

Chapter 09

자바스크립트 기본 문법

웹프로그래밍

AI컴퓨터공학부
정경용 교수

목차

1. 자바스크립트 기본 용어와 출력
2. 자료형과 변수
3. 조건문과 반복문
4. 함수
5. 객체

학습목표

- 자바스크립트의 기본 용어를 이해합니다.
- 자바스크립트 기본 자료형과 연산자를 사용하는 방법을 익힙니다.
- 조건문과 반복문을 사용해 자바스크립트 프로그램 흐름을 제어하는 방법을 익힙니다.
- 함수를 선언하고 호출하는 방법을 이해합니다.
- 객체의 기본 개념을 이해하고 속성과 메서드를 구분할 수 있습니다.

Section 01
자바스크립트
기본 용어와 출력 방법

01. 자바스크립트 기본 용어와 출력 방법

1. 자바스크립트 기본 용어

표현식과 문장

- 표현식: 값을 만들어 내는 간단한 코드.
- 문장: 프로그래밍 언어에 실행할 수 있는 코드의 최소 단위.
- 문장 마지막에 **세미콜론(;) 또는 줄 바꿈**을 넣어 종결을 나타냄.

```
273;  
10 + 20 + 30 * 2;  
var name = '윤' + '인' + '성';  
alert('Hello JavaScript');
```

(a) 문장 예

```
273  
10 + 20 + 30 * 2  
'JavaScript'
```

(b) 표현식 예

그림 9-2 문장과 표현식 예

01. 자바스크립트 기본 용어와 출력 방법

1. 자바스크립트 기본 용어

키워드

- 키워드: 자바스크립트를 처음 만들 때 정해진 특별한 의미가 부여된 단어

표 9-1 자바스크립트 키워드

break	else	instanceof	true	case	false
new	try	catch	finally	null	typeof
continue	for	return	var	default	function
switch	void	delete	if	this	while
debugger	import	export	extends	super	yield
do	in	throw	with	const	class

01. 자바스크립트 기본 용어와 출력 방법

1. 자바스크립트 기본 용어

식별자

- 식별자: 자바스크립트에서 변수나 함수 등에 이름을 붙일 때 사용하는 단어

사용할 수 있는 예

alpha
alpha10
_alpha
\$alpha
AlPha
ALPHA

O

사용할 수 없는 예

break
273alpha
has space

X

- 1 생성자 함수 이름은 항상 대문자로 시작합니다.
- 2 변수, 인스턴스, 함수, 메서드의 이름은 항상 소문자로 시작합니다.
- 3 여러 단어로 된 식별자는 각 단어의 첫 글자를 대문자로 합니다.

그림 9-3 식별자 예

i love you → iLoveYou
i am a boy → iAmABoy
create server → createServer

글자가 모두 붙어 있지만 대·소문자로 구분했으므로
쉽게 끊어서 읽을 수 있습니다.

그림 9-4 식별자 생성 관례

01. 자바스크립트 기본 용어와 출력 방법

1. 자바스크립트 기본 용어

식별자

표 9-2 자바스크립트의 식별자 종류

구분	단독으로 사용	다른 식별자와 함께 사용
식별자 뒤에 괄호 없음	변수	속성
식별자 뒤에 괄호 있음	함수	메서드

<code>alert('Hello World')</code>	→ 함수
<code>Array.length</code>	→ 속성
<code>input</code>	→ 변수
<code>prompt('Message', 'Defstr')</code>	→ 함수
<code>Math.PI</code>	→ 속성
<code>Math.abs(-273)</code>	→ 메서드

그림 9-5 식별자 구분

01. 자바스크립트 기본 용어와 출력 방법

1. 자바스크립트 기본 용어

주석 : 프로그램 진행에 전혀 영향을 주지 않는 코드로, 프로그램 설명에 사용

표 9-3 주석 처리 방법

방법	형태
① 한 행 주석 처리	// 주석문
② 여러 행 주석 처리	/* 주석문 주석문 */

```
<script>  
    // 주석은 코드 실행에 영향을 주지 않습니다.  
    /*  
    alert('Hello JavaScript .. !');  
    alert('Hello JavaScript .. !');  
    alert('Hello JavaScript .. !');  
    */  
</script>
```

01. 자바스크립트 기본 용어와 출력 방법

2. 자바스크립트 출력

- 웹 브라우저에 경고 창 띄우기
 - 가장 기본적인 출력 방법 - alert() 함수 사용

기본 예제 9-1 자바스크립트를 이용한 메시지 출력

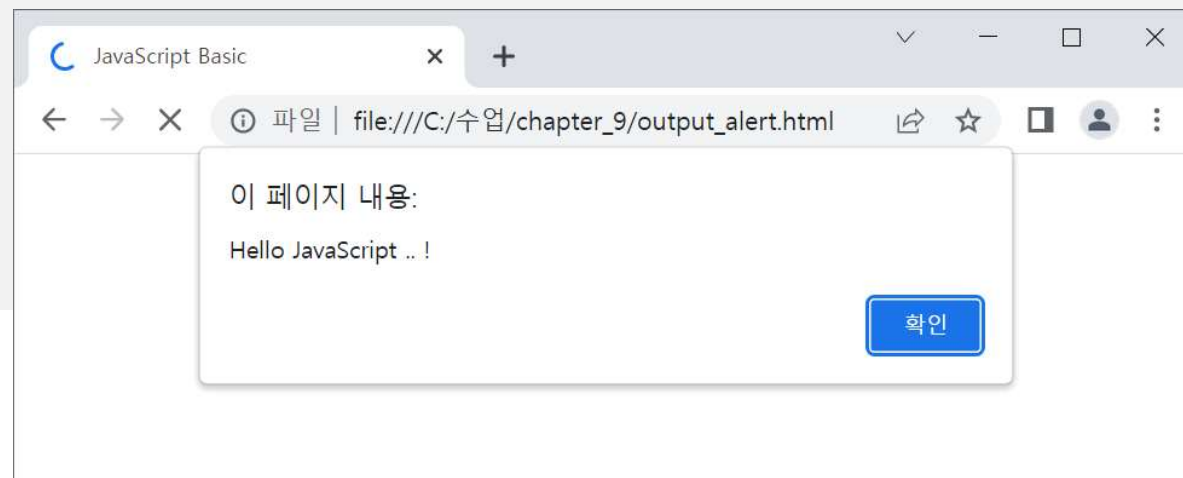
코드 9-1 output_alert.html

```
<!DOCTYPE html>
<html>
<head>
  <title>JavaScript Basic</title>
  <script>
    alert('Hello JavaScript .. !');
  </script>
</head>
<body>

</body>
</html>
```

alert("메시지")

그림 9-6 alert() 함수 형태



01. 자바스크립트 기본 용어와 출력 방법

2. 자바스크립트 출력: 확인 다이얼로그

■ `confirm("메시지")` 함수

- "메시지"를 출력하고 '확인/최소(OK/CANCEL)'버튼을 가진 다이얼로그 출력
- '확인' 버튼을 누르면 `true`, '취소' 버튼이나 강제로 다이얼로그를 닫으면 `false` 리턴

```
var ret = confirm("전송할까요");  
if(ret == true) {  
    // 사용자가 "확인" 버튼을 누른 경우  
}  
else {  
    // 취소 버튼이나 다이얼로그를 닫은 경우  
}
```

