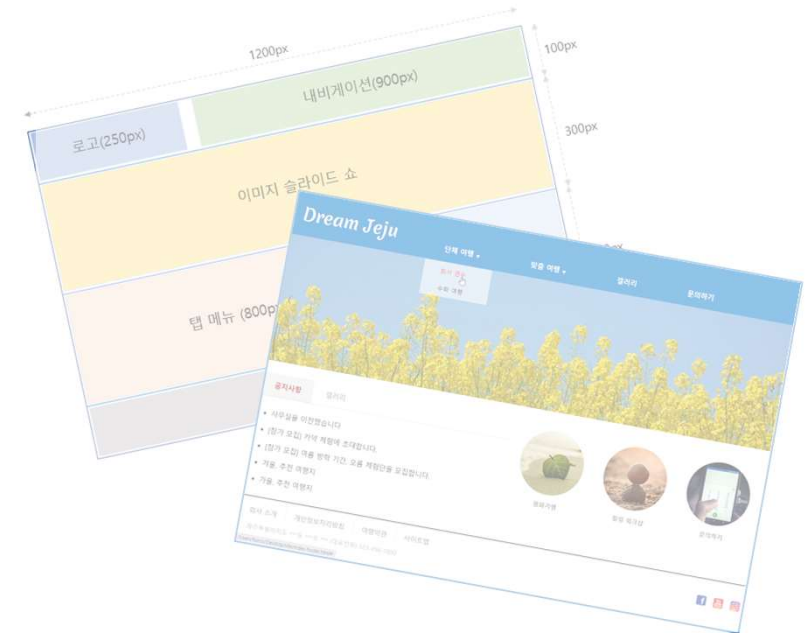


16. 자바스크립트와 객체

16-1 객체 알아보기

16-2 자바스크립트와 내장 객체

16-3 브라우저와 관련된 객체



객체 알아보기

객체(object)란

- 프로그램에서 인식할 수 있는 모든 대상
- 데이터를 저장하고 처리하는 기본 단위

자바스크립트 객체

자바스크립트 안에 미리 객체로 정의해 놓은 것

- 문서 객체 모델(DOM): 문서 뿐만 아니라 웹 문서 안에 포함된 이미지, 링크, 텍스트 필드 등을 모두 별도의 객체로 관리
- 브라우저 관련 객체: 웹 브라우저 정보를 객체로 관리
- 내장 객체: 웹 프로그래밍에서 자주 사용하는 요소를 객체로 정의해 놓음.

사용자 정의 객체

필요할 때마다 사용자가 직접 만드는 객체

객체의 인스턴스 만들기

- 객체는 객체 자체가 아니라 인스턴스 형태로 만들어서 사용
- 인스턴스: 객체를 틀처럼 사용해서 같은 모양으로 찍어낸 것.

기본형 new 객체명

예) Date 객체의 인스턴스를 만들어서 날짜와 시간 표시하기

```
var now = new Date(); // Date 인스턴스 만들고 변수에 저장하기
document.write("현재 시각은 " + now);
```

현재 시각은 Wed Nov 11 2020 15:58:24
GMT+0900 (대한민국 표준시)

객체 알아보기

프로퍼티(property)와 메서드(method)

- 프로퍼티 : 객체의 특징이나 속성
- 메서드 : 객체에서 할 수 있는 동작



프로퍼티	메서드
제조사	시동 걸기
모델명	움직이기
색상	멈추기
배기량	주차하기

그림 16-2 자동차의 프로퍼티와 메서드

- 2) now 변수에 저장 ← 1) Date 객체의 인스턴스를 만들어서

```
var now = new Date();  
document.write("현재 시각은 " + now.toLocaleString());
```

- 3) Date 객체의 메서드 사용

객체 정의

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>Object Methods</h2>
<p>A method is a function definition stored as a property value.</p>

<p id="demo"></p>

<script>
const person = {
  firstName: "John",
  lastName: "Doe",
  id: 5566,
  fullName: function() {
    return this.firstName + " " + this.lastName;
  }
};

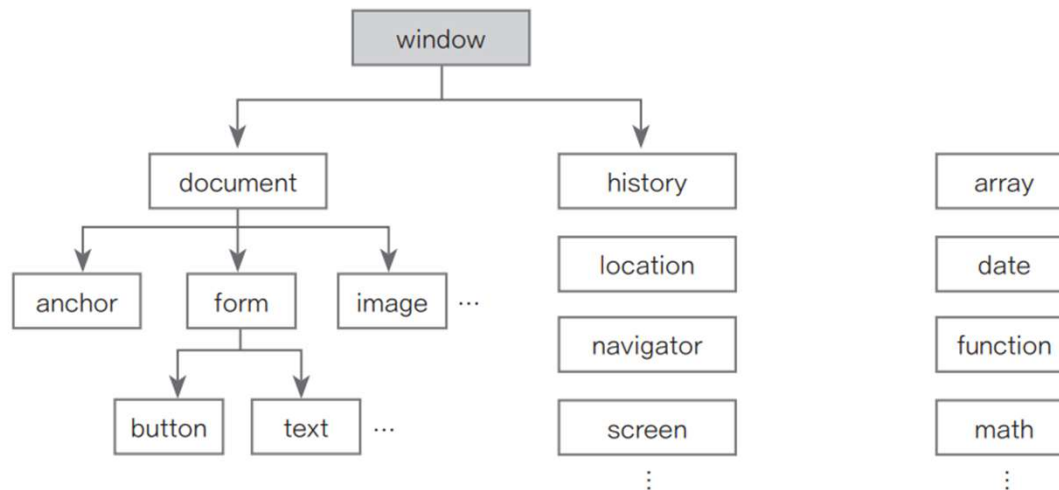
document.getElementById("demo").innerHTML = person.fullName();
</script>
</body>
</html>
```

