

**01**

# **자바스크립트 이해하기**

# 01 자바스크립트 이해하기

## ■ 자바스크립트

### 1. 자바스크립트의 특징

- 자바와 다른 프로그래밍 언어
  - HTML에 명령을 내릴 때 사용
  - HTML에 동적인 기능을 구현
- ECMAScript6(스크립트 언어 표준)

### 2. 자바스크립트의 위치

- HTML 내부 자바스크립트
- 외부 자바스크립트 파일
- 태그 안의 자바스크립트

# 01 자바스크립트 이해하기

## ■ 변수



그림 6-3 프로그래밍과 요리

# 01 자바스크립트 이해하기

## ■ 변수

### • 자료형

- 뛰어서 100미터 11.5초에 가능?



그림 6-4 '뛰어서 100미터 11.5초에 가능?'의 자료형

# 01 자바스크립트 이해하기

## ■ 변수

- 자료형

표 6-1 자바스크립트에서 다루는 데이터와 자료형

데이터 종류	자료형	내용
문자	String	가, 나, 다, a, b, c와 같은 문자 자료형
정수	Number	-1, 0, 100과 같은 정수 데이터
실수	Number	11.5, 3.14, 9.9와 같은 실수 데이터
불린	Boolean	값이 true와 false뿐인 자료형

# 01 자바스크립트 이해하기

## ■ 변수

### 1. 자바스크립트에서 변수 만들기

표 6-2 변수 만들기

let	var
<pre>let num; num = 5;</pre>	<pre>var num; num = 5;</pre>

표 6-3 변수를 만들면서 값을 담고, 이름이 같은 변수를 다시 만들기

let	var
<pre>let text = "hello"; let text = "hi!"; // 불가능 - 오류!</pre>	<pre>var text = "hello"; var text = "hi!"; // 가능</pre>

# 01 자바스크립트 이해하기

## ■ 변수

### 1. 자바스크립트에서 변수 만들기

표 6-4 상수 만들기

const로 상수 만들기	const로 상수를 선언하고 값을 담지 않음
<code>const pi = 3.14;</code>	<code>const pi;    // 선언과 동시에 값을 담아야 함 - 오류!</code> <code>pi = 3.14;</code>

표 6-5 var, let, const의 비교

구분	var	let	const
같은 이름의 새로운 변수를 만들 수 있는가?	○	×	×
만든 변수에 새로운 값을 담을 수 있는가?	○	○	×

# 01 자바스크립트 이해하기

## ■ 변수

### 1. 자바스크립트에서 변수 만들기

- 변수 여러 개 만들기

```
let text = "hello", num = 7;
```

- 변수의 초기화
  - undefined ↔ null



# 01 자바스크립트 이해하기

## ■ 변수

### 2. 만능 변수와 만능 덧셈 연산

```
let text = "hello", num = 7, merge;  
merge = text + num;
```

표 6-6 만능 덧셈

덧셈	내용
$7 + 7 = 14$	일반적인 덧셈 연산
$'7' + '7' = 77$	문자와 문자를 합침
$'7' + 7 = 77$	문자와 숫자의 덧셈 결과는 문자

# 01 자바스크립트 이해하기

예제 6-2

var, let, const로 변수 만들기

ex6-2.html

```
</head>
<body>
  <script>
    // var는 자바스크립트 초기 방식
    var varNum = 1;
    varNum = varNum + 10;
    document.write("varNum = " + varNum + "<hr>");
    // let은 같은 이름의 변수 다시 만들 수 없음
    let letNum = "hello";
    letNum = letNum + varNum;
    document.write("letNum = " + letNum + "<hr>");
    // const는 값을 바꿀 수 없음
    const pi = 3.14;
    document.write("pi = " + pi)
  </script>
</body>
</html>
```

# 01 자바스크립트 이해하기

## ■ 변수

### 3. 변수 이름의 규칙

표 6-7 변수 이름의 규칙

규칙	사용 가능	사용 불가
대소문자 구분		
영문자, 숫자, _, \$만 사용	let_num	let num
숫자부터 시작하면 안 됨	total3	3total
키워드 사용 불가	letNum	let

- 의미 있는 단어를 사용하는 것이 좋다
- 카멜 표기법

# 01 자바스크립트 이해하기

## ■ 연산자

### 1. 연산자의 우선순위

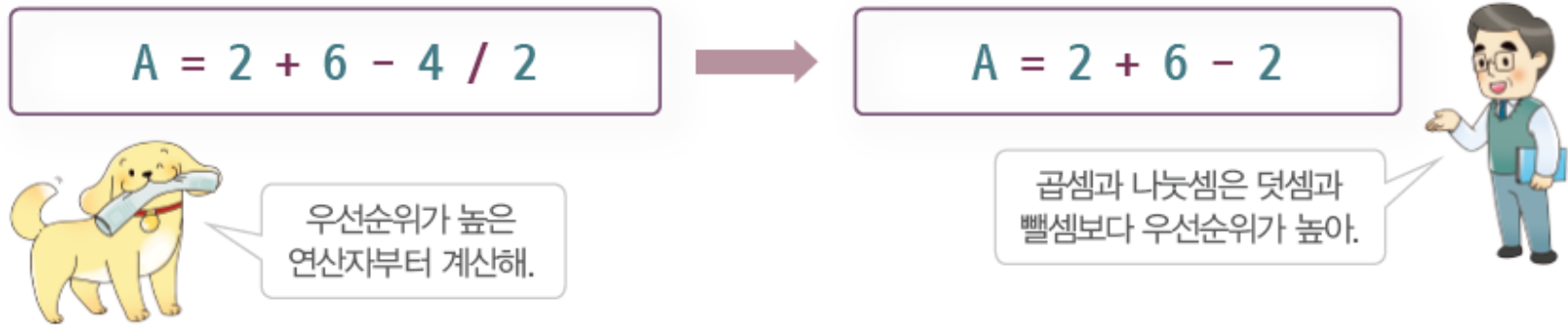


그림 6-5 연산자의 우선순위

# 01 자바스크립트 이해하기

## ■ 연산자

### 2. 연산자의 종류

- 산술 연산자

표 6-8 산술 연산자

연산자	의미	표현식	설명	우선순위
++	증가	++A	A를 1 증가시킨 값	가장 높다
--	감소	--A	A를 1 감소시킨 값	
**	지수	A ** B	A의 B 거듭제곱( $A^B$ )	매우 높다
-	부호	-A	음수를 양수로, 양수를 음수로 변경	높다
*	곱셈	A * B	A와 B의 곱	중간
/	나눗셈	A / B	A를 B로 나눈 값	
%	나머지	A % B	A를 B로 나눈 나머지	
+	덧셈	A + B	A와 B의 합	낮다
-	뺄셈	A - B	A와 B의 차	

