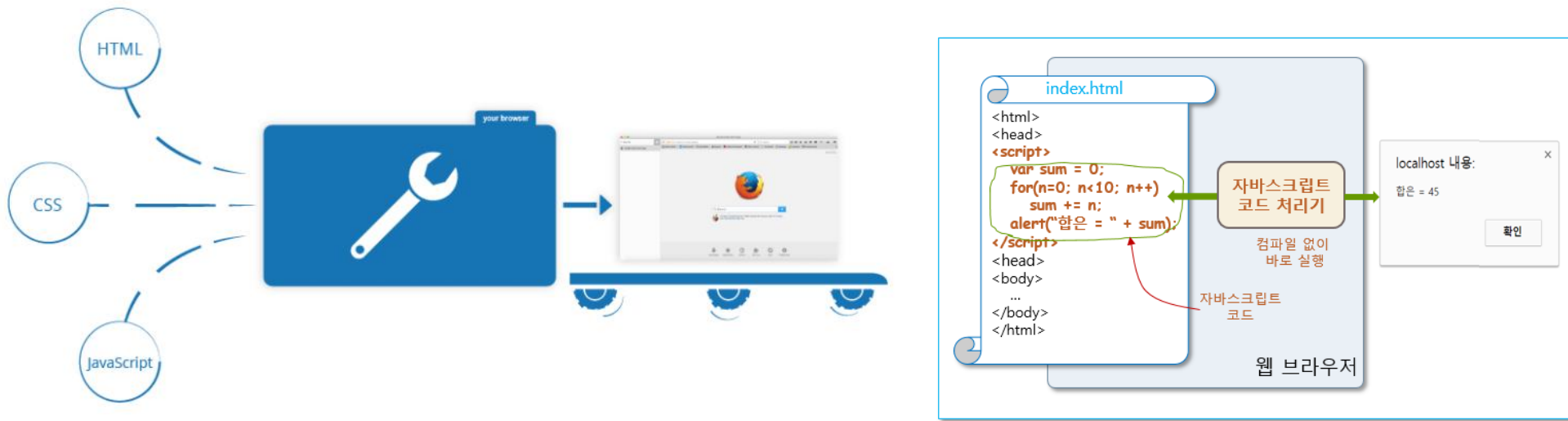


자바스크립트: 기본

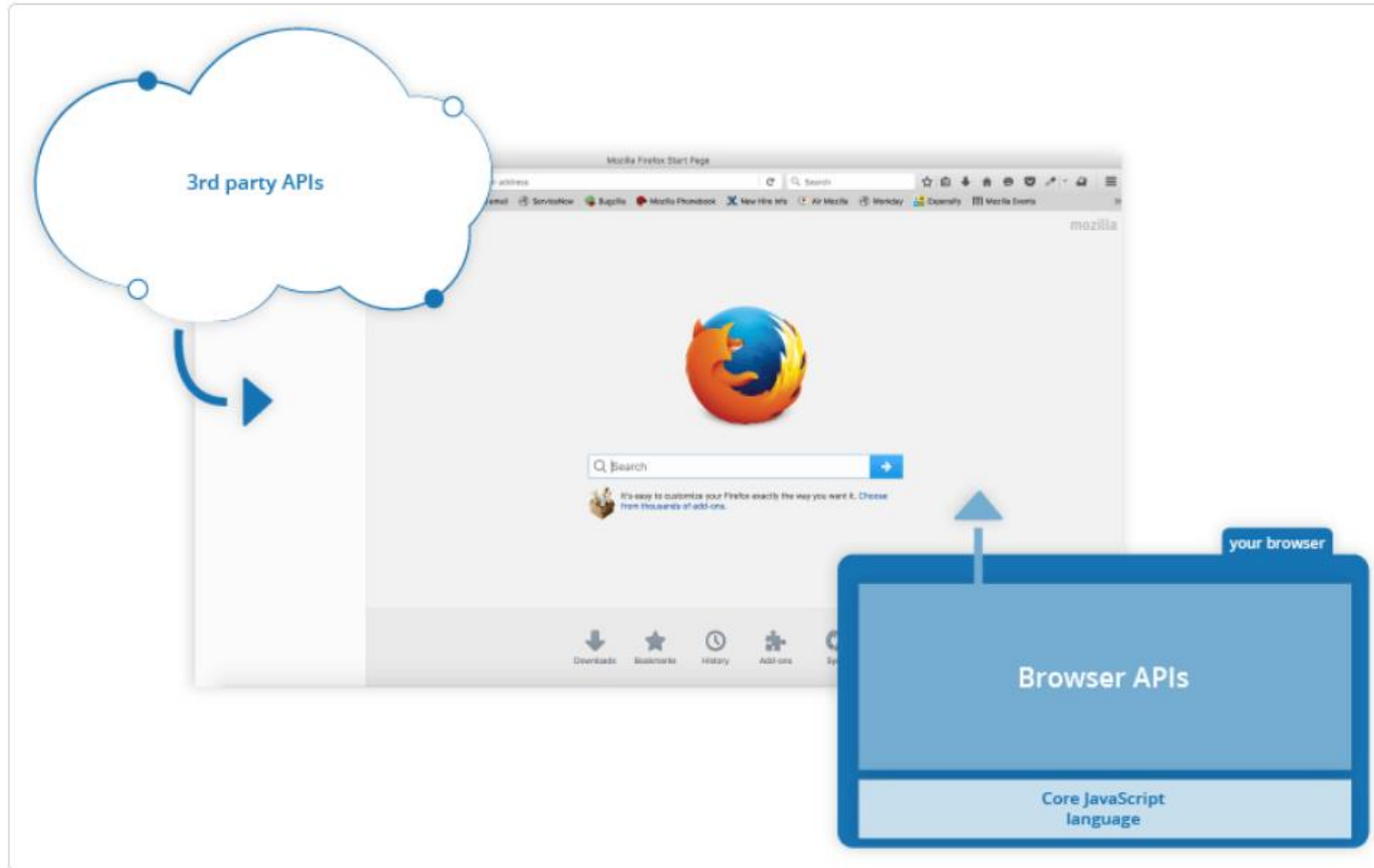
Javascript

- ◆ What happens when you load a web page in a browser:...



- ◆ A very common use of JavaScript is **to dynamically modify HTML and CSS to update a user interface**, via the **DOM API**
- ◆ **ECMAScript** : The standard for JavaScript

APIs in client-side JavaScript



자바스크립트

◆ 자바스크립트 개요

- 자바스크립트 작성 방법, 위치

◆ 데이터 타입과 변수

- 데이터 타입, 변수명 규칙, 변수 사용, 전역 변수와 지역 변수

◆ 연산자

- 문자열 연산자, 산술/비교/논리/조건/대입 연산자

◆ 제어문

- 조건문, 반복문

웹페이지에서 자바스크립트

◆ 자바스크립트의 역할

- ① 요소의 추가 및 삭제
- ② CSS 및 HTML 요소의 스타일 변경 : 내용 및 모양 동적 제어
- ③ 사용자와의 상호작용 : 사용자 입력, 계산
- ④ 폼의 유효성 검증
- ⑤ 마우스와 키보드 이벤트에 대한 스크립트 실행
- ⑥ 웹 브라우저 제어 및 쿠키 등의 설정과 조회
- ⑦ 웹 서버와의 통신

자바스크립트 작성 규칙

- ◆ 대소문자 구분
- ◆ 문장은 세미콜론(;)으로 구분

바른 예	<pre>var age=25 document.write("당신의 나이는 " + age + "입니다.")</pre>
	<pre>var age=25; document.write("당신의 나이는 " + age + "입니다.");</pre>
	<pre>var age=25; document.write("당신의 나이는 " + age + "입니다.");</pre>
잘못된 예	<pre>var age=25 document.write("당신의 나이는 " + age + "입니다.")</pre>

- ◆ 큰따옴표(“ ”)와 작은따옴표(‘ ’)를 구분

바른 예	<pre>document.write("<div style='color: red;'> 자바스크립트 학습 </div>");</pre>
	<pre>document.write("<div style='color: red;'> 자바스크립트 학습 </div>");</pre>
잘못된 예	<pre>document.write("<div style='color: red;'> 자바스크립트 학습 </div>")</pre>

자바스크립트 코드 위치

- ◆ HTML 문서 내부
- ◆ HTML 문서 외부
- ◆ 혼합

자바스크립트 위치 : 문서 내부

```
<head>
  <meta charset="utf-8"/>
  <title>자바스크립트 예제</title>
  <script>
    // 자바스크립트 코드 작성
  </script>
</head>
<body>
  <script>
    // 자바스크립트 코드 작성
  </script>
</body>
```

```
<button type="button" onclick="alert('자바스크립트')">버튼 클릭</button>
```