이 페이지 내용:
전송할까요

확인

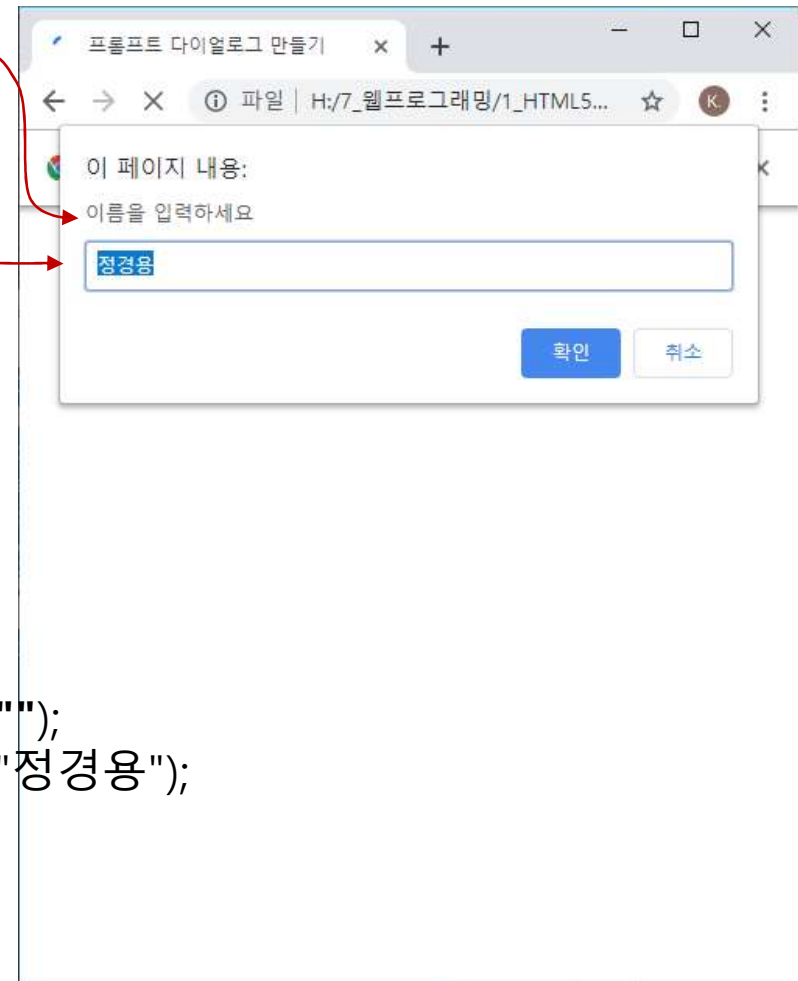
취소

01. 자바스크립트 기본 용어와 출력 방법

2. 자바스크립트 출력: 프롬프트 다이얼로그

- **prompt("메시지", "디폴트 입력값")** 함수
 - 사용자로부터 문자열을 입력 받아 리턴

```
var ret = prompt("이름을 입력하세요", "정경용");  
  
if(ret == null) {  
    // 취소 버튼이나 다이얼로그를 닫은 경우  
}  
else if(ret == "") {  
    // 문자열 입력 없이 확인 버튼 누른 경우  
}  
else {  
    // ret에는 사용자가 입력한 문자열  
}
```



```
var n = prompt("이름을 입력하세요", "");  
var n = prompt("이름을 입력하세요", "정경용");
```

01. 자바스크립트 기본 용어와 출력 방법

2. 자바스크립트 출력

NOTE 크롬 개발자 도구를 이용한 실행 결과 확인

- 크롬 웹 브라우저에서 [F12].
- 크롬 브라우저 개발자 도구 [Console] 탭에 숫자를 입력하면 자동 생성됨.

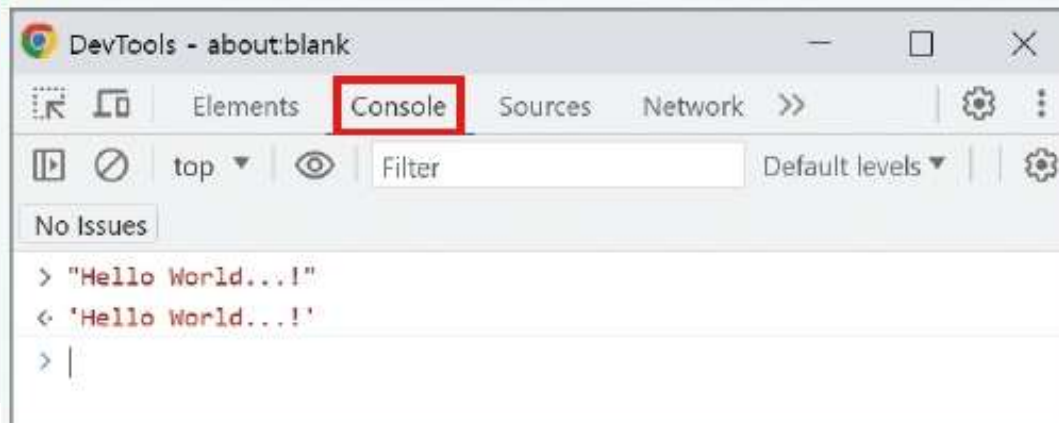


그림 9-7 크롬 개발자 도구 [Console] 탭

Section 02

자료형과 변수

02. 자료형과 변수

1. 자료형

숫자

- 크롬 브라우저 개발자 도구 [Console] 탭에 숫자를 입력하면 자동 생성됨.
- 정수와 실수 구분하지 않음. 가장 기본적인 자료형

표 9-4 사칙 연산자

연산자	설명	예	연산자	설명	예
+	덧셈	<pre>> 52 + 273 325 > 52.273 + 103.57 155.843</pre>	/	나눗셈	<pre>> 52 / 273 0.19047619047619047 > 52.273 / 103.57 0.504711789128126 > 1 / 0 Infinity</pre>
-	뺄셈	<pre>> 52 - 273 -221 > 52.273 - 103.57 -51.29699999999999</pre>			
*	곱셈	<pre>> 52 * 273 14196 > 52.273 * 103.57 5413.91461</pre>			

부동소수점을 계산할 때
약간의 오차 발생.

숫자를 0으로 나누었을 때,
무한을 나타내는 값이 됨.

02. 자료형과 변수

1. 자료형

숫자

- 나머지 연산자: 좌변을 우변으로 나눈 나머지를 표시.

표 9-5 나머지 연산자

연산자	설명	예
%	나머지	<pre>> 10 % 5 0 > 7 % 3 1</pre>

02. 자료형과 변수

1. 자료형

- 문자열: 문자 집합
- 'abcdefg', 'Hello World', '안녕하세요.'