# 01 자바스크립트 이해하기

## ■ 연산자

### 2. 연산자의 종류

- 산술 연산자
- 증감 연산자

표 6-9 증감 연산자

전위 증감 연산자	후위 증감 연산자	내용	원래 연산
<code>++A</code>	<code>A++</code>	A 값을 1 증가시킴	$A = A + 1$
<code>--A</code>	<code>A--</code>	A 값을 1 감소시킴	$A = A - 1$

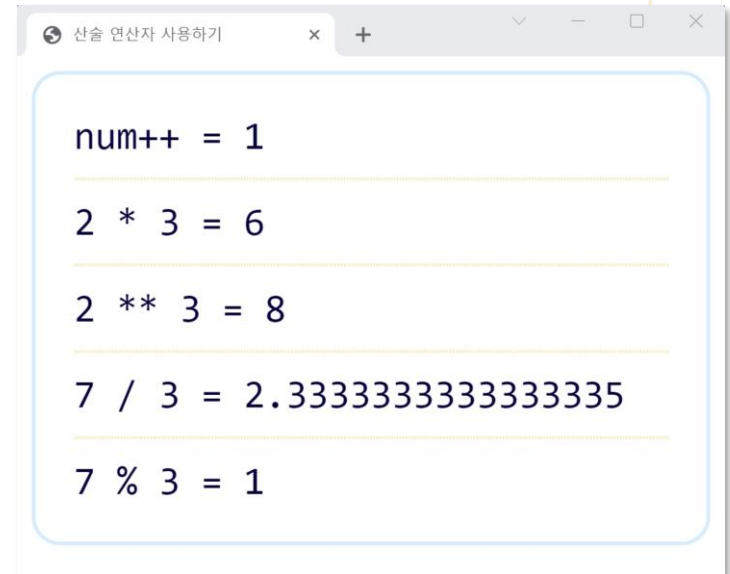
# 01 자바스크립트 이해하기

예제 6-3

산술 연산자 사용하기

ex6-3.html

```
</head>
<body>
  <script>
    let num = 0;
    num++;
    document.write("num++ = " + num + "<hr>");
    num = 2 * 3;
    document.write("2 * 3 = " + num + "<hr>");
    num = 2 ** 3;
    document.write("2 ** 3 = " + num + "<hr>");
    num = 7 / 3;
    document.write("7 / 3 = " + num + "<hr>");
    num = 7 % 3;
    document.write("7 % 3 = " + num);
  </script>
</body>
</html>
```



# 01 자바스크립트 이해하기

## ■ 연산자

### 2. 연산자의 종류

- 산술 연산자
  - 복합 대입 연산자

표 6-10 복합 대입 연산자

원래 연산자	복합 대입 연산자
$A = A ** B$	$A ** = B$
$A = A * B$	$A * = B$
$A = A / B$	$A / = B$
$A = A \% B$	$A \% = B$
$A = A + B$	$A += B$
$A = A - B$	$A -= B$



# 01 자바스크립트 이해하기

## ■ 연산자

### 2. 연산자의 종류

- 비교 연산자

표 6-11 비교 연산자

연산자	의미	우선순위
>	크다	높다
<	작다	
>=	크거나 같다	
<=	작거나 같다	
==	값이 같다	낮다
===	값과 종류가 같다	
!=	다르다	
!==	값 혹은 종류가 다르다	

# 01 자바스크립트 이해하기

## ■ 연산자

### 2. 연산자의 종류

- 비교 연산자

표 6-12 값과 자료형이 일치하는지 비교하는 연산자 ===

연산	결과
10 == "10"	자료형은 다르나 값이 같으므로 true
10 === "10"	값은 같으나 자료형이 다르므로 false

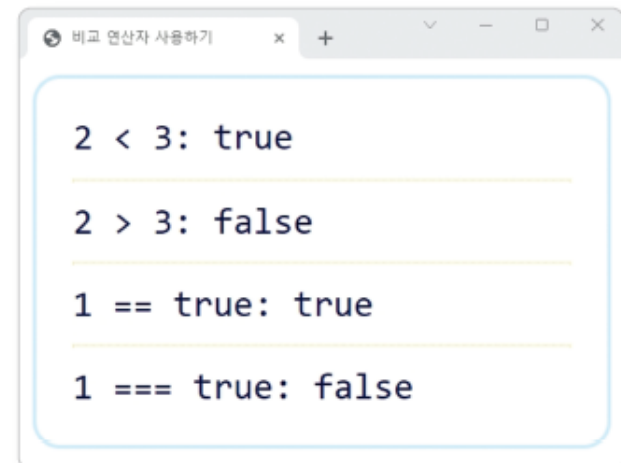
# 01 자바스크립트 이해하기

예제 6-4

비교 연산자 사용하기

ex6-4.html

```
</head>
<body>
  <script>
    let jud;
    jud = 2 < 3;
    document.write("2 < 3: " + jud + "<hr>");
    jud = 2 > 3;
    document.write("2 > 3: " + jud + "<hr>");
    jud = 1 == true;
    document.write("1 == true: " + jud + "<hr>");
    jud = 0 === false;
    document.write("1 === true: " + jud);
  </script>
</body>
</html>
```



# 01 자바스크립트 이해하기

## ■ 연산자

### 2. 연산자의 종류

- 논리 연산자

표 6-13 논리 연산자의 우선순위

연산자	의미	표현식	의미	우선순위
!	부정	not A	참을 거짓으로, 거짓을 참으로 변경	매우 높다
&&	논리곱	A and B	양쪽이 모두 참일 때 참	낮다
	논리합	A or B	한쪽만 참일 때 참	