예제 9-1 자바스크립트 코드의 실행 순서 살펴보기

```
<head>
  <meta charset="utf-8"/>
  <title>자바스크립트 예제</title>
  <script>
    var num=0;
    document.write("head 태그 내 실행 순서 : " + num + "<br/>");
  </script>
  <script>
    var num=1;
    document.write("head 태그 내 실행 순서 : " + num + "<br/>");
  </script>
</head>
<body>
  <script>
    var num=2;
    document.write("body 태그 내 실행 순서 : " + num + "<br/>");
  </script>
  <script>
    var num = 3;
    document.write("body 태그 내 실행 순서 : " + num + "<br/>");
  </script>
</body>
```

자바스크립트 위치 : 문서 외부

◆ 별도의 자바스크립트 파일 작성

- HTML 문서의 <script> 태그에 src 속성 추가

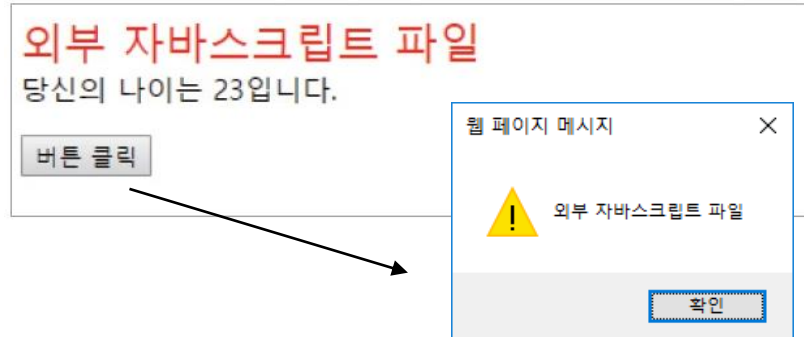
표 9-3 자바스크립트 파일 위치에 따른 src 속성값

위치	src 속성값
HTML 문서와 같은 디렉터리에 있는 경우	<script src="myscript.js"> </script>
HTML 문서와 다른 디렉터리에 있는 경우	<script src="../ejs/myscript.js"> </script>
HTML 문서와 다른 서버 디렉터리에 있는 경우	<script src="http://www.hanbit.co.kr/jsfile/myscript.js"> </script>

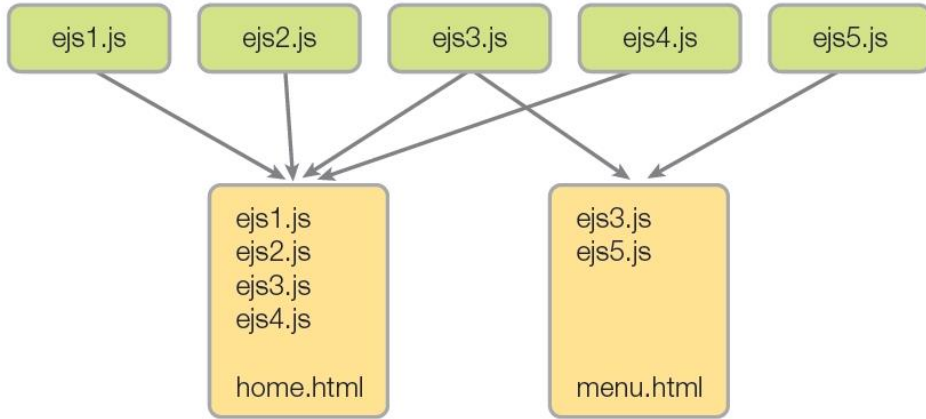
예제 9-2 외부 자바스크립트 문서 작성 후 참조하기

```
var age=23;
/* 문자에 스타일 속성 적용 */
document.write("<div style='color: red; font-size: 24px;'>외부 자바스크립트 파일</div>");
document.write("당신의 나이는 " + age + "입니다.");
```

