noName = function () {
    alert("익명함수: " + "자바스크립트 수업");
};
noName();
/*실습2. 선언적함수를 만들고 = test, 로딩시 (브라우저)"자바스크립트 개념정리"*/
function test() {
    alert("선언적함수 : " + "자바스크립트 개념정리");
}
test();
/*
실습3.
버튼1을 클릭했을때 => (경고창)"안녕하세요."
버튼2를 클릭했을때 => (경고창)"반갑습니다." 매개변수 사용
*/
function btn1() {
    alert("안녕하세요");
}
function btn2() {
    alert("반갑습니다");
}

/* 실습4. 배열실습 = 배열색 바꾸기 이벤트 코딩 해보기 */
var color = ["red", "green", "blue", "black", "pink"];
var i = 0; //초깃값
//console.log(color[2]);

function bgCh() {
    var body = document.body;
    body.style.backgroundColor = color[i];
    i++; // i ++ 증감

    if (i >= color.length) { // i가 >= 같거나 높을때까지 color 배열에 담긴 갯수만
        i = 0; //조건 완료시 안에 내용 실행 , 다시 i를 0으로 만들어준다
    }
}

```

```
    }  
  }  
</script>  
</head>  
<body>  
  <button onclick="btn1();" >버튼1</button>  
  <button onclick="btn2();" >버튼2</button>  
  <button onclick="bgCh();" >배경바꾸기</button>  
</body>
```

attachment:e10a1c53-0a4c-433b-b76b-94d411150f44:함수_종합실습_-_Chrome_2025-09-12_17-04-05.mp4