그림 6-6 논리 연산자

# 01 자바스크립트 이해하기

## ■ 연산자

### 3. 여러 종류의 연산자가 섞여 있을 때의 우선순위



그림 6-7 연산자의 우선순위

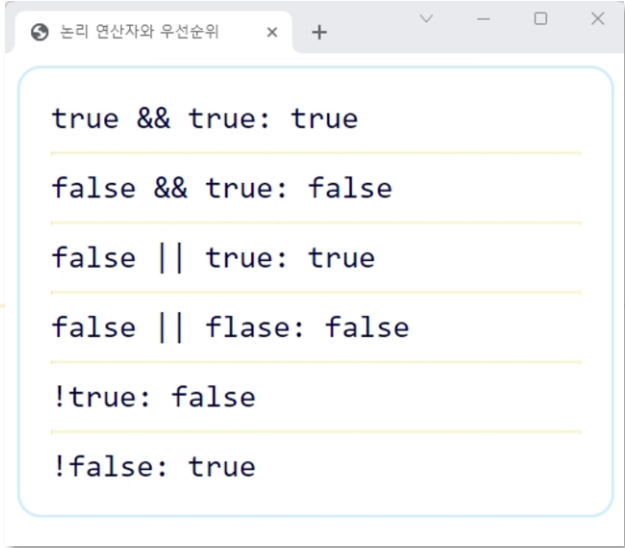
# 01 자바스크립트 이해하기

예제 6-5

논리 연산자와 우선순위

ex6-5.html

```
</head>
<body>
  <script>
    document.write("true && true: " + (true && true) + "<hr>");
    document.write("false && true: " + (false && true) + "<hr>");
    document.write("false || true: " + (false || true) + "<hr>");
    document.write("false || flase: " + (false || false) + "<hr>");
    document.write("!true: " + !true + "<hr>");
    document.write("!false: " + !false);
  </script>
</body>
</html>
```



true && true: true

false && true: false

false || true: true

false || flase: false

!true: false

!false: true

# 01 자바스크립트 이해하기

## ■ 사용자 입력받기

```
let num; //num 변수 선언  
num = prompt("숫자를 입력하세요"); //prompt()로 입력받은 문자열을 num에 담기
```

표 6-14 문자를 Number 자료형으로 바꾸는 함수

함수	내용
<code>parseInt("10")</code>	문자를 정수 10으로 바꿈
<code>parseFloat("10")</code>	문자를 실수 10.0으로 바꿈

```
let num = parseInt( prompt("숫자를 입력하세요") );
```

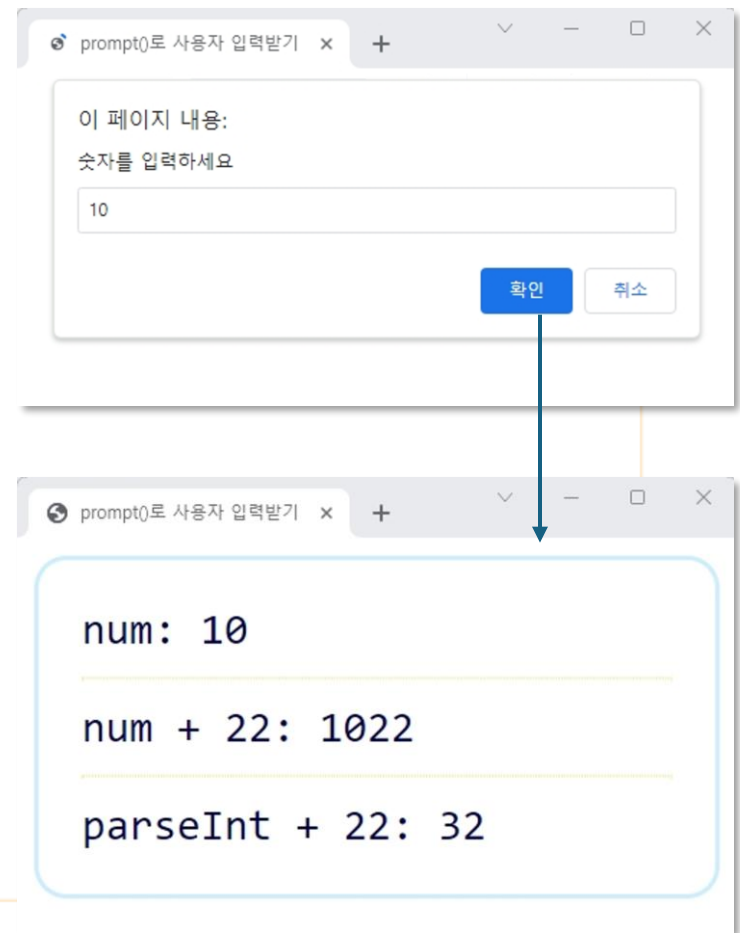
# 01 자바스크립트 이해하기

예제 6-6

prompt()로 사용자 입력받기

ex6-6.html

```
</head>
<body>
  <script>
    let num, add;
    num = prompt("숫자를 입력하세요");
    document.write("num: " + num + "<hr>");
    add = num + 22;
    document.write("num + 22: " + add + "<hr>");
    num = parseInt(num);
    add = num + 22;
    document.write("parseInt + 22: " + add);
  </script>
</body>
</html>
```





**02**

# **조건문과 반복문**

## 02 조건문과 반복문

### ■ 조건문

#### 1. if 조건문

```
if ( 조건 ) {  
    참일 때 실행;  
}  
일반 코드;
```



조건이 참(true)일 때  
실행될 거야.

그림 6-9 if 조건문의 구조

```
if (rain == true) {  
    document.write("우산 가져가기<hr>");  
}
```

## 02 조건문과 반복문

### ■ 조건문

#### 2. if-else 조건문

```
if ( 조건 ) {  
    참일 때 실행;  
}  
else {  
    거짓일 때 실행;  
}  
일반 코드;
```



조건이 참(true)일 때와  
거짓(false)일 때  
실행되는 코드가 달라.

그림 6-10 if-else 조건문의 구조

## 02 조건문과 반복문

### ■ 조건문

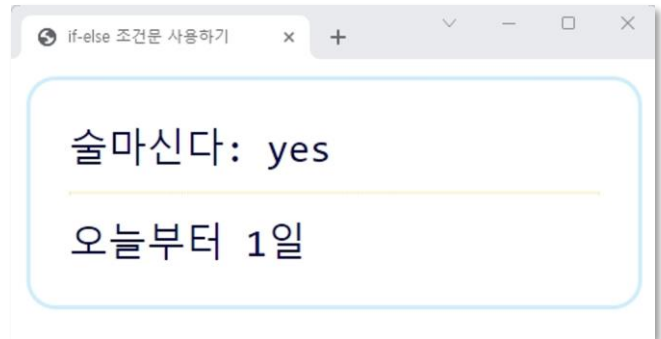
#### 2. if-else 조건문

예제 6-7

if-else 조건문 사용하기

ex6-7.html

```
</head>
<body>
  <script>
    let drink = prompt("이거 마시면 사귀는 거다!");
    document.write("술마신다: " + drink + "<hr>");
    if (drink == "yes") {
      document.write("오늘부터 1일");
    } else {
      document.write("헤어진다");
    }
  </script>
</body>
</html>
```



## 02 조건문과 반복문

### ■ 조건문

#### 2. if-else 조건문

- 3항 조건문

조건 ? 참일 때 실행 : 거짓일 때 실행

그림 6-11 3항 조건문의 구조

```
drink == "yes" ? document.write("오늘부터 1일") : document.write("헤어진다");
```