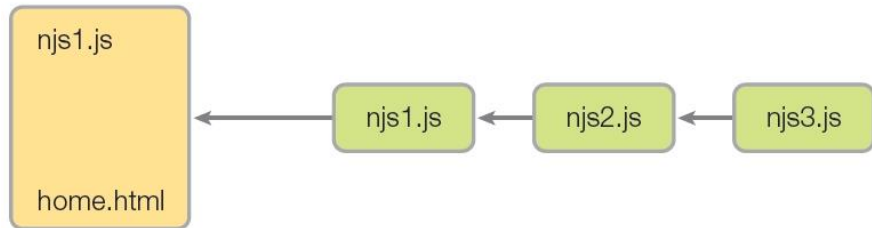
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8"/>
  <script src="./ejs/ejs.js"> </script>
</head>
<body>
  <p>
    <!-- 버튼을 클릭하면 메시지 창 출력 -->
    <button type="button" onclick="alert('외부 자바스크립트 파일')">버튼 클릭</button>
  </p>
</body>
</html>
```



자바스크립트 위치 : 하나 이상의 문서 외부



(a) 기능 단위로 분리된 여러 개의 자바스크립트 파일 참조



(b) 내포 관계를 가진 자바스크립트 파일 참조

예제 9-3 여러 개의 외부 자바스크립트 파일 참조하기

```
document.write("ejs1.js");  
document.write("<div style='color: red; font-size: 24px;'>외부 자바스크립트 파일</div>");
```

```
document.write("ejs2.js");  
document.write("<div style='color: blue; font-size: 20px;'>외부 자바스크립트 파일</div>");
```

```
document.write("ejs3.js");  
document.write("<div style='color: green; font-size: 16px;'>외부 자바스크립트 파일</div>");
```

```
<!DOCTYPE html>  
<html>  
<head>  
  <meta charset="utf-8"/>  
  <script src="./ejs/ejs1.js"> </script>  
</head>  
<body>  
  <script src="./ejs/ejs2.js"> </script>  
  <script src="./ejs/ejs3.js"> </script>  
</body>  
</html>
```

```
ejs1.js  
외부 자바스크립트 파일  
ejs2.js  
외부 자바스크립트 파일  
ejs3.js  
외부 자바스크립트 파일
```

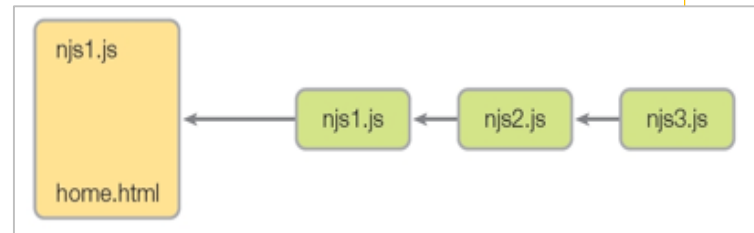
예제 9-4 내포 관계인 자바스크립트 파일 참조하기

```
document.write("njs1.js");  
document.write("<div style='color: red; font-size: 24px;'>외부 자바스크립트 파일</div>");  
document.write("<script src='./ejs/njs2.js'> </script>");
```

```
document.write("njs2.js는 njs1.js에 포함");  
document.write("<div style='color: blue; font-size: 20px;'>외부 자바스크립트 파일</div>");  
document.write("<script src='./ejs/njs3.js'> </script>");
```

```
document.write("njs3.js는 njs2.js에 포함");  
document.write("<div style='color: green; font-size: 16px;'>외부 자바스크립트 파일</div>");  
alert('Nested Script File');
```

```
<!DOCTYPE html>  
<html>  
<head>  
  <meta charset="utf-8"/>  
</head>  
<body>  
  <script src="./ejs/njs1.js"> </script>  
</body>  
</html>
```



njs1.js
외부 자바스크립트 파일
njs2.js는 njs1.js에 포함
외부 자바스크립트 파일
njs3.js는 njs2.js에 포함
외부 자바스크립트 파일

예제 9-5 혼합 방법으로 자바스크립트 파일 포함하기

```
document.write("mjs1.js");  
document.write("<div style='color: red; font-size: 24px;'>외부 자바스크립트 파일</div>");
```