```
<script>
// Create an Object
const person = new Object();
person.firstName = "John";
person.lastName = "Doe";
person.age = 50;
person.eyeColor = "blue";

// Display Object Content
document.getElementById("demo").innerHTML =
person.firstName + " is " + person.age + " years old.";
</script>
```

내장객체

- 사용자가 가져다 쓸 수 있도록 미리 만들어진 객체.
- 웹 브라우저에 웹 문서가 열리면 가장 먼저 window 객체가 만들어지고 하위에 웹 브라우저 요소에 해당하는 객체들이 만들어진다.
- window 객체를 비롯해 하위에 연결된 객체들은 모두 HTML 웹 API에 만들어진 객체들



window 객체

- Window 객체는 웹 브라우저 창의 상태를 제어하는 객체
- 자바스크립트 객체 중 최상위이자 기본이 되는 객체
- 자바스크립트의 모든 객체는 Window 객체에 포함된다
- Window 객체에는 웹 브라우저 창과 관련된 여러 가지 속성이 있다.
- Window 객체의 속성과 메서드에 접근하는 마침표(.)를 사용한다.

속성	설명
document	브라우저 창에 표시된 웹 문서에 접근할 수 있습니다.
frameElement	현재 창이 다른 요소 안에 포함되어 있으면 그 요소를 반환합니다. 포함되어 있지 않으면 null을 반환합니다.
innerHeight	내용 영역의 높이를 나타냅니다.
innerWidth	내용 영역의 너비를 나타냅니다.
localStorage	웹 브라우저에서 데이터를 저장하는 로컬 스토리지를 반환합니다.
location	Window 객체의 위치/현재 URL을 나타냅니다.
name	브라우저 창의 이름을 가져오거나 수정합니다.
outerHeight	브라우저 창의 바깥 높이를 나타냅니다.
outerWidth	브라우저 창의 바깥 너비를 나타냅니다.

pageXOffset	스크롤했을 때 화면이 수평으로 이동하는 픽셀 수. scrollX와 같습니다.
pageYOffset	스크롤했을 때 화면이 수직으로 이동하는 픽셀 수. scrollY와 같습니다.
parent	현재 창이나 서브 프레임의 부모 프레임입니다.
screenX	브라우저 창의 왼쪽 테두리가 모니터 왼쪽 테두리에서부터 떨어져 있는 거리를 나타냅니다.
screenY	브라우저 창의 위쪽 테두리가 모니터 위쪽 테두리에서부터 떨어져 있는 거리를 나타냅니다.
scrollX	스크롤했을 때 수평으로 이동하는 픽셀 수를 나타냅니다.
scrollY	스크롤했을 때 수직으로 이동하는 픽셀 수를 나타냅니다.
sessionStorage	웹 브라우저에서 데이터를 저장하는 세션 스토리지를 반환합니다.

window 객체의 주요 메서드(함수)

- 대화 창을 표시하거나 브라우저 창의 크기나 위치를 알아내고 지정하는 등
- 웹 브라우저 창 자체와 관련된 것이 대부분
- 앞에서 사용했던 alert() 함수나 prompt() 함수도 window 객체의 함수
- 객체의 함수 표기법에 따르면 window.alert()라고 입력해야 하지만, window.를 생략할 수도 있다.

함수	설명
alert()	알림 창(Alert Dialog)을 표시합니다.
blur()	창에서 포커스를 제거합니다.
close()	현재 열려 있는 창을 닫습니다.
confirm()	[확인], [취소]가 있는 확인 창을 표시합니다.
focus()	현재 창에 포커스를 부여합니다.
moveBy()	현재 창을 지정한 크기만큼 이동합니다.
moveTo()	현재 창을 지정한 좌표로 이동합니다.
open()	새로운 창을 엽니다.
postMessage()	다른 창으로 메시지를 전달합니다.
print()	현재 문서를 인쇄합니다.
prompt()	프롬프트 창에 입력한 텍스트를 반환합니다.

resizeBy()	지정한 크기만큼 현재 창 크기를 조절합니다.
resizeTo()	동적으로 브라우저 창의 크기를 조절합니다.
scroll()	문서에서 특정 위치로 스크롤합니다.
scrollBy()	지정한 크기만큼씩 스크롤합니다.
scrollTo()	지정한 위치까지 스크롤합니다.
setCursor()	현재 창의 커서를 변경합니다.
showModalDialog()	모달 창을 표시합니다.
sizeToContent()	내용에 맞게 창 크기를 맞춥니다.
stop()	로딩을 중지합니다.

팝업 창을 여는 open() 함수

```
window.open(URL, name, specs, replace)
```

```
name : _blank, _parent, _self, _top
```

```
<body>

<h1>The Window Object</h1>
<h2>The open() Method</h2>

<p>Click the button to open a new browser window.</p>

<button onclick="myFunction()">Try it</button>

<script>
function myFunction() {
  window.open("https://www.w3schools.com");
}
</script>