표 9-6 문자열 생성

방법	예
작은따옴표 사용	<pre>> 'Hello JavaScript .. !' "Hello JavaScript .. !" > '"문자열"입니다.'</pre>
큰따옴표 사용	<pre>> "Hello JavaScript .. !" "Hello JavaScript .. !" > "'문자열'입니다."</pre>

02. 자료형과 변수

1. 자료형

- 문자열

표 9-7 이스케이프 문자

이스케이프 문자	설명	예
\t	수평 탭	> '한빛\t아카데미' "한빛 아카데미"
\n	행 바꿈	> '한빛\n아카데미' "한빛 아카데미"
\\	역 슬래시	> '\\\\' \\"
\'	작은따옴표	> '\\\'' ''''
\"	큰따옴표	> '\"\"\"' \"\"\"\"

02. 자료형과 변수

1. 자료형

- 문자열

표 9-8 문자열 연결 연산자

연산자	설명	예
+	문자열 연결	> '가나다' + '라마' + '바사아' + '자차카타' + '파하' "가나다라마바사아자차카타파하"

- 불(bool) : 참과 거짓을 표현할 때 사용하는 자료

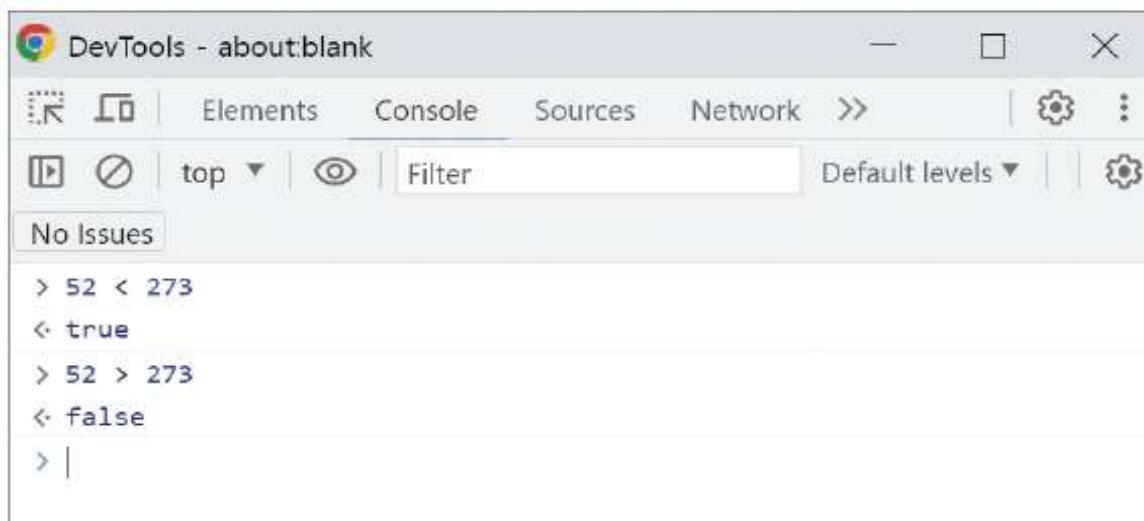


그림 9-8 불 사용 예

02. 자료형과 변수

1. 자료형

- 논리 연산자

표 9-9 비교 연산자

연산자	설명	예
>=	좌변이 우변보다 크거나 같음	<pre>> 10 >= 20 false > '가방' >= '하마' false</pre> <div>문자열 순서도 비교 가능합니다.</div>
<=	우변이 좌변보다 크거나 같음	<pre>> 10 <= 20 true</pre>
>	좌변이 우변보다 큼	<pre>> 10 > 20 false</pre>
<	우변이 좌변보다 큼	<pre>> 10 < 20 true</pre>
==	좌변과 우변이 같음	<pre>> 10 == 20 false</pre>
!=	좌변과 우변이 다름	<pre>> 10 != 20 true</pre>

02. 자료형과 변수

1. 자료형

- 논리 연산자

표 9-10 논리 연산자

연산자	설명	예
!	논리 부정(참이면 거짓, 거짓이면 참)	<pre>> !true false > !(10 == 10) false</pre>
&&	논리곱(둘 다 참이어야 참)	<pre>> true && true true > true && false false > false && true false > false && false false > (10 < 20) && (20 < 30) true</pre>

둘 다 참일 때만 참.

20이 10과 30 사이에 있다는 의미

02. 자료형과 변수

1. 자료형

- 논리 연산자

연산자	설명	예
	논리합(둘 중 하나만 참이어도 참)	<pre>> true true true > true false true > false true true > false false false</pre> <div>둘 중 하나만 참이어도 참.</div>

02. 자료형과 변수

2. 변수

- 변수: 값을 저장할 때 사용하는 식별자
 - ❶ 변수를 선언
 - ❷ 변수에 값을 할당

```
❶ > let pi;           // 변수 선언
    undefined
❷ > pi = 3.14159265;   // 값 할당
    undefined
```

- 변수 초기화

```
> let pi = 3.14159265;
undefined
```

02. 자료형과 변수

2. 변수

- 변수: 값을 저장할 때 사용하는 식별자

```
> let pi = 3.14159265;  
undefined  
> alert(pi);  
undefined
```



그림 9-9 변수에 저장된 값 출력

02. 자료형과 변수

2. 변수

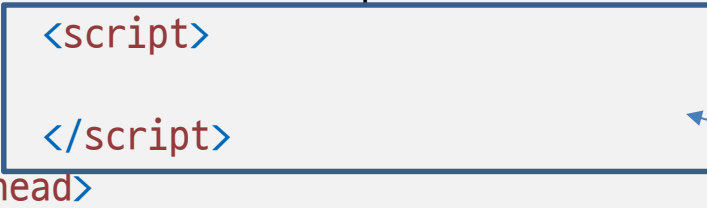
기본 예제 9-2 변수의 선언과 초기화

1 HTML 페이지 만들기

코드 9-2 HTMLPage.html

```
<!DOCTYPE html>
<html>
<head>
  <title>JavaScript Basic</title>
  <script>
    </script>
</head>
<body>

</body>
</html>
```



02. 자료형과 변수

2. 변수

기본 예제 9-2 변수의 선언과 초기화

2 변수 사용하기

코드 9-3 variable_use.html

```
<script>
  // 변수를 선언 및 초기화합니다.
  let radius = 10;
  let pi = 3.14159265;

  // 출력합니다.
  alert(2 * radius * pi);
</script>
```

이 페이지 내용:

62.831853

확인

02. 자료형과 변수

2. 변수

NOTE const 키워드

let을 붙인 식별자는 변수, const를 붙인 식별자는 상수.

```
let a = 10  
const b = 10
```

[코드 9-3]을 다음과 같이 변경.

```
<script>  
  // radius와 pi가 이후로 바뀌지 않으므로  
  // let을 const로 변경합니다.  
  const radius = 10;  
  const pi = 3.14159265;  
  
  alert(2 * radius * pi)  
</script>
```