```
<!DOCTYPE html>  
<html>  
<head>  
  <meta charset="utf-8"/>  
  <script src="./ejs/mjs.js"> </script>  
</head>  
<body>  
  <script>  
    document.write("<div style='color: blue; font-size: 20px;'>내부 자바스크립트</div>");  
  </script>  
</body>  
</html>
```

mjs1.js

외부 자바스크립트 파일

내부 자바스크립트

자바스크립트

- ◆ 자바스크립트 개요

- 자바스크립트 작성 방법, 위치

- ◆ 데이터 타입과 변수

- 데이터 타입, 변수명 규칙, 변수 사용, 전역 변수와 지역 변수

- ◆ 연산자

- 문자열 연산자, 산술/비교/논리/조건/대입 연산자

- ◆ 제어문

- 조건문, 반복문

데이터 타입

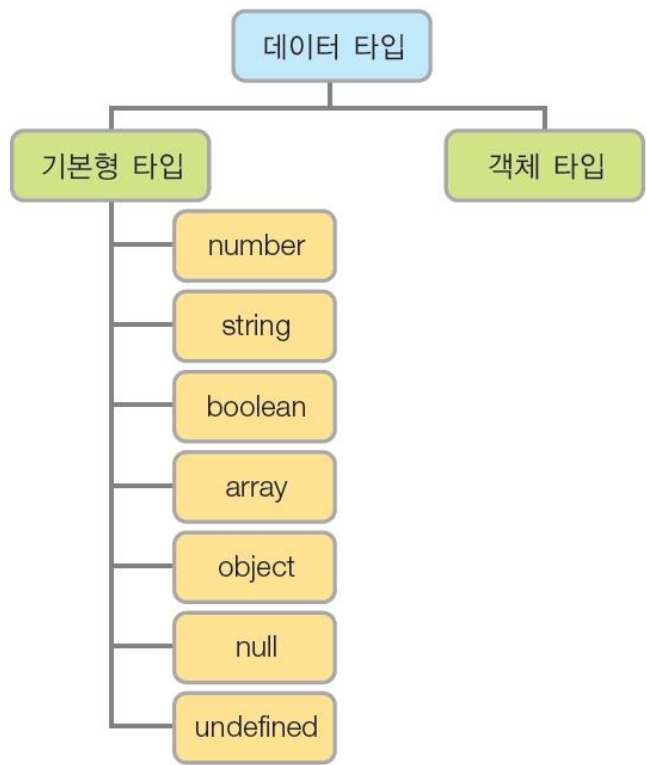


그림 9-4 자바스크립트의 데이터 타입

표 9-4 기본형 타입의 종류

종류	설명	사용 예
number	정수 혹은 실수	100, 10.5, 10e+3
string	문자 혹은 문자열	"홍길동", '홍길동'
boolean	참 혹은 거짓	true, false
array	데이터의 집합(배열, 객체로 취급)	["서울", "부산", "인천"]
object	데이터 속성과 값으로 이루어진 집합	{name: '홍길동', age: 25}
null	객체 값이 없음	null
undefined	데이터 값이 정해지지 않음	undefined

- ♦ JavaScript is a ***loosely typed*** and ***dynamic*** language.
- ♦ Variables are not directly associated with any particular value type, and any variable can be assigned (and re-assigned) values of all types:

```
let foo = 42;    // foo is now a number  
foo      = 'bar'; // foo is now a string  
foo      = true;  // foo is now a boolean
```

예제 9-6 typeof 연산자를 사용하여 데이터 타입 확인하기

```
<head>
  <meta charset="utf-8"/>
</head>
<body>
  <script>
    var num;      // 변수 값이 없음
    var obj=null;  // 객체 변수 값이 없음
    document.write( typeof 100+"<br>" );
    document.write( typeof 10.5+"<br>" );
    document.write( typeof "홍길동"+"<br>" );
    document.write( typeof true+"<br>" );
    document.write( typeof [1,2,3]+"<br>" );
    document.write( typeof {name: '홍길동', age:25}+"<br>" );
    document.write( typeof num+"<br>" );
    document.write( typeof obj+"<br>" );
  </script>
</body>
```

식별자

◆ 작성규칙

- 문자, 밑줄(_), 달러 기호(\$)로 시작
- 대소문자 구별, 영문자 사용 권장
- 예약어는 사용 불가능

표 9-5 자바스크립트 예약어

abstract	Arguments	boolean	break	byte
case	catch	char	class	const
continue	debugger	default	delete	do
double	else	enum	eval	export
extends	false	final	finally	float
for	function	goto	if	implements
import	in	instanceof	int	interface
let	long	native	new	null
package	private	protected	public	return
short	static	super	switch	synchronized
this	throw	throws	transient	true
try	typeof	var	void	volatile
while	with	yield		

예제 9-7 변수의 데이터 타입 변경하기