</body>
```



```
9    <p>Click the button to open a new browser window.</p>
10
11    <button onclick="myFunction()">Try it</button>
12    <button onclick="myClose()">창닫기</button>
13    <script>
14        let myWin;
15        function myFunction() {
16            myWin = window.open("https://www.w3schools.com");
17        }
18        function myClose(){
19            myWin.close();
20        }
21    </script>
```

screen 객체

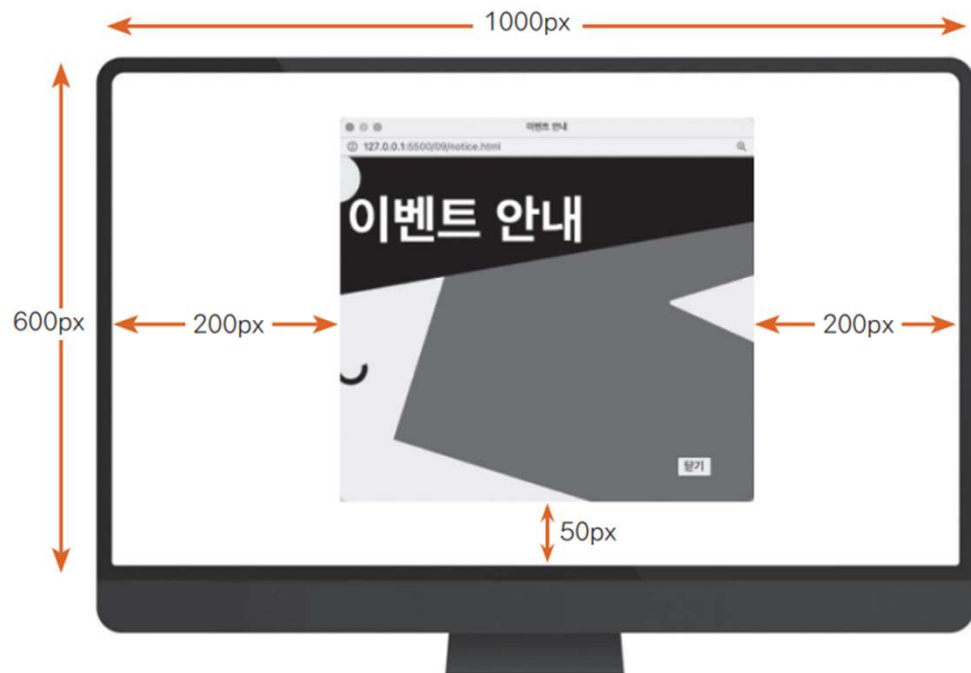
사용자의 화면 크기나 방향 등의 정보를 담고 있는 객체

구분		설명
프로퍼티	availHeight	UI 영역(예를 들어 윈도우의 작업 표시줄이나 Mac의 독)을 제외한 영역의 높이를 나타냅니다.
	availWidth	UI 영역을 제외한 내용 표시 영역의 너비를 나타냅니다.
	colorDepth	화면에서 픽셀을 렌더링할 때 사용하는 색상 수를 나타냅니다.
	height	UI 영역을 포함한 화면의 높이를 나타냅니다.
	orientation	화면의 현재 방향을 나타냅니다.
	pixelDepth	화면에서 픽셀을 렌더링할 때 사용하는 비트 수를 나타냅니다.
	width	UI 영역을 포함한 화면의 너비를 나타냅니다.
메서드	lockOrientation()	화면 방향을 잠급니다.
	unlockOrientation()	화면 방향 잠금을 해제합니다.

팝업 창, 화면 가운데 표시하기

팝업 창의 기본 위치는 화면의 왼쪽 위이므로 화면 가운데에 배치하려면 위치를 계산해야 한다.

예를 들어, 화면의 너비 1000px, 높이 600px, 팝업 창의 너비 500px, 높이 400px이라면?



- 팝업 창의 left = (화면 너비 - 팝업 창 너비) / 2
- 팝업 창의 top = (화면 높이 - 팝업 창 높이) / 2

팝업 창, 화면 가운데 표시하기

```
<body>
  <button>팝업 창 열기</button>
  <script>
    const btn = document.querySelector("button");
    const popWidth = 600;
    const popHeight = 500;