Section 03

조건문과 반복문

03. 조건문과 반복문

1. 조건문

- If 조건문
 - 조건이 true이면 문장을 실행하고 false이면 문장 무시
 - 실행 문장이 한 행이면 중괄호 생략 가능함
 - 실행 문장이 여러 행이라면 중괄호 필요함

```
if (조건) {  
    문장  
}
```

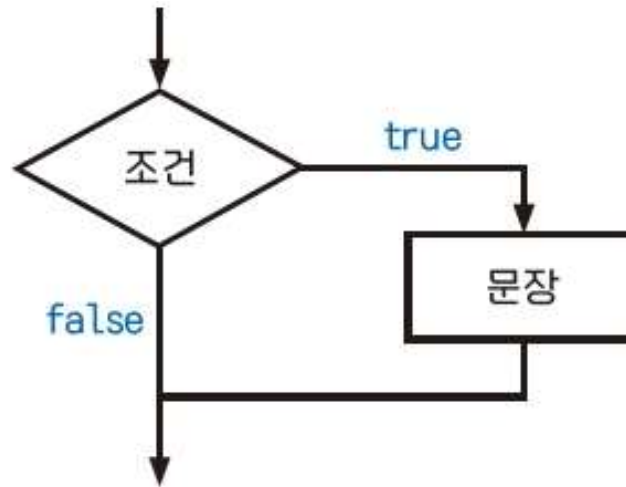


그림 9-10 if 조건문 기본 형태와 순서도

03. 조건문과 반복문

1. 조건문

기본 예제 9-3 if 조건문으로 참과 거짓 판별

코드 9-4 output_alert.html

```
<script>
  // 조건문
  if (273 < 52) {
    // 표현식 "273 < 52"가 참일 때 실행합니다.
    alert('273 < 52 => true');
  }
  // 프로그램 종료
  alert('프로그램 종료');
</script>
```

이 페이지 내용:

프로그램 종료

확인

03. 조건문과 반복문

1. 조건문

기본 예제 9-4 if 조건문으로 오전과 오후 판별

1 현재 시간 구하기

코드 9-5 condition_getDate.html

```
<script>
// Date 객체를 선언합니다.
let date = new Date();
// 요소를 추출합니다.
let year = date.getFullYear();
let month = date.getMonth() + 1;
let day = date.getDay();
let hours = date.getHours();
let minutes = date.getMinutes();
let seconds = date.getSeconds();
</script>
```

우리에게 익숙한 월 표현으로
변경하기 위해 1씩 더함.

03. 조건문과 반복문

1. 조건문

기본 예제 9-4 if 조건문으로 오전과 오후 판별

2 오전과 오후 구분하기

코드 9-6 condition_date.html

```
<script>
  // 변수를 선언합니다.
  let date = new Date();
  let hours = date.getHours();
  // 조건문
  if (hours < 12) {
    // 표현식 "hours < 12"가 참일 때 실행합니다.
    alert('오전입니다.');
  }
  if (12 <= hours) {
    // 표현식 "12 <= hours"가 참일 때 실행합니다.
    alert('오후입니다.');
  }
</script>
```

이 페이지 내용:
오후입니다.

확인

03. 조건문과 반복문

1. 조건문

- If else 조건문
 - 두 가지로 분명하게 나뉘는 상황에서 편리하게 사용 가능
 - 실행 문장이 한 행이면 중괄호 생략 가능함
 - 실행 문장이 여러 행이라면 중괄호 필요함

```
if (조건) {  
    문장 1  
} else {  
    문장 2  
}
```

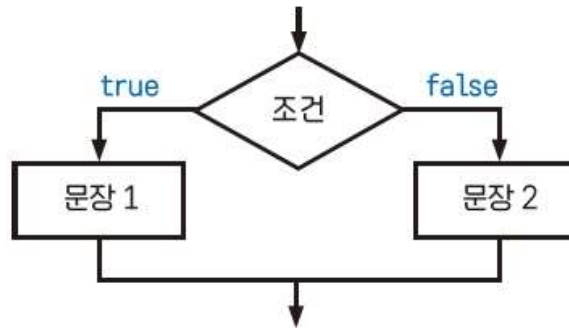


그림 9-11 if else 조건문 기본 형태와 순서도

<h3>조건 연산</h3> <hr>

```
<script>  
    var a=3, b=5;  
    document.write("a=" + a + ", b=" + b + "<br><br>");  
    document.write("두수의 차이 : " + ((a>b)?(a-b):(b-a)));  
</script>
```

조건 연산

a=3, b=5

두수의 차이 : 2

03. 조건문과 반복문

1. 조건문

기본 예제 9-5 if else 조건문으로 오전과 오후 판별

- if else 조건문

코드 9-7 condition_else.html

```
<script>
// 변수를 선언합니다.
let date = new Date();
let hours = date.getHours();

// 조건문
if (hours < 12) {
    // 표현식 "hours < 12"가 참일 때 실행합니다.
    alert('오전입니다. ');
} else {
    // 표현식 "hours < 12"가 거짓일 때 실행합니다.
    alert('오후입니다. ');
}
</script>
```

조건을 한 번만 비교해서
두 경우를 처리하므로 효율적.

[실습] '9시 28분 오전입니다.' 메시지가 나올 수 있도록 프로그램하시오.

03. 조건문과 반복문

1. 조건문

- 중첩 조건문과 if else if 조건문

```
if (조건) {  
    if (조건) {  
        문장  
    } else {  
        문장  
    }  
} else {  
    if (조건) {  
        문장  
    } else {  
        문장  
    }  
}
```

그림 9-12 중첩 조건문 형태

03. 조건문과 반복문

1. 조건문

기본 예제 9-6 / 중첩 조건문으로 하루 일정 표현

코드 9-8 condition_duplication.html

```
<script>
  // Date 객체를 선언합니다: 현재 시간 측정
  let date = new Date();
  let hours = date.getHours();

  // 조건문
  if (hours < 5) {
    alert('잠을 자렴....');
  } else {
    if (hours < 7) {
      alert('준비');
    } else {
      if (hours < 9) {
        alert('출근');
      }
    }
  }
}
```

03. 조건문과 반복문

1. 조건문

기본 예제 9-6

중첩 조건문으로 하루 일정 표현

```
    } else {  
        if (hours < 12) {  
            alert('빈둥빈둥');  
        } else {  
            if (hours < 14) {  
                alert('식사');  
            } else {  
                // 여러 가지 업무를 수행합니다.  
            }  
        }  
    }  
}  
}  
}  
</script>
```

이 페이지 내용:

빈둥빈둥

확인

03. 조건문과 반복문

1. 조건문

- if else if 조건문:
 - 중복되지 않는 조건 세 가지 이상을 구분할 때 사용
 - 중첩 조건문의 종괄호를 생략했을 때 만드는 조건문

```
if (조건) {  
    문장  
} else if (조건) {  
    문장  
} else if (조건) {  
    문장  
} else {  
    문장  
}
```