```
<head>
  <meta charset="utf-8"/>
</head>
<body>
  <script>
    var num=10;
    document.write("num 변수 : " + typeof num + " 타입<p/>");
    document.write("--- 데이터 값 변경 후 ---<p/>");
    var num="홍길동";
    document.write("num 변수 : " + typeof num + " 타입<p/>");
  </script>
</body>
```

예제 9-8 변수명 재선언 시 데이터 값 변화 살펴보기

```
<head>
  <meta charset="utf-8" />
</head>
<body>
  <script>
    stdName="홍길동";    // var 키워드 생략
    comGrade=96;        // var 키워드 생략
    var stdName;         // 변수명 재선언
    var comGrade;        // 변수명 재선언
    document.write("학생 이름 : " + stdName + "<br>");
    document.write("컴퓨터 점수 : " + comGrade + "<br>");
  </script>
</body>
```

변수의 메모리 수명

◆ Memory life cycle

- JavaScript will automatically allocate memory **when values are initially declared**
- Release when the memory is not needed anymore (***garbage collection***)

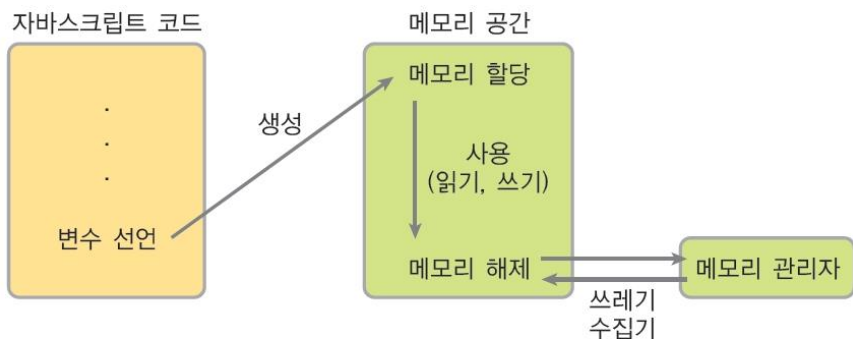


그림 9-5 변수의 메모리 수명

```
var n = 123; // allocates memory for a number
var s = 'azerty'; // allocates memory for a string

var o = {
  a: 1,
  b: null
}; // allocates memory for an object and contained values
```

전역 변수와 지역 변수

◆ 키워드 var

- Declares a **function-scoped** or **globally-scoped** variable

```
<script>
  var globValue1;    // 전역 변수 선언
  globValue2;        // 전역 변수 선언, var 생략

  function 함수() {
    var locValue;    // 지역 변수 선언
    globValue;       // 함수 내부에서 var 생략 시 자동 전역 변수로 선언
    locValue=10;     // 지역 변수 사용
  }

  globValue=10;      // 전역 변수 사용
</script>
```

그림 9-6 전역 변수와 지역 변수 선언

◆ 키워드 let

- Declares a **block-scoped local variable**

```
function varTest() {  
  var x = 1;  
  {  
    var x = 2; // same variable!  
    console.log(x); // 2  
  }  
  console.log(x); // 2  
}  
  
function letTest() {  
  let x = 1;  
  {  
    let x = 2; // different variable  
    console.log(x); // 2  
  }  
  console.log(x); // 1  
}
```

예제 9-10 전역 변수와 지역 변수 이해

```
<script>
    function getGrade() {      // 함수 정의
        kor=95;                // 자동 전역 변수
    }
    var kor=100;               // 전역 변수
    getGrade();                // 함수 호출
    document.write("국어 점수 : " + kor + "<br>");
</script>
```

국어 점수 : 95

예제 9-11 전역 변수와 지역 변수 이해

```
<script>
    function getGrade() {
        var kor=95;
    }
    getGrade();
    document.write("지역 변수 값은 함수 외부에서 사용할 수 없습니다.<br>");
    document.write("국어 점수 : " + kor + "<br>");
</script>
```

지역 변수 값은 함수 외부에서 사용할 수 없습니다.

예제 9-12 전역 변수와 지역 변수 이해