    btn.addEventListener("click",function() {
      let left = (screen.availWidth - popWidth) / 2;
      let top = (screen.availHeight - popHeight) / 2;
      window.open("https://www.naver.com", "event"
        , `width = ${popWidth} height= ${popHeight} left = ${left} top = ${top}`);
    });
  </script>
</body>
```

history 객체

- history 객체에는 브라우저에서 [뒤로]나 [앞으로] 또는 주소 표시줄에 입력해서 방문한 사이트 주소가 배열 형태로 저장된다.
- 브라우저 히스토리는 보안 문제 때문에 읽기 전용

구분		설명
프로퍼티	length	현재 브라우저 창의 history 목록에 있는 항목의 개수, 즉 방문한 사이트 개수가 저장됩니다.
메서드	back()	history 목록에서 이전 페이지를 현재 화면으로 불러옵니다
	forward()	history 목록에서 다음 페이지를 현재 화면으로 불러옵니다
	go()	history 목록에서 현재 페이지를 기준으로 상대적인 위치에 있는 페이지를 현재 화면으로 불러옵니다. 예를 들어 history.go(1)은 다음 페이지를 가져오고, history.go(-1)은 이전 페이지를 불러옵니다.

location 객체

현재 문서의 URL 주소와 관련된 정보가
들어 있음

구분	설명	
프로퍼티	hash	URL 중에서 #로 시작하는 해시 부분의 정보를 담고 있습니다.
	host	URL의 호스트 이름과 포트 번호를 담고 있습니다.
	hostname	URL의 호스트 이름이 저장됩니다.
	href	전체 URL입니다. 이 값을 변경하면 해당 주소로 이동할 수 있습니다.
	pathname	URL 경로가 저장됩니다.
	port	URL의 포트 번호를 담고 있습니다.
	protocol	URL의 프로토콜을 저장합니다.
	password	도메인 이름 앞에 username과 password를 함께 입력해서 접속하는 사이트의 URL 일 경우에 password 정보를 저장합니다.
	search	URL 중에서 ?로 시작하는 검색 내용을 저장합니다.
	username	도메인 이름 앞에 username을 함께 입력해서 접속하는 사이트의 URL일 경우에 username 정보를 저장합니다.
메서드	assign()	현재 문서에 새 문서 주소를 할당해서 새 문서를 가져옵니다.
	reload()	현재 문서를 다시 불러옵니다.
	replace()	현재 문서의 URL을 지우고 다른 URL의 문서로 교체합니다.
	toString()	현재 문서의 URL을 문자열로 반환합니다.



내장 객체 – Array 객체

배열 만들기

1) 초깃값이 없는 경우

```
var numbers = new Array();    // 배열의 크기를 지정하지 않음  
var numbers = new Array(4);   // 배열의 크기를 지정함
```

2) 초깃값이 있는 경우

```
var numbers = ["one", "two", "three", "four"];    // 배열 선언  
var numbers = Array("one", "two", "three", "four"); // Array 객체를 사용한 배열 선언
```

배열의 크기 – Array 객체의 length 프로퍼티

```
var numbers = ["one", "two", "three", "four"]; // 배열 선언  
for(i = 0; i < numbers.length; i++) {        // 배열 선언  
    document.write("<p>" + numbers[i] + "</p>");  
}
```

배열명 옆에 마침표(.)와
length 프로퍼티를
작성하면 됩니다!

배열의 각 요소

one

two

three

four

내장 객체 – Array 객체

배열 만들기 – Array 객체의 인스턴스 만들기

1) 초깃값이 없는 경우

```
var numbers = new Array();    // 배열의 크기를 지정하지 않음  
var numbers = new Array(4);   // 배열의 크기를 지정함
```

2) 초깃값이 있는 경우

```
var numbers = ["one", "two", "three", "four"];    // 배열 선언  
var numbers = Array("one", "two", "three", "four"); // Array 객체를 사용한 배열 선언
```

배열의 크기 – Array 객체의 length 프로퍼티

```
var numbers = ["one", "two", "three", "four"]; // 배열 선언  
for(i = 0; i < numbers.length; i++) {        // 배열 선언  
    document.write("<p>" + numbers[i] + "</p>");  
}
```

배열명 옆에 마침표(.)와
length 프로퍼티를
작성하면 됩니다!

배열의 각 요소

one

two

three

four

내장 객체 – Array 객체

Array 객체의 메서드

종류	설명
concat	기존 배열에 요소를 추가해 새로운 배열을 만듭니다.
every	배열의 모든 요소가 주어진 함수에 대해 참이면 true를 반환하고 그렇지 않으면 false를 반환합니다.
filter	배열 요소 중에서 주어진 필터링 함수에 대해 true인 요소만 골라 새로운 배열을 만듭니다.
forEach	배열의 모든 요소에 대해 주어진 함수를 실행합니다.
indexOf	주어진 값과 일치하는 값이 있는 배열 요소의 첫 인덱스를 찾습니다.
join	배열 요소를 문자열로 합칩니다. 이때 구분자를 지정할 수 있습니다.
push	배열의 맨 끝에 새로운 요소를 추가한 후 새로운 length를 반환합니다.
unshift	배열의 시작 부분에 새로운 요소를 추가합니다.
pop	배열의 마지막 요소를 꺼내 그 값을 결과로 반환합니다.
shift	배열에서 첫 번째 요소를 꺼내 그 값을 결과로 반환합니다.
splice	배열에 요소를 추가하거나 삭제합니다.
slice	배열에서 특정한 부분만 잘라 냅니다.
reverse	배열의 배치 순서를 역순으로 바꿉니다.
sort	배열 요소를 지정한 조건에 따라 정렬합니다.
toString	배열에서 지정한 부분을 문자열로 반환합니다. 이때 각 요소는 쉼표(,)로 구분합니다.

내장 객체 – Array 객체

배열끼리 합치는 concat() 메서드

- 서로 다른 배열 2개를 합쳐서 새로운 배열을 만들
- 기존 배열에 영향을 주지 않음

```
// 배열 2개 합치기
var nums = [1, 2, 3];
var chars = ["a", "b", "c", "d"];

var numsChars = nums.concat(chars);
var charsNums = chars.concat(nums);
document.write("nums에 chars 합치면: ", numsChars, "<br> chars에 nums 합치면: ",
charsNums);
```

nums에 chars 합치면: 1,2,3,a,b,c,d
chars에 nums 합치면: a,b,c,d,1,2,3

배열 요소끼리 합치는 join() 메서드

- 배열 요소를 연결해서 하나의 문자열로 만들
- 요소 사이에 원하는 구분자를 넣을 수 있음.
- 구분자를 지정하지 않으면 쉼표(,)로 구분