그림 9-13 if else if 조건문 형태

03. 조건문과 반복문

1. if else if 조건문

기본 예제 9-7

if else if 조건문으로 하루 일정 표현

코드 9-9

condition_ifelseif.html

```
<script>
  // Date 객체를 선언합니다: 현재 시간 측정
  let date = new Date();
  let hours = date.getHours();

  // 조건문
  if (hours < 5) {
    alert('잠을 자렴....');
  } else if (hours < 7) {
    alert('준비');
  } else if (hours < 9) {
    alert('출근');
  } else if (hours < 12) {
    alert('빈둥빈둥');
  } else if (hours < 14) {
    alert('식사');
  } else {
    // 여러 가지 업무를 수행합니다.
  }
</script>
```

03. 조건문과 반복문

1. 논리 연산자와 조건문

NOTE 논리연산자와 조건문

코드 9-10 note_logic.html

```
<script>
  // Date 객체를 선언합니다: 현재 시간 측정
  let date = new Date();
  let month = date.getMonth() + 1;

  // 조건문
  if (3 <= month && month <= 5) {
    alert('봄입니다. ');
  } else if (6 <= month && month <= 8) {
    alert('여름입니다. ');
  } else if (9 <= month && month <= 11) {
    alert('가을입니다. ');
  } else {
    alert('겨울입니다. ');
  }
</script>
```

이 페이지 내용:
여름입니다.

확인

03. 조건문과 반복문

2. 반복문

```
<script>
  alert('출력');
  alert('출력');
  alert('출력');
  alert('출력');
  alert('출력');
</script>
```

```
<script>
  for (let i = 0; i < 1000; i++) {
    alert('출력');
  }
</script>
```

03. 조건문과 반복문

2. 반복문

배열

- 배열: 변수 여러 개를 한꺼번에 다룰 수 있는 자료형.

```
// 변수를 선언합니다.  
let array = [273, 32, 103, 57, 52];
```

코드 9-11 array_basic.html

```
<script>  
  // 변수를 선언합니다.  
  let array = [273, '문자열', true, function () { }, {}, [32, 103]];  
  alert(array);  
</script>
```

이 페이지 내용:

273,문자열,true,function () { },[object Object],32,103

확인

03. 조건문과 반복문

■ 문자열 연산: document.write()

```
<!DOCTYPE html>
<html>
<head>
<title>문자열 연산</title> </head>
<body>
<h3>문자열 연산</h3>
<hr>

<script>
  document.write("abc" + 23 + "<br>");
  document.write(23 + "abc" + "<br>");
  document.write(23 + "35" + "<br>");
  document.write(23 + 35 + "<br>");
  document.write(23 + 35 + "abc" + "<br>");
  document.write("abc" + 23 + 35 + "<br><hr>");

  var name = "dragon";
  document.write(name == "dragon");
  document.write("<br>");
  document.write(name > "park");
</script>

</body>
</html>
```

문자열 연산

```
abc23
23abc
2335
58
58abc
abc2335
```

```
true
false
```

03. 조건문과 반복문

- 조건문: if-else 사용해서 학점 계산

```
<html>
<head>
<title>if-else</title> </head>
<body>
<h3>if-else를 이용한 학점 매기기</h3>
<hr>
<script>
  var grade;
  var score = prompt("웹프로그래밍 점수를 입력하세요", 100);
  score = parseInt(score); // 문자열을 숫자로 바꿈
```

```
</body>
</html>
```

이 페이지 내용:

웹프로그래밍 점수를 입력하세요

확인

취소

if-else를 이용한 학점 계산

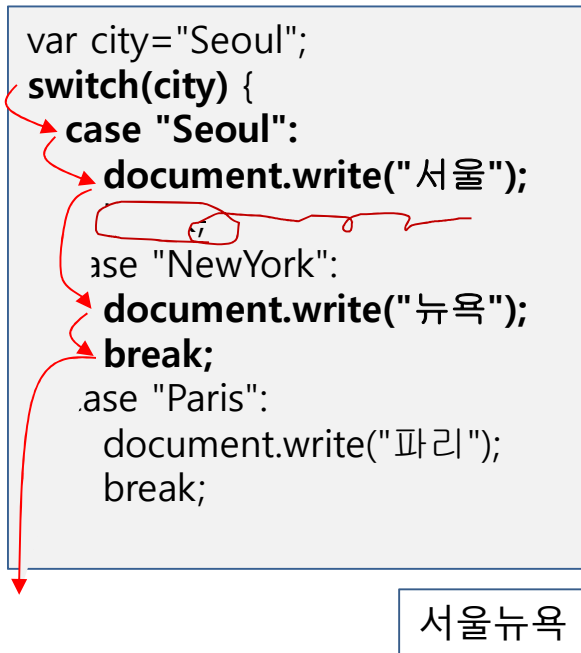
100는 A입니다.

03. 조건문과 반복문

▪ Break문

- switch 문 종료
- break; 문을 만날 때까지 아래로 코드 계속 실행

```
var city="Seoul";  
switch(city) {  
  case "Seoul":  
    document.write("서울");  
    break;  
  case "NewYork":  
    document.write("뉴욕");  
  case "Paris":  
    document.write("파리");  
    break;  
}
```



서울뉴욕

(a) break;를 만날 때까지 아래로 실행을 계속하는 사례

```
var day="월";  
switch(day) {  
  case "월":  
  case "화":  
  case "수":  
  case "목":  
  case "금": document.write("정상영업");  
             break;  
  case "토":  
  case "일": document.write("휴일");  
             break;  
}
```

정상영업

(b) 여러 case에 대해 동일한 코드를 실행하도록 의도적으로 break;를 생략한 경우

03. 조건문과 반복문

■ 조건문: switch 문 사용

```
<html>
<head> <title>switch</title> </head>
<body>
<h3>switch 문으로 커피 주문</h3>
<hr>
<script>
  var price = 0;
  var coffee = prompt("무슨 커피 드릴까요?", "");
  switch(coffee) {
    case "espresso" :
    case "에스프레소" : price = 2000;
      break;
    case "카푸치노" : price = 3000;
      break;
    case "카페라떼" : price = 3500;
      break;
    default :
      document.write(coffee + "는 없습니다.");
  }
  if(price != 0)
    document.write(coffee + "는 " + price + "원입니다.");
</script>
</body>
</html>
```

"espresso"나
"에스프레소"의 경우
모두 실행

이 페이지 내용:
무슨 커피 드릴까요?

확인

취소

switch 문으로 커피 주문

espresso는 2000원입니다.