```
<script>
    function getGrade() {
        var kor=95;
        return kor;
    }
    getKor=getGrade();
    document.write("국어 점수 : " + getKor + "<br>");
</script>
```

자바스크립트

◆ 자바스크립트 개요

- 자바스크립트 작성 방법, 위치

◆ 데이터 타입과 변수

- 데이터 타입, 변수명 규칙, 변수 사용, 전역 변수와 지역 변수

◆ 연산자

- 문자열 연산자, 산술/비교/논리/조건/대입 연산자

◆ 제어문

- 조건문, 반복문

연산자

◆ 연산자의 종류

연산자	기호
문자열 연산자	+(문자열 연결)
산술 연산자	++(증가 연산), --(감소 연산), *(곱셈), /(나눗셈), %(나머지), +(덧셈), -(뺄셈)
비교 연산자	<(작다), <=(작거나 같다), >(크다), >=(크거나 같다), ==(값이 같다), !=(값이 다르다), ===(값과 타입 모두 같다), !==(값 또는 타입이 다르다)
논리 연산자	&(비트 AND), (비트 OR), ^(비트 XOR), &&(논리 AND), (논리 OR)
조건 연산자	(판단) ? true : false;
대입 연산자	=, +=, -=, *=, /=, %=, <<=, >>=, >>>=, &=, =, ^=

◆ Destructuring assignment :

- Makes it possible to extract data from arrays or objects

```
var foo = ['one', 'two', 'three'];  
  
// without destructuring  
var one   = foo[0];  
var two   = foo[1];  
var three = foo[2];  
  
// with destructuring  
var [one, two, three] = foo;
```

◆ Logical operators : &&, ||

- Typically used with Boolean (logical) values
- If these operators are used with **non-Boolean values**, they may return a non-Boolean value.

```
var a1 = true && true;    // t && t returns true
var a2 = true && false;   // t && f returns false
var a3 = false && true;   // f && t returns false
var a4 = false && (3 == 4); // f && f returns false
var a5 = 'Cat' && 'Dog';  // t && t returns Dog
var a6 = false && 'Cat';  // f && t returns false
var a7 = 'Cat' && false;  // t && f returns false
```

```
var o1 = true || true;   // t || t returns true
var o2 = false || true;  // f || t returns true
var o3 = true || false;  // t || f returns true
var o4 = false || (3 == 4); // f || f returns false
var o5 = 'Cat' || 'Dog'; // t || t returns Cat
var o6 = false || 'Cat';  // f || t returns Cat
var o7 = 'Cat' || false;  // t || f returns Cat
```


Complete list of JavaScript falsy values

<code>false</code>	The keyword false .
<code>0</code>	The Number zero (so, also <code>0.0</code> , etc., and <code>0x0</code>).
<code>-0</code>	The Number negative zero (so, also <code>-0.0</code> , etc., and <code>-0x0</code>).
<code>0n</code> , <code>-0n</code>	The BigInt zero and negative zero (so, also <code>0x0n</code> / <code>-0x0n</code>).
<code>""</code> , <code>''</code> , <code>``</code>	Empty string value.
null	null — the absence of any value.
undefined	undefined — the primitive value.
NaN	NaN — not a number.

자바스크립트

◆ 자바스크립트 개요

- 자바스크립트 작성 방법, 위치

◆ 데이터 타입과 변수

- 데이터 타입, 변수명 규칙, 변수 사용, 전역 변수와 지역 변수

◆ 연산자

- 문자열 연산자, 산술/비교/논리/조건/대입 연산자

◆ 제어문

- 조건문, 반복문

자바스크립트 제어문

◆ 제어문 종류

유형	설명	구조
조건문	조건에 따라 다음 문장을 선택적으로 실행한다.	<ul style="list-style-type: none">• If문• if~else문• 다중 if~else문• switch~case문
반복문	동일한 명령을 여러 번 처리하거나 특정 연산을 반복적으로 처리한다.	<ul style="list-style-type: none">• for문• while문• do~while문
보조 제어문	조건문을 만나면 건너뛰거나 반복 수행을 종료한다. 반복문 내에서 사용한다.	<ul style="list-style-type: none">• continue문• break문

예제 9-20 로그인 프로그램 만들기

이 페이지 내용: ×

아이디 입력

확인

취소

이 페이지 내용: ×

비밀번호 입력

확인

취소

▲ 아이디/비밀번호 입력

회원 인증에 성공했습니다.

저자 홈페이지를 클릭하세요.

[차세대 웹 프로그래밍](#)

회원 인증에 실패했습니다.

웹 문서에 접근할 수 없습니다. 관리자에게 문의하시기 바랍니다.

관리자 e-mail : gosyhong@gmail.com

▲ 로그인 성공/실패

예제 9-20 로그인 프로그램 만들기

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8"/>
</head>
<body>
  <p>아이디, 비밀번호 입력</p>
  <script src="./ejs/script.js"> </script>
</body>
</html>
```