```
// 배열 안의 요소 합치기
var nums = [1, 2, 3];
var chars = ["a", "b", "c", "d"];

.....
var string1 = nums.join();
document.write("구분자 없이: ", string1);
document.write("<br>");
var string2 = chars.join('/');
document.write("'/' 구분자 지정: ", string2);
```

구분자 없이: 1,2,3
'/' 구분자 지정: a/b/c/d

내장 객체 – Array 객체

새로운 요소를 추가하는 push(), unshift() 메서드

- push() 메서드 : 배열 맨 끝에 요소 추가
- unshift() 메서드 : 배열 맨 앞에 요소 추가
- 배열의 길이가 반환, 기존 배열이 바뀜

```
// 요소 추가하기 - 새로운 length값 반환
var nums = [1, 2, 3];
.....
var ret1 = nums.push(4, 5); // 배열 맨 끝에 새로운 요소를 추가
document.write("length: ", ret1, " | 배열: ", nums);
document.write("<br>");
var ret2 = nums.unshift(0); // 배열 맨 앞에 새로운 요소를 추가
document.write("length: ", ret2, " | 배열: ", nums);
```

length: 5 | 배열: 1,2,3,4,5
length: 6 | 배열: 0,1,2,3,4,5

배열에서 요소를 꺼내는 pop(), shift() 메서드

- pop() 메서드 : 배열 뒤쪽에서 요소를 꺼냄
- shift() 메서드 : 배열 앞쪽에서 요소를 꺼냄
- 꺼낸 요소를 반환, 기존 배열을 꺼낸 요소가 빠진 상태로 변경됨

```
// 요소 꺼내기 - 꺼낸 요소값 반환
var chars = ["a", "b", "c", "d"];
.....
var popped1 = chars.pop(); // 마지막 요소 꺼냄
document.write("꺼낸 요소: ", popped1, " | 배열: ", chars);
document.write("<br>");
var popped2 = chars.shift(); // 1번째 요소 꺼냄
document.write("꺼낸 요소: ", popped2, " | 배열: ", chars);
```

꺼낸 요소: d | 배열: a,b,c
꺼낸 요소: a | 배열: b,c

내장 객체 - Array 객체

중간에 요소를 추가하거나 삭제하는 splice() 메서드

- 배열 중간에 2개 이상의 요소를 추가하거나 삭제할 수 있음
- 새로운 배열이 결과값으로 반환됨

1) 괄호 안에 인수가 1개일 경우

인수가 지정한 인덱스의 요소부터 배열의 맨 끝 요소까지 삭제

```
// 인수가 1개인 경우
var numbers = [1, 2, 3, 4, 5];
var newNumbers = numbers.splice(2);
document.write("반환된 배열: " + newNumbers + "<br>");
document.write("변경된 배열: " + numbers);
```

반환된 배열: 3,4,5
변경된 배열: 1,2

2) 괄호 안에 인수가 2개일 경우

- 첫 번째 인수는 인덱스값이고 두 번째 인수는 삭제할 요소의 개수
- 메서드를 실행한 후에는 삭제한 요소를 반환하고, 기존 배열은 나머지 요소만 남음

```
// 인수가 2개일 경우
var study = ["html", "css", "web", "jquery"];
var newStudy = study.splice(2, 1);
document.write("반환된 배열: " + newStudy + "<br>");
document.write("변경된 배열: " + study);
```

반환된 배열: web
변경된 배열: html,css,jquery

3) 괄호 안에 인수가 3개 이상일 경우

첫 번째 인수는 배열에서 삭제할 시작 위치, 두 번째 인수는 삭제할 개수, 세 번째 인수부터는 삭제한 위치에 새로 추가할 요소를 지정

```
// 인수가 3개 이상인 경우
var newStudy2 = study.splice(2, 1, "js");
document.write("반환된 배열: " + newStudy2 + "<br>");
document.write("변경된 배열: " + study);
```

반환된 배열: jquery
변경된 배열: html,css,js

실습

저장된 배열을 역순으로...

```
<body>
  <script>
    var nums = new Array();
    var nums2 = new Array();
    var nums3 = new Array();
    const nMax = 5;

    for ( i=1; i <= nMax; i++)
      nums.push(i);
    document.write('<p>'+ nums + '</p>')

    for ( i= nums.length-1; i >= 0 ; i--)
      nums2.push(nums[i]);
    document.write('<p> push : '+ nums2 + '</p>')

    for ( i= -1; i < nums.length ; i++)
      nums3.unshift(nums[i]);
    document.write('<p>unshift : '+ nums3 + '</p>')

    nums.reverse()
    document.write('<p>reverse: '+ nums + '</p>')
  </script>
</body>
```

내장 객체 – Array 객체

기존 배열을 바꾸지 않으면서 삭제하는 slice() 메서드

- 요소를 여러 개 꺼낼 수 있음
- 요소를 삭제한 후에도 기존 배열이 바뀌지 않음

1) 괄호 안에 인수가 1개일 경우

인수가 지정한 인덱스의 요소부터 마지막 요소까지 꺼내서 반환