## 02 조건문과 반복문

### ■ 조건문

#### 3. 다중 조건문



그림 6-12 별다방 쿠폰 행사

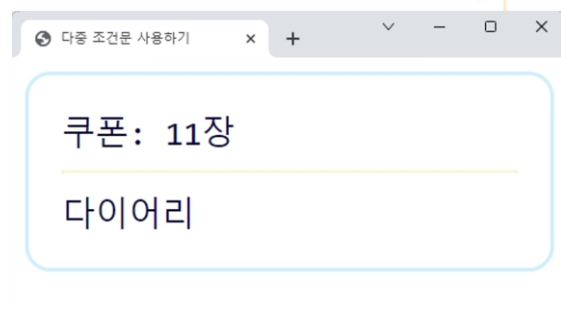
## 02 조건문과 반복문

예제 6-8

다중 조건문 사용하기

ex6-8.html

```
</head>
<body>
  <script>
    let coupon = parseInt(prompt("쿠폰 개수를 입력하십시오"));
    document.write("쿠폰: " + coupon + "장<hr>");
    if (coupon >= 20) {
      document.write("여행용 가방");      // 20 이상일 때
    } else if (coupon >= 10) {
      document.write("다이어리");        // 10 이상일 때
    } else {
      document.write("커피 한 잔");      // 10 미만일 때
    }
  </script>
</body>
</html>
```



## 02 조건문과 반복문

### ■ 조건문

#### 4. switch 조건문

```
switch ( 조건 ) {  
    case A:  
        실행문 A;  
        break;  
    case B:  
        실행문 B;  
        break;  
    default:  
        실행문 C;  
}
```



**case A :**  
조건이 A이면  
실행문 A만 실행



**case B :**  
조건이 B이면  
실행문 B만 실행



**default :**  
조건이 A도 B도 아니면  
실행문 C만 실행

그림 6-13 switch 조건문의 구조



## 02 조건문과 반복문

### ■ 조건문

#### 4. switch 조건문

```
<script>
  let coupon = parseInt(prompt("쿠폰 개수를 입력하시오"));
  document.write("쿠폰: " + coupon + "장<hr>");
  coupon = Math.floor(coupon / 10);
  switch (coupon) {
    case 2:
      document.write("여행 가방");    // 20 이상일 때
      break;
    case 1:
      document.write("다이어리");    // 10 이상일 때
      break;
    default:
      document.write("커피 한 잔");    // 10 미만일 때
  }
</script>
```

## 02 조건문과 반복문

### ■ 조건문

#### 4. switch 조건문

- switch 조건문의 활용

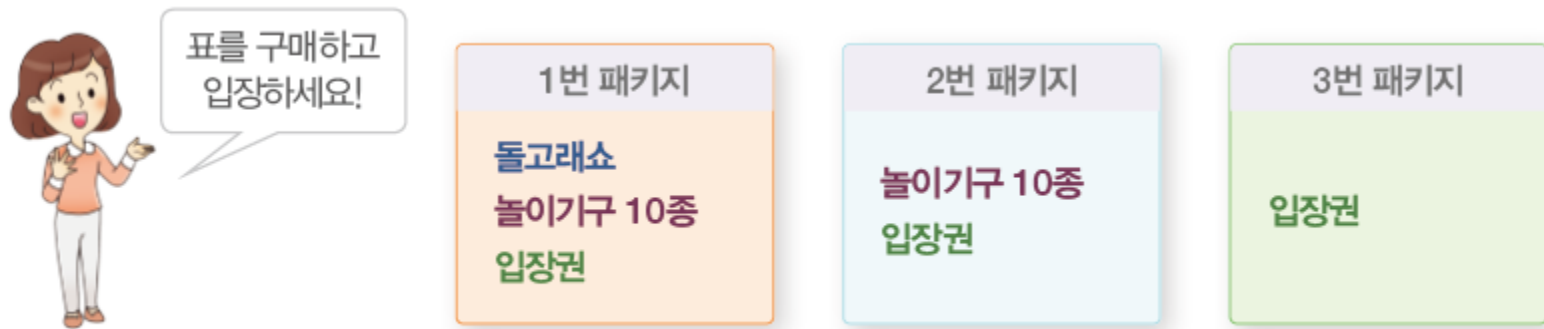


그림 6-14 놀이동산 패키지

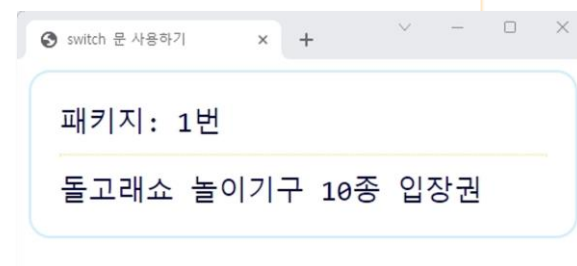
## 02 조건문과 반복문

예제 6-9

switch 조건문 사용하기

ex6-9.html

```
</head>
<body>
  <script>
    let coupon = parseInt(prompt("패키지 번호를 입력하세요"));
    document.write("패키지: " + coupon + "번<hr>");
    switch (coupon) {
      case 1:
        document.write("돌고래쇼 ");           // 1번 패키지
      case 2:
        document.write("놀이기구 10종 ");       // 2번 패키지
      default:
        document.write("입장권");               // 3번 패키지
    }
  </script>
</body>
</html>
```