ch09/script.js

```
id=prompt('아이디 입력');
if(id=='admin') {
  password=prompt('비밀번호 입력');
  if(password=='123456') {
    location.href="20_login.html"
  }
  else {
    location.href="20_error.html"
  }
}
else {
  location.href="20_error.html"
}
```

The screenshot displays two side-by-side browser windows. The left window shows a login form with the title '이 페이지 내용:' (Page Content:). It contains a label '아이디 입력' (ID input) and a text box with 'admin'. Below it are '확인' (Confirm) and '취소' (Cancel) buttons. The right window shows a similar form with the title '이 페이지 내용:' (Page Content:). It contains a label '비밀번호 입력' (Password input) and a text box with '123456'. Below it are '확인' (Confirm) and '취소' (Cancel) buttons. Below the forms, there is a section titled '▲ 아이디/비밀번호 입력' (ID/Password input). It contains two boxes: the left one says '회원 인증에 성공했습니다.' (Member authentication successful) and the right one says '회원 인증에 실패했습니다.' (Member authentication failed). Below these boxes, there is a section titled '▲ 로그인 성공/실패' (Login success/failure). It contains two boxes: the left one says '저자 홈페이지를 클릭하세요.' (Click the author's homepage) and the right one says '웹 문서에 접근할 수 없습니다. 관리자에게 문의하시기 바랍니다.' (Cannot access the web document. Please contact the administrator).

이 페이지 내용:

아이디 입력

admin

확인 취소

이 페이지 내용:

비밀번호 입력

123456

확인 취소

▲ 아이디/비밀번호 입력

회원 인증에 성공했습니다.

저자 홈페이지를 클릭하세요.

[차세대 웹 프로그래밍](#)

회원 인증에 실패했습니다.

웹 문서에 접근할 수 없습니다. 관리자에게 문의하시기 바랍니다.

관리자 e-mail : gosyhong@gmail.com

▲ 로그인 성공/실패

예제 9-22 요일을 알려주는 프로그램

```
<script>
    var day;
    var week=new Date().getDay();    // 0(일요일)~6(토요일)
    switch(week) {
        case 0:
            day="일요일";           break;
        case 1:
            day="월요일";           break;
        case 2:
            day="화요일";           break;
        case 3:
            day="수요일";           break;
        case 4:
            day="목요일";           break;
        case 5:
            day="금요일";           break;
        case 6:
            day="토요일";           break;
        default:
            day="없는 요일";
    }
    document.write("오늘은 <b>" + day + "</b>입니다. <p/>");
</script>
```

예제 9-24 구구단 프로그램 만들기

```
<script>
    var x, y;
    for(x=2; x<=5; x++) {
        document.write("<b> ---[ " + x + "단]--- </b>" + "<br>");
        for(y=1; y <= 9; y++) {
            document.write(x + "*" + y + "=" + (x * y) + "<br>");
        }
    }
</script>
```

---[2단]---

2*1=2

2*2=4

2*3=6

2*4=8

2*5=10

2*6=12

2*7=14

예제 9-26 1부터 10000까지 합 구하기

```
<script>
  var x=1;
  var sum=0;
  while(1) {
    sum+=x;
    x++;
    if(x==10001)
      break;
  }
  document.write("1~10000까지 합 : <b>" + sum + "</b>");
</script>
```


예제 9-27 do~while문으로 1부터 100까지 합 구하기

```
<script>
  var x=1;
  var sum=0;
  do {
    sum+=x;
    x++;
  } while(x<=100);
  document.write("1~100까지 합 : <b>" + sum + "</b>");
</script>
```

자바스크립트 : 기본 문법

◆ 자바스크립트 개요

- 자바스크립트 작성 방법, 위치

◆ 데이터 타입과 변수

- 데이터 타입, 변수명 규칙, 변수 사용, 전역 변수와 지역 변수

◆ 연산자

- 문자열 연산자, 산술/비교/논리/조건/대입 연산자

◆ 제어문

- 조건문, 반복문