```
var colors = ["red", "green", "blue", "white", "black"];  
var colors2 = colors.slice(2); //인덱스 값이 2인 요소부터 마지막 요소까지 꺼내기  
document.write(colors2);
```

blue,white,black

2) 괄호 안에 인수가 2개일 경우

첫 번째 인수는 시작 인덱스, 두 번째 인수는 끝 인덱스의 직전 인덱스

```
var colors = ["red", "green", "blue", "white", "black"];  
.....  
var colors3 = colors.slice(2, 4); // 인덱스값이 2부터 3인 요소까지 꺼내기  
document.write(colors3);
```

blue,white

slice() 메서드는 기존 배열에 영향을 주지 않지만, splice() 메서드는 요소를 추가·삭제하면 기존 배열 자체가 수정됨
→ 기존 배열에서 꺼낸 요소로 새로운 배열을 만들어 사용하려면 slice() 메서드를 사용하고, 기존 배열의 일부 요소만 삭제하려면 splice() 메서드를 선택하는 것이 좋다

내장 객체 – Array 객체

foreach()

```
<body>
  <p id="demo"></p>

  <script>
    let text = "";
    const fruits = ["apple", "orange", "cherry"];

    fruits.forEach( myFunction);

    document.getElementById("demo").innerHTML = text;

    function myFunction(item, index) {
      text += index + ": " + item + "<br>";
    }
    for( i=0; i<fruits.length ; i++)
      text += i + ": " + fruits[i] + "<br>";
  </script>
</body>
```


내장 객체 – Array 객체

sort()

```
<script>
  let text = "";
  const fruits = ["apple", "orange", "cherry"];
  document.write("<p> 정렬이전 : " + fruits + "</p>");

  fruits.sort();
  document.write("<p> 정렬 기본 오름차순 : " + fruits + "</p>");
  fruits.reverse();
  document.write("<p> 정렬 내림차순 : " + fruits + "</p>");

  const points = [40, 100, 1, 5, 25, 10];
  document.write("<p> 정렬이전 : " + points + "</p>");
  points.sort( (a,b) => {return b-a})
  document.write("<p> 정렬후 : " + points + "</p>");
</script>
```

내장 객체 – Date 객체

`new Date()`

`new Date(date string)`

`new Date(year, month)`

`new Date(year, month, day)`

`new Date(year, month, day, hours)`

`new Date(year, month, day, hours, minutes)`

`new Date(year, month, day, hours, minutes, seconds)`


`new Date(year, month, day, hours, minutes, seconds, ms)`

`new Date(milliseconds)`

내장 객체 – Date 객체

Date 객체 인스턴스 만들기

현재 날짜로 설정할 경우


 Data 객체로 현재 날짜 나타내기

```
new Date();
```

특정 날짜로 설정할 경우 – 괄호 안에 날짜 또는 날짜와 시간 입력

 Data 객체로 특정 날짜 나타내기

```
new Date("2020-02-25")
```

 Data 객체로 특정 날짜와 시간 나타내기

```
new Date("2020-02-25T18:00:00")
```

자바스크립트의 날짜와 시간 입력 방식

1) YYYY-MM-DD 형식

```
new Date("2020")  
new Date("2020-02")  
new Date("2020-02-25")
```

2) YYYY-MM-DDTHH 형식

```
new Date("2020-02-25T18:00:00")  
new Date("2020-02-25T18:00:00Z")
```

3) MM/DD/YYYY 형식