## 02 조건문과 반복문

### ■ 반복문

#### 1. while 반복문

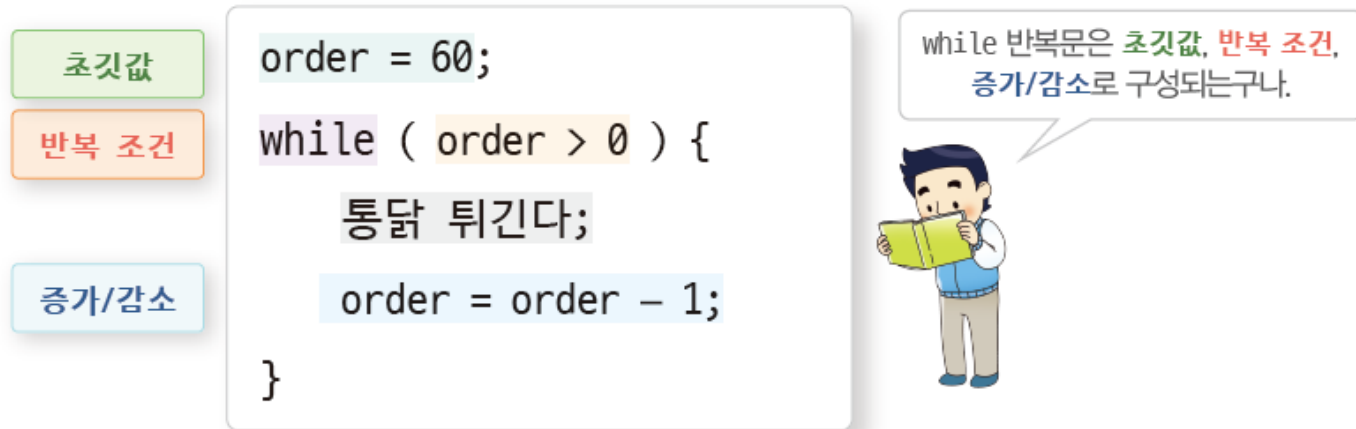


그림 6-15 while 반복문의 구조

표 6-15 if 조건문과 while 반복문

조건문	내용
<code>if (order &gt; 0)</code> 통닭 튀긴다;	조건이 참이면 1마리 튀긴다.
<code>while (order &gt; 0)</code> 통닭 튀긴다;	조건이 참인 동안 계속 튀긴다.

## 02 조건문과 반복문

### ■ 반복문

#### 2. for 반복문

for 반복문은 **초깃값**,  
**반복 조건**, **증가/감소**를  
괄호 안에 합친 구조야.



초깃값;



반복 조건;

3가지 구성 요소를  
;으로 구분해.

증가/감소

마지막 요소인 증가/감소  
뒤에는 ;을 생략해.



```
for ( order = 60 ; order > 0 ; order = order - 1 )
```

그림 6-16 for 반복문의 구조

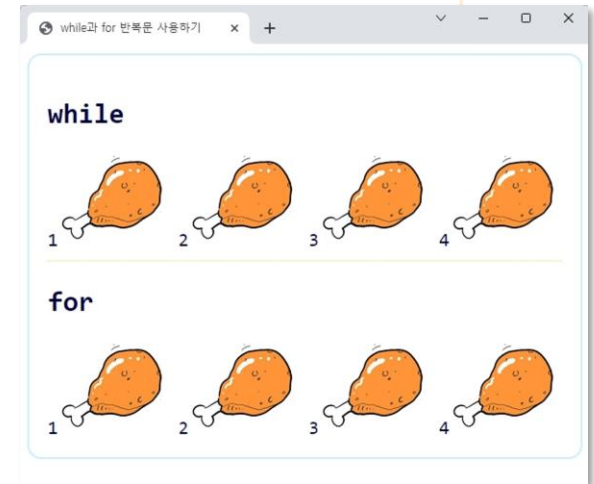
## 02 조건문과 반복문

### 예제 6-10

### while과 for 반복문 사용하기

ex6-10.html

```
</head>
<body>
  <h2>while</h2>
  <script>
    let i = 1;                // 초깃값
    while (i < 5) {           // 반복 조건
      document.write(i + "<img src='../media/chicken.png'> ");
      i++;                    // 증가
    }
  </script>
  <hr>
  <h2>for</h2>
  <script>
    for (let k = 1; k < 5; k++)
      document.write(k + "<img src='../media/chicken.png'> ");
  </script>
</body>
</html>
```



## 02 조건문과 반복문

### ■ 반복문

#### 3. do while 반복문

```
let order = 1;
do {
  document.write(i + "<img src='../media/chicken.png'> ");
  order++;
} while (order < 5);
```

## 02 조건문과 반복문

### ■ 반복문

#### 4. 다중 반복문

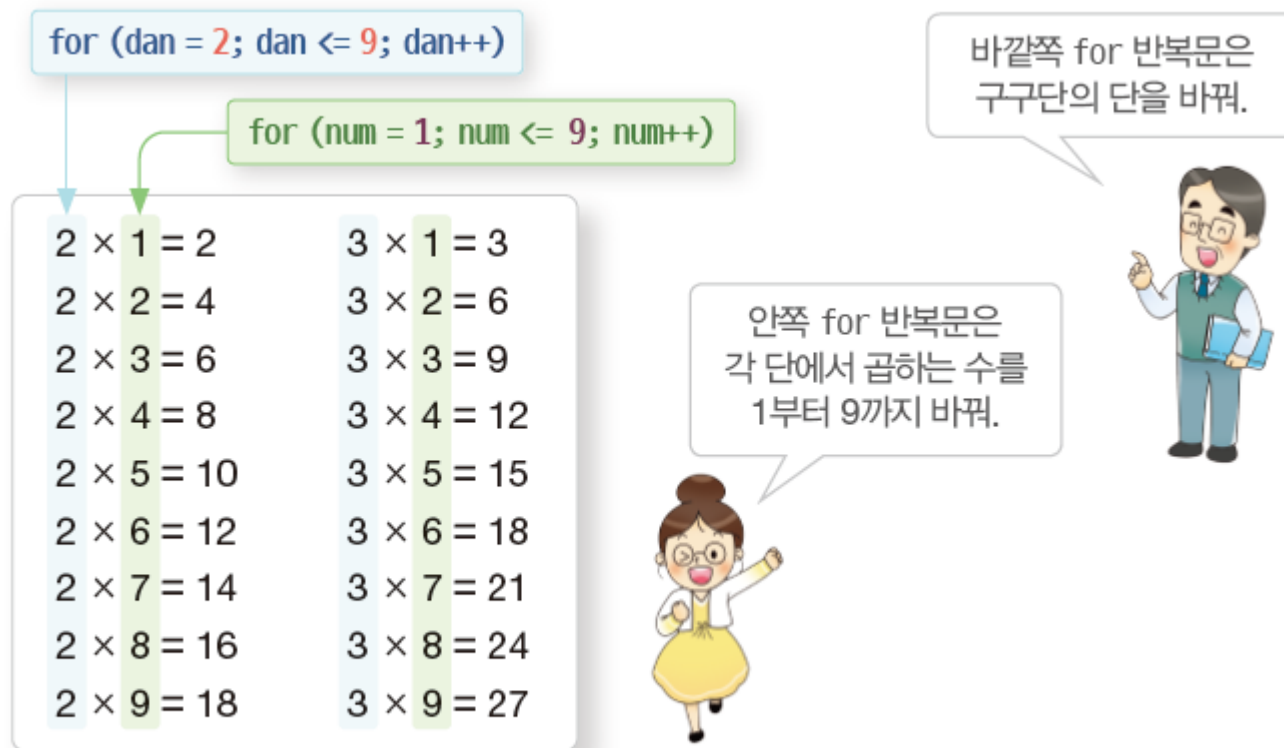


그림 6-17 구구단의 다중 반복문 구조



## 02 조건문과 반복문

### 예제 6-11

### 다중 반복문으로 구구단 출력하기

```
</head>
<body>
  <script>
    let dan, num;
    for (dan = 2; dan < 10; dan++){
      document.write(dan + "단: ");
      for (num = 1; num < 10; num++){
        document.write(dan + "X" + num + "=" + dan * num + " ");
      }
      document.write("<hr>");
    }
  </script>
</body>
</html>
```