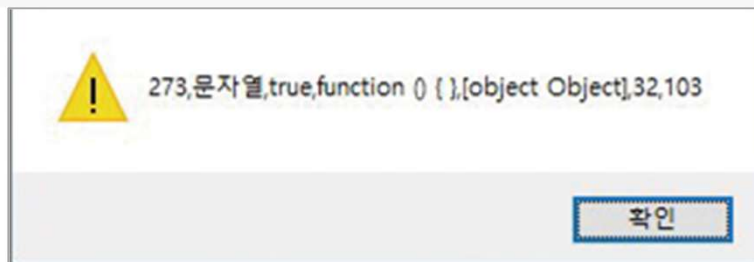
03. 조건문과 반복문

2. 반복문

■ 배열

- 변수 여러 개를 한꺼번에 다룰 수 있는 자료형 (객체의 일종)
- 요소 - 배열 내부에 입력된 자료 하나하나
- 배열 내부에 다양한 자료형을 입력 가능
- 배열 전체를 출력하면 요소가 순서대로 표시

```
<script>  
  // 변수를 선언합니다.  
  var array = [273, '문자열', true, function () { }, {}, [32, 103]];  
  alert(array);  
</script>
```



03. 조건문과 반복문

2. 반복문

기본 예제 9-8 / 배열 생성과 배열 요소 접근

코드 9-12 array_index.html

```
<script>
  // 변수를 선언합니다.
  let array = [273, 32, 103, 57, 52]; <script>

  // 변수를 선언합니다.
  let array = ['가', '나', '다', '라'];
  // 배열 요소를 변경합니다.
  array[0] = '윤';
  // 요소를 출력합니다

  alert(array[0]);
  alert(array[1]);
  alert(array[2]);
  alert(array[3]);
</script>
```

이 페이지 내용:

윤

이 페이지 내용:

나

이 페이지 내용:

다

이 페이지 내용:

라

확인

03. 조건문과 반복문

2. 반복문

NOTE 배열 요소의 개수

`alert(array.length);` // length 속성은 배열 요소의 개수

코드 9-13 note_arrayLength.html

```
<script>
  // 변수를 선언합니다.
  let array = [10, 20, 30, 40, 50];

  // 출력합니다.
  alert(array.length);
</script>
```

이 페이지 내용:

5

확인

03. 조건문과 반복문

2. 반복문

■ while 반복문

- 가장 기본적인 반복문
- 불 표현식이 참이면 중괄호 안 문장을 계속 실행.

```
while (조건) {  
    문장  
}
```

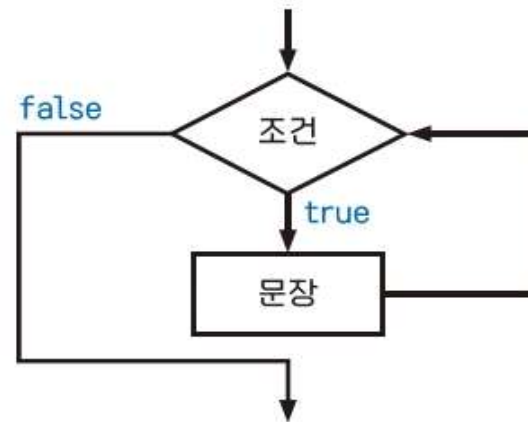


그림 9-14 while 반복문 기본 형태와 순서도

- 조건을 거짓으로 만드는 문장이 없으면 무한 반복.

```
<script>  
    // 반복을 수행합니다.  
    while (true) {  
        alert('무한 반복');  
    }  
</script>
```

03. 조건문과 반복문

2. 반복문

- 특정한 숫자를 증가시켜 불 표현식을 거진으로 만들어 반복문을 벗어남

코드 9-14 loop_while.html

기본 예제 9-9 while 반복문

```
<script>
  // 변수를 선언합니다.
  let i = 0;
  let array = ['가', '나', '다'];

  // 반복을 수행합니다. i가 배열 원소 개수인 3보다 작을 때 반복합니다.
  while (i < array.length) {
    // 출력합니다.
    alert(i + '번째 출력: ' + array[i]);

    // 탈출하려고 변수를 더합니다.
    i++;
  }
</script>
```

이 페이지 내용:

0번째 출력: 가

확인

이 페이지 내용:

1번째 출력: 나

확인

이 페이지 내용:

2번째 출력: 다

확인

03. 조건문과 반복문

2. 반복문

for 반복문

- for 반복문: 원하는 횟수만큼 반복하고 싶을 때 사용
 - for 반복문은 원하는 횟수만큼 반복하고 싶을 때 사용
 - ❶ 초기식을 비교합니다.
 - ❷ 조건식을 비교합니다. 조건이 거짓이면 반복문을 종료합니다.
 - ❸ 문장을 실행합니다.
 - ❹ 종결식을 실행합니다.

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

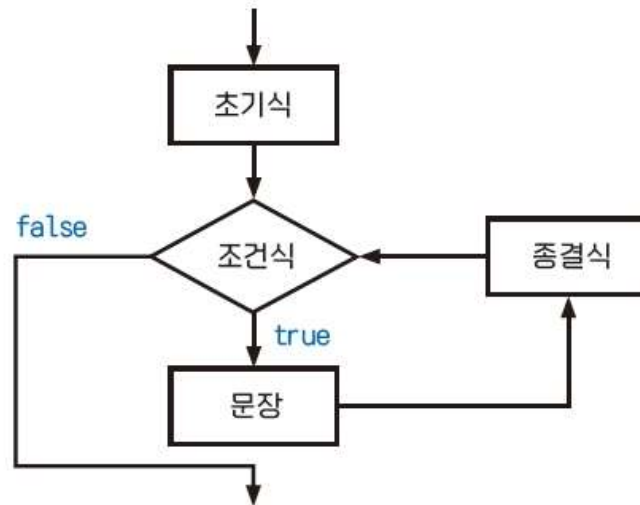


그림 9-15 for 반복문 기본 형태와 순서도

03. 조건문과 반복문

2. 반복문

for 반복문

- for 반복문 단계

- ① 초기식을 비교합니다.
- ② 조건식을 비교합니다. 조건이 거짓이면 반복문을 종료합니다.
- ③ 문장을 실행합니다.
- ④ 종결식을 실행합니다.
- ⑤ ❸ 앞의 ❷ 단계로 이동합니다.

```
for (let i = 0; i < 반복 횟수; i++) {  
  }  
}
```

03. 조건문과 반복문

2. 반복문

기본 예제 9-10 for 반복문

코드 9-15 loop_for.html

```
<script>
  // 변수를 선언합니다.
  let array = ['가', '나', '다'];

  // 반복을 수행합니다.
  for (let i = 0; i < 3; i++) {
    // 출력합니다.
    alert(i + '번째 출력: ' + array[i]);
  }
</script>
```

03. 조건문과 반복문

2. 반복문

NOTE 반복문 조건을 외부 요인으로 변경

코드 9-16 note_changeConditionFromOuterSpace.html

```
<script>
  // 변수를 선언합니다.
  let start = new Date().getTime();
  let count = 0;

  // 반복을 수행합니다.
  while (start + 1000 > new Date().getTime()) {
    count++;
  }

  // 출력합니다.
  alert(count + '만큼 반복했습니다. ');
</script>
```

이 페이지 내용:

6517447만큼 반복했습니다.