```
new Date("02/25/2020")
```

4) 이름 형식

```
new Date("Mon Jan 20 2020 15:00:41 GMT+0900 (대한민국 표준시)")
```

내장 객체 – Date 객체

Date 객체의 메서드

날짜/시간 정보를 가져오는 메서드,
날짜/시간 정보를 설정하는 메서드,
날짜/시간 형식을 바꿔주는 메서드로 구분됨

	구분	설명
날짜·시간 정보 가져오기	getFullYear()	연도를 4자리 숫자로 표시합니다.
	getMonth()	0~11 사이의 숫자로 월을 표시합니다. 0부터 1월이 시작되고 11은 12월입니다.
	getDate()	1~31 사이의 숫자로 일을 표시합니다.
	getDay()	0~6 사이의 숫자로 요일을 표시합니다. 0부터 일요일이 시작되고 6은 토요일입니다.
	getTime()	1970년 1월 1일 자정 이후의 시간을 밀리 초(1/1000초)로 표시합니다.
	getHours()	0~23 사이의 숫자로 시를 표시합니다.
	getMinutes()	0~59 사이의 숫자로 분을 표시합니다.
	getSeconds()	0~59 사이의 숫자로 초를 표시합니다.
	getMilliseconds()	0~999 사이의 숫자로 밀리초를 표시합니다.
날짜·시간 설정하기	setFullYear()	연도를 4자리 숫자로 설정합니다.
	setMonth()	0~11 사이의 숫자로 월을 설정합니다. 0부터 1월이 시작되고 11은 12월입니다.
	setDate()	1~31 사이의 숫자로 일을 설정합니다.
	setTime()	1970년 1월 1일 자정 이후의 시간을 밀리초로 설정합니다.
	setHours()	0~23 사이의 숫자로 시를 설정합니다.
	setMinutes()	0~59 사이의 숫자로 분을 설정합니다.
	setSeconds()	0~59 사이의 숫자로 초를 설정합니다.
	setMilliseconds()	0~999 사이의 숫자로 밀리초를 설정합니다.
날짜·시간 형식 바꾸기	toLocaleString()	현재 날짜와 시간을 현지 시간(local time)으로 표시합니다.
	toString()	Data 객체 타입을 문자열로 표시합니다.

실습문제

- 디지털 시계 만들기
- p. 569 페이지 날짜 계산하는 프로그램 만들기

내장 객체 – Math 객체

Math 객체의 특징

- 수학 계산과 관련된 메서드가 많이 포함되어 있지만 수학식에서만 사용하는 것은 아님.
- 무작위 수가 필요하거나 반올림이 필요한 프로그램 등에서도 Math 객체의 메서드 사용함
- Math 객체는 인스턴스를 만들지 않고 프로퍼티와 메서드 사용

Math 객체의 프로퍼티

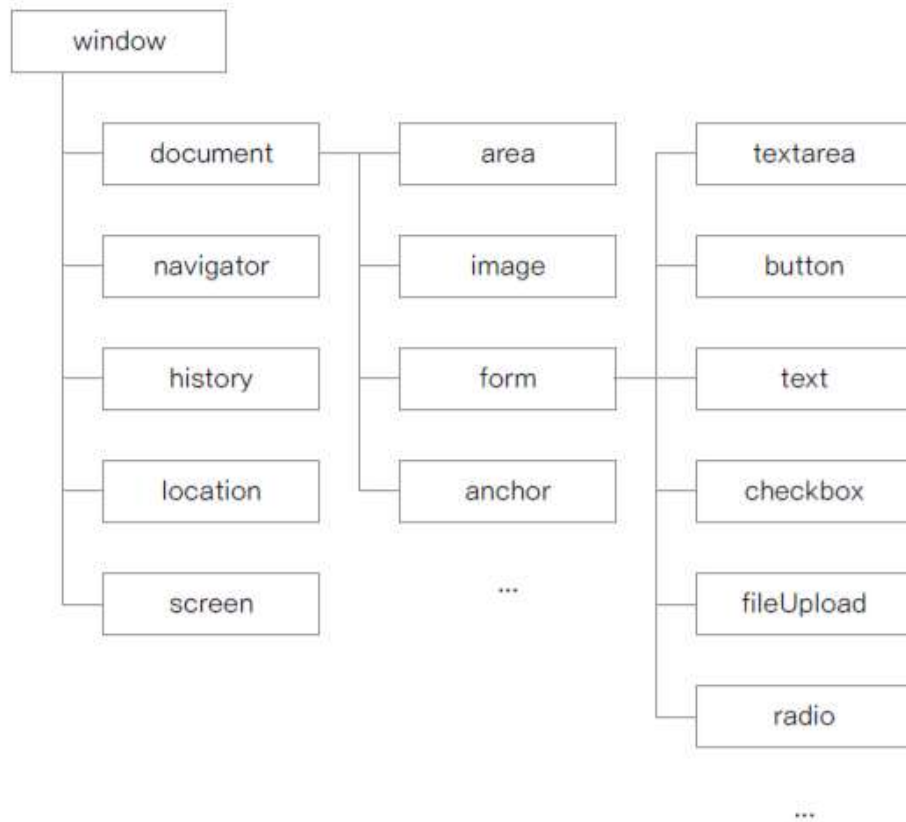
종류	설명
E	오일러 상수
PI	원주율(π) (약 3.141592653589793의 값)
SQRT2	$\sqrt{2}$ (약 1.4142135623730951의 값)
SQRT1_2	$1/\sqrt{2}$ (약 0.7071067811865476의 값)
LN2	\log_2 (약 0.6931471805599453의 값)
LN10	\log_{10} (약 2.302585092994046의 값)
LOG2E	$\log_2 e$ (약 1.4426950408889634의 값)
LOG10E	$\log_{10} e$ (약 0.4342944819032518의 값)

Math 객체의 메서드

종류	설명
abs()	절댓값을 반환합니다.
acos()	아크 코사인(arc cosine)값을 반환합니다.
asin()	아크 사인(arc sine)값을 반환합니다.
atan()	아크 탄젠트(arc tangent)값을 반환합니다.
atan2()	아크 탄젠트(arc tangent)값을 반환합니다.
ceil()	매개변수의 소수점 이하 부분을 올립니다.
cos()	코사인(cosine)값을 반환합니다.
exp()	지수 함수를 나타냅니다.
floor()	매개변수의 소수점 이하 부분을 버립니다.
log()	매개변수에 대한 로그(log)값을 반환합니다.
max()	매개변수 중 최댓값을 반환합니다.
min()	매개변수 중 최솟값을 반환합니다.
pow()	매개변수의 지수값을 반환합니다.
random()	0과 1 사이의 무작위 수를 반환합니다.
round()	매개변수의 소수점 이하 부분을 반올림합니다.
sin()	사인(sine)값을 반환합니다.
sqrt()	매개변수에 대한 제곱근을 반환합니다.
tan()	탄젠트(tangent)값을 반환합니다.

브라우저 관련 객체

브라우저 관련 객체의 계층 구조



종류	설명
window	브라우저 창이 열릴 때마다 하나씩 만들어집니다. 브라우저 창 안의 요소 중에서 최상위에 있습니다.
document	웹 문서마다 하나씩 있으며 <body> 태그를 만나면 만들어집니다. HTML 문서의 정보가 담겨 있습니다.
navigator	현재 사용하는 브라우저의 정보가 들어 있습니다.