2단: 2X1=2 2X2=4 2X3=6 2X4=8 2X5=10 2X6=12 2X7=14 2X8=16 2X9=18

3단: 3X1=3 3X2=6 3X3=9 3X4=12 3X5=15 3X6=18 3X7=21 3X8=24 3X9=27

4단: 4X1=4 4X2=8 4X3=12 4X4=16 4X5=20 4X6=24 4X7=28 4X8=32 4X9=36

5단: 5X1=5 5X2=10 5X3=15 5X4=20 5X5=25 5X6=30 5X7=35 5X8=40 5X9=45

6단: 6X1=6 6X2=12 6X3=18 6X4=24 6X5=30 6X6=36 6X7=42 6X8=48 6X9=54

7단: 7X1=7 7X2=14 7X3=21 7X4=28 7X5=35 7X6=42 7X7=49 7X8=56 7X9=63

8단: 8X1=8 8X2=16 8X3=24 8X4=32 8X5=40 8X6=48 8X7=56 8X8=64 8X9=72

9단: 9X1=9 9X2=18 9X3=27 9X4=36 9X5=45 9X6=54 9X7=63 9X8=72 9X9=81

## 02 조건문과 반복문

### ■ 반복문

#### 5. 반복문에서 유용한 키워드

- **break 문**

예제 6-12 break를 사용하여 구구단 출력하기

```
<script>
  let dan, num;
  for (dan = 2; dan < 10; dan++){
    document.write(dan + "단: ");
    for (num = 1; num < 10; num++){
      if (num == 5) break;
      document.write(dan + "X" + num + "=" + dan * num + " ");
    }
    document.write("<hr>");
  }
</script>
```

2단: 2X1=2 2X2=4 2X3=6 2X4=8

3단: 3X1=3 3X2=6 3X3=9 3X4=12

4단: 4X1=4 4X2=8 4X3=12 4X4=16

5단: 5X1=5 5X2=10 5X3=15 5X4=20

6단: 6X1=6 6X2=12 6X3=18 6X4=24

7단: 7X1=7 7X2=14 7X3=21 7X4=28

8단: 8X1=8 8X2=16 8X3=24 8X4=32

9단: 9X1=9 9X2=18 9X3=27 9X4=36

// 추가된 코드

## 02 조건문과 반복문

### ■ 반복문

#### 5. 반복문에서 유용한 키워드

- **continue 문**

예제 6-13 continue를 사용하여 구구단 출력하기

```
<script>
  let dan, num;
  for (dan = 2; dan < 10; dan++) {
    document.write(dan + "단: ");
    for (num = 1; num < 10; num++) {
      if (num == 5) continue;
      document.write(dan + "X" + num + "=" + dan * num + " ");
    }
    document.write("<hr>");
  }
</script>
</body>
</html>
```

// 추가된 코드

2단: 2X1=2 2X2=4 2X3=6 2X4=8 2X6=12 2X7=14 2X8=16 2X9=18  
3단: 3X1=3 3X2=6 3X3=9 3X4=12 3X6=18 3X7=21 3X8=24 3X9=27  
4단: 4X1=4 4X2=8 4X3=12 4X4=16 4X6=24 4X7=28 4X8=32 4X9=36  
5단: 5X1=5 5X2=10 5X3=15 5X4=20 5X6=30 5X7=35 5X8=40 5X9=45  
6단: 6X1=6 6X2=12 6X3=18 6X4=24 6X6=36 6X7=42 6X8=48 6X9=54  
7단: 7X1=7 7X2=14 7X3=21 7X4=28 7X6=42 7X7=49 7X8=56 7X9=63  
8단: 8X1=8 8X2=16 8X3=24 8X4=32 8X6=48 8X7=56 8X8=64 8X9=72  
9단: 9X1=9 9X2=18 9X3=27 9X4=36 9X6=54 9X7=63 9X8=72 9X9=81

**03**

# **함수와 변수의 범위**

## 03 함수와 변수의 범위

### ■ 함수

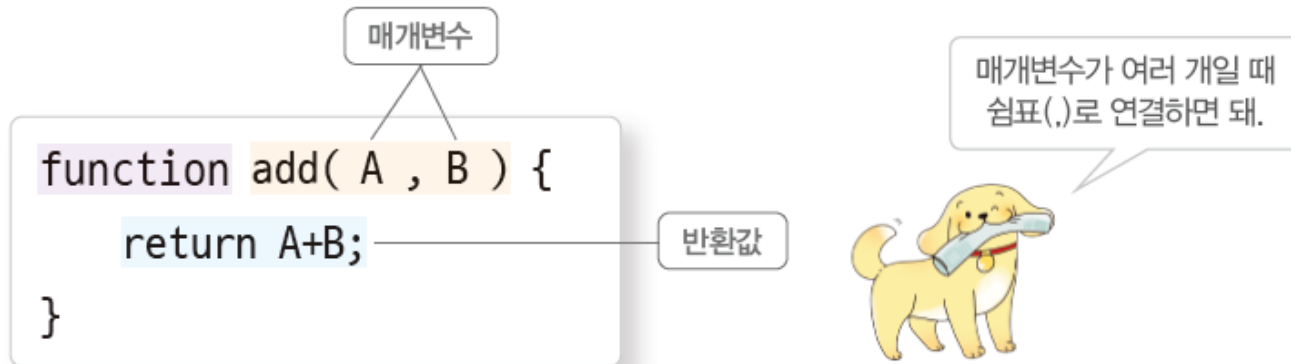


그림 6-18 사용자 정의 함수의 구조

## 03 함수와 변수의 범위

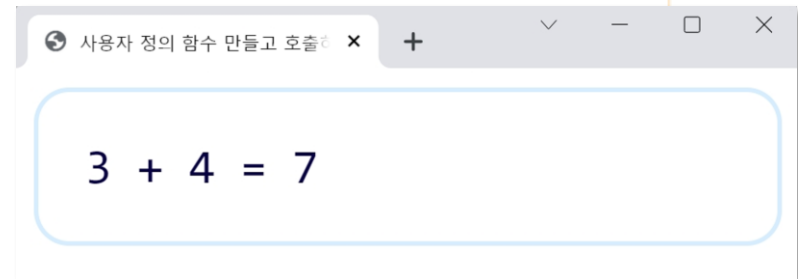
### ■ 함수

#### 1. 사용자 정의 함수

예제 6-14 사용자 정의 함수 만들고 호출하기

ex6-14.html

```
<script>
  function add(num1, num2) {
    return num1 + num2;
  }
  let a = 3, b = 4;
  const sum = add(a, b);      // add() 함수 호출
  document.write(a + " + " + b + " = " + sum);
</script>
</body>
</html>
```