확인

03. 조건문과 반복문

2. 반복문

❖ for 문

1 2 4
for(초기문; 조건식; 반복 후 작업) {
... 작업문 ...
3
}

```
// 0에서 9까지 출력  
for(var i=0; i<10; i++) {  
    document.write(i);  
}
```

0123456789

❖ while 문

1
while(조건식) {
... 작업문 ...
2
}

```
var i=0;  
while(i<10) { // i가 0에서 9까지 출력  
    document.write(i);  
    i++;  
}
```

0123456789

❖ do-while 문

1
do {
... 작업문 ...
} while(조건식);
2

```
var i=0;  
do { // i가 0에서 9까지 출력  
    document.write(i);  
    i++;  
} while(i<10);
```

0123456789

03. 조건문과 반복문

2. 반복문

기본 예제 9-11 / for 반복문을 사용한 0부터 100까지 합 계산

코드 9-17 loop_forSum.html

```
<script>
  // 변수를 선언합니다.
  let output = 0;

  // 반복을 수행합니다.
  for (let i = 0; i <= 100; i++) {
    output += i;
  }

  // 출력합니다.
  alert(output);
</script>
```

100까지 더해야 하므로
< = 연산자를 사용.

이 페이지 내용:

5050

확인

03. 조건문과 반복문

- [실습] for 반복문을 사용한 홀수합, 짝수합, 총합
 - 0부터 100까지 홀수합, 0부터 100까지 짝수합, 0부터 100까지 총합

5050 2550 2500

5050
2550
2500

- 자바스크립트로 HTML 콘텐츠를 웹 페이지에 직접 삽입
- 바로 브라우저 윈도우에 출력
- document.write()
 - 예) document.write("<h3>Welcome!</h3>");
- document.writeln()
 - writeln()은 텍스트에 '\n' 을 덧붙여 출력
 - '\n'을 덧붙이는 것은 고작해야 빈칸 하나 출력
 - 다음 줄로 넘어가는 것은 아님

03. 조건문과 반복문

- [실습] for 문으로 10px~35px 크기로 출력

```
<!DOCTYPE html>
<html>
<head>
<title>for 문</title>
</head>
<body>
<h3>for 문으로 10px~35px 크기 출력</h3>
<hr>
<script>
  for(var size=10; size<=35; size+=5) { // 5씩 증가
    document.write("<span ");
    document.write("style='font-size:" + size + "px'>");
    document.write(size + "px");
    document.write("</span>");
  }
</script>
</body>
</html>
```

for 문으로 10px~35px 크기 출력

10px15px20px25px30px35px

Section 04

함수

04. 함수

1. 선언과 호출, 실행 우선순위

- 선언과 호출
 - 함수 - 코드 집합을 나타내는 자료형
 - **익명 함수** 생성 - 함수 이름을 입력하지 않고 만들기
 - **선언적 함수** 생성 - 함수 이름을 입력해서 만들기

표 9-11 함수 생성 방법

방법	표현
익명 함수	<code>function () { }</code>
선언적 함수	<code>function 함수() { }</code>

익명 함수(무조건 한번 실행하는 자바스크립트 함수)

- 보통의 함수(선언적 함수) 경우 'function' 다음 함수 이름이 있으나, 익명 함수는 함수의 이름이 없음.
- 익명 함수는 단 한 번만 실행됨

04. 함수

1. 선언과 호출, 실행 우선순위

기본 예제 9-12

함수 선언

■ 1 익명 함수 선언하기

코드 9-18 function_noname.html

```
<script>
// 함수를 선언합니다.
let 함수 = function () {
    alert('함수_01');
    alert('함수_02');
};

// 출력합니다.
alert(typeof (함수) + ' : ' + 함수);
</script>
```

익명함수 구조: 변수에 함수의 코드를 저장하는 대신 함수명을 사용하지 않습니다. 대신 변수명을 마치 함수명처럼 사용해서 함수를 호출하거나 변수값을 이동시키는데 사용합니다.

- 변수값을 변수에 넣은 형태이므로 **함수 코드 끝에 세미콜론을 사용**

```
<script>
var 변수명 = function( 매개변수 )
{
    실행문;
};
</script>
```

이 페이지 내용:

```
function : function () {
    alert('함수_01');
    alert('함수_02');
}
```

확인

04. 함수

1. 선언과 호출, 실행 우선순위

기본 예제 9-12 / 함수 선언

- 2 선언적 함수 선언하기

코드 9-19 function_name.html

```
<script>
  // 함수를 선언합니다.
  function 함수() {
    alert('함수_01');
    alert('함수_02');
  };

  // 출력합니다.
  alert(typeof (함수) + ' : ' + 함수);
</script>
```

이 페이지 내용:

```
function : function 함수() {
  alert('함수_01');
  alert('함수_02');
}
```

확인

04. 함수

1. 함수선언

- 3 익명 함수/선언적 함수 선언하기 - 예제

```
<script>
//익명 함수 선언 및 변수에 대입
var hello = function()
{
    document.write("Hello World!");
};

//익명 함수 변수명으로 호출
hello();
</script>
```

Hello World!

```
<script>
//익명 함수 선언 및 변수에 대입
var hi = function()
{
    var string = "Hello World!";
    return string;
};

//익명 함수의 변수를 출력
document.write( hi() );
</script>
```

Hello World!

```
<script>
//익명 함수 선언 및 매개변수 사용
var fusion = function( a, b )
{
    var z = a + b;
    return z;
};

//익명 함수의 변수를 출력 및 인수 입력
document.write( fusion( "Hello", "World!" ) );
</script>
```

Hello World!

```
<script>
//익명 함수 호출
orange();

//익명 함수 선언
var orange = function()
{
    document.write("This is an orange.");
};
</script>
```

오류

```
<script>
//일반 함수 호출
Good();

//일반 함수 선언
function Good()
{
    document.write("Good Job!");
}
</script>
```

Good Job!

04. 함수

1. 선언과 호출, 실행 우선순위

```
// 함수를 호출합니다.  
함수();
```

- 실행 우선순위
 - 가장 마지막에 입력된 값이 저장

코드 9-20 function_priorityBetweenNoname.html

```
<script>  
  // 함수를 선언합니다.  
  함수 = function () { alert('함수_A'); };  
  함수 = function () { alert('함수_B'); };  
  
  // 함수를 호출합니다.  
  함수();  
</script>
```

이 페이지 내용:

함수_B

확인

04. 함수

1. 선언과 호출, 실행 우선순위

- 실행 우선순위
 - ※주의 ※ 선언적 함수와 익명 함수를 함께 사용할 때
 - 모든 코드를 읽기 전에 선언적 함수를 먼저 읽음

코드 9-21 function_priority.html

```
<script>
  // 함수를 선언합니다.
  함수 = function () { alert('함수_A'); };
  function 함수(){ alert('함수_B'); };

  // 함수를 호출합니다.
  함수();
</script>
```

이 페이지 내용:

함수_A

확인

04. 함수

2. 매개변수와 반환 값

- 매개변수: 함수의 괄호 안에 집어넣어 함수 쪽에 추가적인 정보를 전달.

```
// 함수를 호출합니다.  
alert('매개변수');
```