## 03 함수와 변수의 범위

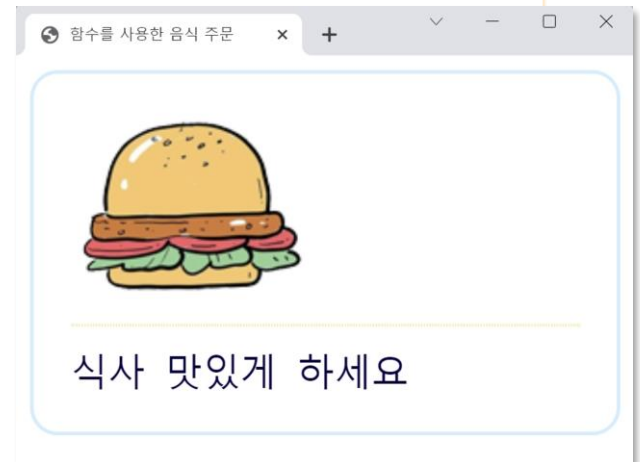
- 사용자 정의 함수의 활용

예제 6-15

함수를 사용한 음식 주문하기

ex6-15.html

```
<script>
function makeFood(num) {
  if (num >= 1 && num <= 3 ) {
    document.write("<img src='../media/food' + num + ".png'>");
    return true;
  } else return false;
}
const food = parseInt(prompt("햄버거=1, 피자=2, 감자튀김=3"));
const result = makeFood(food);
if (result)
  document.write("<hr>식사 맛있게 하세요");
else
  document.write("없는 음식을 주문하셨습니다");
</script>
</body>
</html>
```



## 03 함수와 변수의 범위

### ■ 함수

#### 1. 사용자 정의 함수

- 이름 없는 함수
- 화살표 함수

표 6-16 이름 없는 함수와 화살표 함수의 비교

이름 없는 함수	화살표 함수
<pre>let a = 3, b = 4; let sum = function (num1, num2) {   return num1 + num2 }</pre>	<pre>let a = 3, b = 4; let sum = (num1, num2) =&gt; num1 + num2;</pre>



## 03 함수와 변수의 범위

### ■ 함수

#### 1. 사용자 정의 함수

표 6-17 화살표 함수의 매개변수 지정

매개변수	표기	내용
없을 때	<code>() =&gt; {}</code>	괄호만 사용
1개일 때	<code>x =&gt; {}</code>	괄호 생략 가능
2개 이상일 때	<code>(x, y) =&gt; {}</code>	함수와 동일

표 6-18 화살표 함수의 몸통 규칙

코드의 줄 수	표기	내용
2줄 이상일 때	<code>x =&gt; { let r = x + x; return r; }</code>	함수와 동일
1줄일 때	<code>x =&gt; { x + x; }</code>	return이 없어도 결과를 자동으로 반환
	<code>x =&gt; x + x;</code>	중괄호 생략 가능

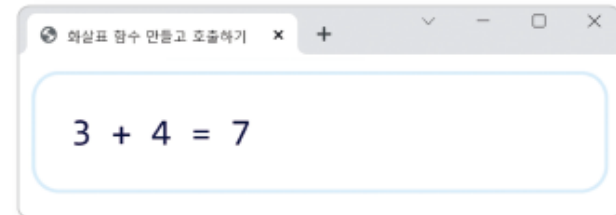
## 03 함수와 변수의 범위

예제 6-16

화살표 함수 만들고 호출하기

ex6-16.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>화살표 함수 만들고 호출하기</title>
  <link rel="stylesheet" href="style_js.css">
</head>
<body>
  <script>
    let a = 3, b = 4;
    let sum = (num1, num2) => num1 + num2;
    document.write(a + " + " + b + " = " + sum(a, b));
  </script>
</body>
</html>
```



## 03 함수와 변수의 범위

### ■ 함수

#### 1. 사용자 정의 함수

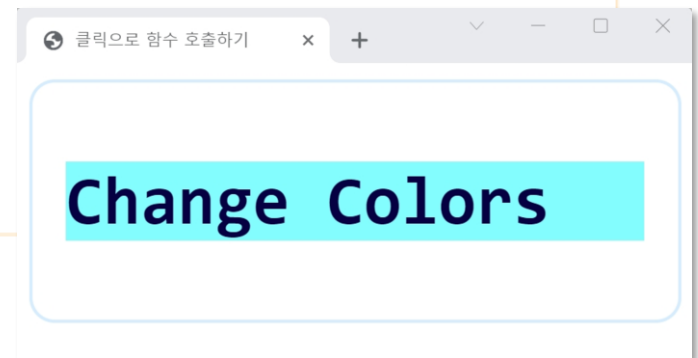
- 클릭으로 함수 호출하기

예제 6-17

클릭으로 함수 호출하기

ex6-17.html

```
<script>
  function change() {
    const alter = document.getElementById("alter");
    alter.style.background = "aqua";
  }
</script>
</body>
</html>
```



## 03 함수와 변수의 범위

### 2. 함수의 입력

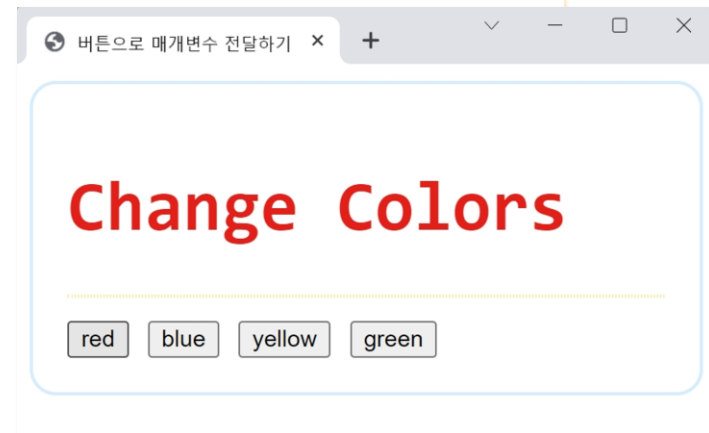
- 버튼으로 함수에 매개변수 전달하기

예제 6-18

버튼으로 매개변수 전달하기

ex6-18.html

```
</head>
<body>
  <h1 id="alter">Change Colors</h1>
  <button onclick="change('red')">red</button>
  <button onclick="change('blue')">blue</button>
  <button onclick="change('yellow')">yellow</button>
  <button onclick="change('green')">green</button>
  <script>
    function change(colors) {
      const alter = document.getElementById("alter");
      alter.style.color = colors;
    }
  </script>
</body>
</html>
```



## 03 함수와 변수의 범위

### 2. 함수의 입력

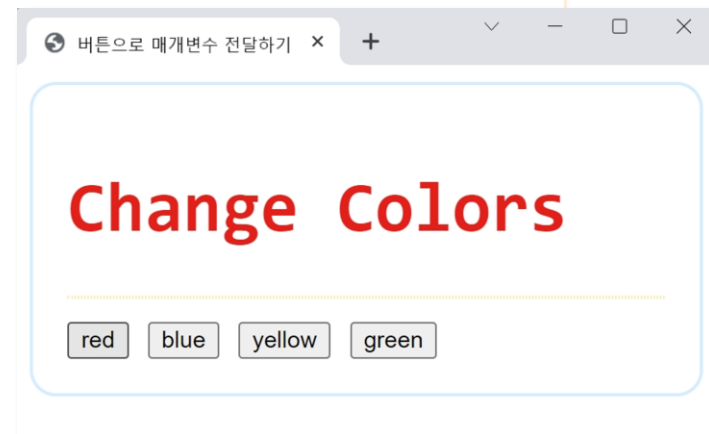
- 버튼으로 함수에 매개변수 전달하기

예제 6-18

버튼으로 매개변수 전달하기

ex6-18.html

```
</head>
<body>
  <h1 id="alter">Change Colors</h1>
  <button onclick="change('red')">red</button>
  <button onclick="change('blue')">blue</button>
  <button onclick="change('yellow')">yellow</button>
  <button onclick="change('green')">green</button>
  <script>
    function change(colors) {
      const alter = document.getElementById("alter");
      alter.style.color = colors;
    }
  </script>
</body>
</html>
```



## 03 함수와 변수의 범위

### 2. 함수의 입력

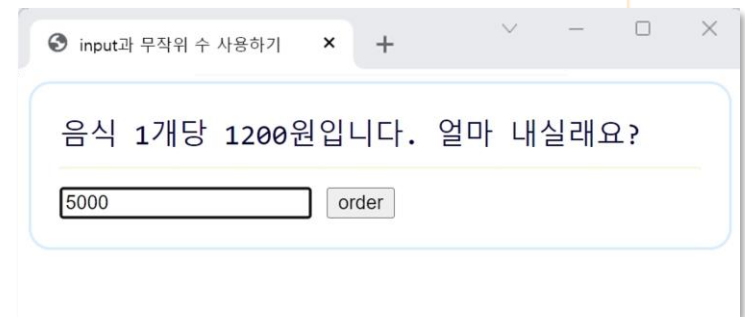
- HTML의 입력 태그와 함수 연결하기

예제 6-19

input과 무작위 수 사용하기

ex6-19.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>input과 무작위 수 사용하기</title>
  <link rel="stylesheet" href="style_js.css">
</head>
<body>
  음식 1개당 1200원입니다. 얼마 내실래요?<br>
  <input type="text" id="input">
  <button onclick="randomFood()">order</button>
```

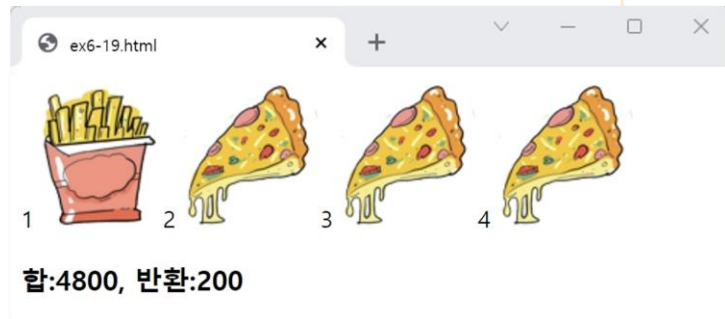


## 03 함수와 변수의 범위

### 2. 함수의 입력

- HTML의 입력 태그와 함수 연결하기

```
<script>
function randomFood() {
  const money = parseInt(document.getElementById('input').value);
  const fnum = Math.floor(money / 1200);
  const change = money - (fnum * 1200);
  if (money >= 1200) {
    for (let i = 1; i <= fnum ; i++) {
      food = Math.floor(Math.random() * 3) + 1;
      document.write(i + "<img src='../media/food' + food + ".png'>");
    }
    document.write("<h3>합:" + fnum * 1200 + ", 반환:" + change + "</h3>");
  } else
    document.write("금액이 모자랍니다");
}
</script>
</body>
</html>
```



## 03 함수와 변수의 범위

### ■ 변수의 범위

#### 1. 전역변수와 지역변수

예제 6-20

전역변수와 지역변수 차이 알아보기

ex6-20.html

```
<script>
  let gb = "전역";
  function test() {
    let lo = "지역";
    document.write("함수 안 gb = " + gb + "<hr>");
    document.write("함수 안 lo = " + lo + "<hr>");
  }
  test();
  document.write("함수 바깥 gb = " + gb + "<hr>");
  document.write("함수 바깥 lo = " + lo + "<hr>");
</script>
</body>
</html>
```



## 03 함수와 변수의 범위

### ■ 변수의 범위

#### 1. 전역변수와 지역변수

```
function test() {  
  for (let num = 1; num < 10; num++) {  
    // num 변수 사용 가능  
  }  
  // num 변수 사용 불가능  
}
```

## 03 함수와 변수의 범위

### ■ 변수의 범위

#### 2. 템플릿 문자열과 변수

```
let str = "yes";  
document.write("결과 : " + str + " 오늘부터 1일")
```

```
let str = "yes";  
document.write(`결과 : ${str} 오늘부터 1일`)
```

## 03 함수와 변수의 범위

### ■ 변수의 범위

#### 2. 템플릿 문자열과 변수

예제 6-22

템플릿과 변수 같이 사용하기

ex6-22.html

```
<script>
  let drink = prompt(`한 잔 따르다.
"이거 마시면 사귀는 거다!"`);
  if (drink == "yes") {
    document.write(`답:${drink} 오늘부터 1일`);
  } else {
    document.write(`답:${drink} 헤어진다`);
  }
</script>
</body>
</html>
```