- 리턴 값: 함수를 실행한 결과를 반환 값

```
let minutes = date.getMinutes();  
let seconds = date.getSeconds();
```

```
function 함수 이름(매개변수, 매개변수, 매개변수) {  
    // 함수 코드  
    // 함수 코드  
    // 함수 코드  
    return 반환 값;  
}
```

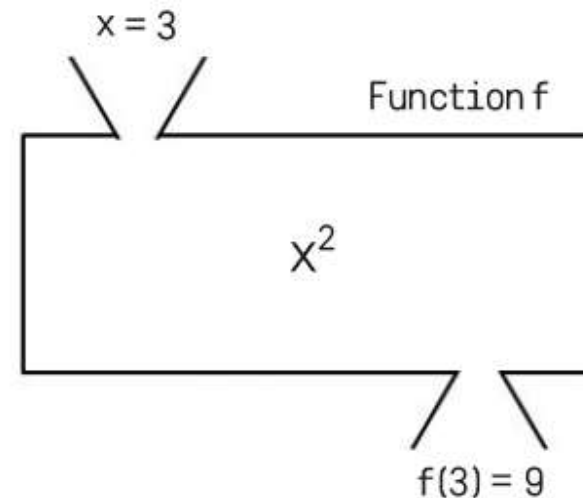


그림 9-16 함수

04. 함수

2. 매개변수와 반환 값

기본 예제 9-13

매개변수와 반환 값이 있는 함수

코드 9-22 function_return.html

```
<script>
  // 함수를 선언합니다.
  function f(x) {
    return x * x;
  }

  // 함수를 호출합니다.
  alert(f(3));
</script>
```

이 페이지 내용:

9

확인

04. 함수

3. 콜백 함수

- 매개변수로 전달되는 함수

기본 예제 9-14

콜백 함수

코드 9-23

function_callback.html

```
<script>
  // 함수를 선언합니다.
  function callTenTimes(callback) {
    // 10회 반복합니다.
    for (let i = 0; i < 10; i++) {
      callback(); // 매개변수로 전달된 함수를 호출합니다.
    }
  }

  // 변수를 선언합니다.
  let fun = function () {
    alert('함수 호출');
  };
  // 함수를 호출합니다.
  callTenTimes(fun);
</script>
```

열 번 출력됨.

매개변수 callback으로 fun을 전달.

이 페이지 내용:
함수 호출

확인

- 파라미터로 함수를 전달받아, 함수의 내부에서 실행하는 함수

04. 함수

3. 콜백 함수

- 익명함수로 실행하기

기본 예제 9-14

콜백 함수

코드 9-24 function_nonameCallback.html

```
<script>
  // 함수를 선언합니다.
  function callTenTimes(callback) {
    for (let i = 0; i < 10; i++) {
      callback();
    }
  }

  // 함수를 호출합니다.
  callTenTimes(function () {
    alert('함수 호출');
  });
</script>
```

Section 05

객체

05. 객체

1. 객체 개요

- 객체는 자료형 여러 개를 한 번에 저장
- 배열은 요소에 접근할 때 인덱스를 사용하지만, 객체는 키를 사용

```
<script>  
  // 배열을 선언합니다.  
  let array = ['사과', '바나나', '망고', '딸기'];  
</script>
```

(a) 배열 선언

```
array[0] → '사과'  
array[2] → '망고'
```

(b) 배열 요소 접근

그림 9-17 배열 선언과 요소 접근 예

인덱스	요소
0	사과
1	바나나
2	망고
3	딸기

05. 객체

1. 객체 개요

- 객체는 요소에 접근할 때 키를 사용.

코드 9-25 object_create.html

```
<script>
  // 객체를 선언합니다.
  let product = {
    제품명: '7D 건조 망고',
    유형: '당절임',
    성분: '망고, 설탕, 메타중아황산나트륨, 치자황색소',
    원산지: '필리핀'
  };
</script>
```

키	속성
제품명	7D 건조 망고
유형	당절임
성분	망고, 설탕, 메타중아황산나트륨, 치자황색소
원산지	필리핀

그림 9-18 객체 생성 예

05. 객체

1. 객체 개요

- 객체 뒤에 대괄호를 사용해 키를 입력하면 객체 속성에 접근.
- 객체 뒤에 .을 입력해 객체 속성에 접근.

```
product['제품명'] → '7D 건조 망고'  
product['유형'] → '당절임'  
product['성분'] → '망고, 설탕, 메타중아황산나트륨, 치자황색소'  
product['원산지'] → '필리핀'
```

```
product.제품명 → '7D 건조 망고'  
product.유형 → '당절임'  
product.성분 → '망고, 설탕, 메타중아황산나트륨, 치자황색소'  
product.원산지 → '필리핀'
```

05. 객체

1. 객체 개요

NOTE 식별자로 사용할 수 없는 키

코드 9-26 note_objectWithoutIdentifier.html

```
<script>
  // 객체를 선언합니다.
  let object = {
    'with space': 273,
    'with ~!@#$%^&*()_+': 52
  };

  // 출력합니다.
  alert(object['with space']);
  alert(object['with ~!@#$%^&*()_+']);
</script>
```

05. 객체

1. 객체 개요

- 다음과 같은 형식으로 **for in** 반복문을 작성해 객체를 순환.

```
for (let 키 in 객체) {  
    문장  
}
```

코드 9-27 object_withForIn.html

```
<script>  
    // 객체를 선언합니다.  
    let product = {  
        제품명: '7D 건조 망고',  
        유형: '당절임',  
        성분: '망고, 설탕, 메타중아황산나트륨, 치자황색소',  
        원산지: '필리핀'  
    };  
  
    // 출력합니다.  
    for (let i in product) {  
        alert(i + ':' + product[i]);  
    }  
</script>
```

05. 객체

2. 속성과 메서드

- 요소(element)
 - 배열에 있는 값 하나하나
- 속성(property)
 - 객체에 있는 값 하나하나

```
// 객체를 선언합니다.  
let object = {  
  number: 273,  
  string: 'rintiantta',  
  boolean: true,  
  array: [52, 273, 103, 32],  
  method: function () {  
  }  
};
```

05. 객체

2. 속성과 메서드

- 메서드: 객체 속성 중 자료형이 함수인 속성

```
// 객체를 선언합니다.  
let person = {  
  name: '윤인성',  
  eat: function (food) {  
    alert(food + '을/를 먹습니다.');  }  
};  
  
// 메서드를 호출합니다.  
person.eat('밥');
```

05. 객체

2. 속성과 메서드

- this 키워드
 - 객체에 있는 속성을 메서드에서 사용하고 싶을 때는
 - 자신이 가진 속성임을 분명하게 표시해야 함.

코드 9-28 object_this.html

```
<script>
  // 객체를 선언합니다.
  let person = {
    name: '윤인성',
    eat: function (food) {
      alert(this.name + '이 ' + food + '을/를 먹습니다.');
```

이 페이지 내용:
윤인성이 밥을/를 먹습니다.

확인