history	현재 창에서 사용자의 방문 기록을 저장합니다.
location	현재 페이지의 URL 정보가 담겨 있습니다.
screen	현재 사용하는 화면 정보를 다룹니다.

브라우저 관련 객체 – window 객체

window 객체의 프로퍼티

주로 웹 브라우저 창의 정보를 가져오거나 값을 바꿀 때 사용

종류	설명
document	브라우저 창에 표시된 웹 문서에 접근할 수 있습니다.
frameElement	현재 창이 다른 요소 안에 포함되어 있을 경우 그 요소를 반환하고, 반대로 포함되어 있지 않으면 null을 반환합니다.
innerHeight	내용 영역의 높이를 나타냅니다.
innerWidth	내용 영역의 너비를 나타냅니다.
localStorage	웹 브라우저에서 데이터를 저장하는 로컬 스토리지를 반환합니다.
location	window 객체의 위치 또는 현재 URL을 나타냅니다.
name	브라우저 창의 이름을 가져오거나 수정합니다.
outerHeight	브라우저 창의 바깥 높이를 나타냅니다.
outerWidth	브라우저 창의 바깥 너비를 나타냅니다.
pageXOffset	스크롤했을 때 수평으로 이동하는 픽셀 수로 scrollX와 같습니다.
pageYOffset	스크롤했을 때 수직으로 이동하는 픽셀 수로 scrollY와 같습니다.
parent	현재 창이나 서브 프레임의 부모입니다.
screenX	브라우저 창의 왼쪽 테두리가 모니터 왼쪽 테두리에서 떨어져 있는 거리를 나타냅니다.
screenY	브라우저 창의 위쪽 테두리가 모니터 위쪽 테두리에서 떨어져 있는 거리를 나타냅니다.
scrollX	스크롤했을 때 수평으로 이동하는 픽셀 수를 나타냅니다.
scrollY	스크롤했을 때 수직으로 이동하는 픽셀 수를 나타냅니다.
sessionStorage	웹 브라우저에서 데이터를 저장하는 세션 스토리지를 반환합니다.

window 객체의 메서드

window 객체는 기본 객체이므로 'window.'를 생략하고 메서드 이름만 사용해도 됨

종류	설명
alert()	알림 창을 표시합니다.
blur()	현재 창에서 포커스를 제거합니다.
close()	현재 창을 닫습니다.
confirm()	[확인], [취소] 버튼이 있는 확인 창을 표시합니다.
focus()	현재 창에 포커스를 부여합니다.
moveBy()	현재 창을 지정한 크기만큼 이동합니다.
moveTo()	현재 창을 지정한 좌표로 이동합니다.
open()	새로운 창을 엽니다.
postMessage()	메시지를 다른 창으로 전달합니다.
print()	현재 문서를 인쇄합니다.
prompt()	프롬프트 창에 입력한 텍스트를 반환합니다.
resizeBy()	지정한 크기만큼 현재 창의 크기를 조절합니다.
resizeTo()	동적으로 브라우저 창의 크기를 조절합니다.
scroll()	문서에서 특정 위치로 스크롤합니다.
scrollBy()	지정한 크기만큼씩 스크롤합니다.
scrollTo()	지정한 위치까지 스크롤합니다.
sizeToContent()	내용에 맞게 창의 크기를 맞춥니다.
stop()	로딩을 중지합니다.

브라우저 관련 객체 – navigator 객체, history 객체

navigator 객체

- 사용하는 브라우저가 많아지고, 웹 애플리케이션이 등장하면서 navigator 객체에 여러 프로퍼티가 등장하고 있음.
- 일부 브라우저에서만 지원하는 프로퍼티도 있음

주요 프로퍼티

종류	설명
battery	배터리 충전 상태를 알려 줍니다.
cookieEnabled	쿠키 정보를 무시하면 false, 허용하면 true를 반환합니다.
geolocation	모바일 기기를 이용한 위치 정보를 나타냅니다.
language	브라우저 UI의 언어 정보를 나타냅니다.
oscpu	현재 운영체제 정보를 나타냅니다.
userAgent	현재 브라우저 정보를 담고 있는 사용자 에이전트 문자열입니다.

history 객체

방문한 사이트 주소가 배열 형태로 저장됨

프로퍼티와 메서드

구분		설명
프로퍼티	length	현재 브라우저 창의 history 목록에 있는 항목의 개수, 즉 방문한 사이트 개수가 저장됩니다.
메서드	back()	history 목록에서 이전 페이지를 현재 화면으로 불러옵니다
	forward()	history 목록에서 다음 페이지를 현재 화면으로 불러옵니다
	go()	history 목록에서 현재 페이지를 기준으로 상대적인 위치에 있는 페이지를 현재 화면으로 불러옵니다. 예를 들어 history.go(1)은 다음 페이지를 가져오고, history.go(-1)은 이전 페이지를 불러옵니다.

브라우저 관련 객체 – location 객체, screen 객체

location 객체

- 현재 문서의 URL 주소 정보가 담겨 있음
- 이 정보를 편집해서 브라우저 창에 열 사이트/문서 지정

구분		설명
프로퍼티	hash	URL 중에서 #로 시작하는 해시 부분의 정보를 담고 있습니다.
	host	URL의 호스트 이름과 포트 번호를 담고 있습니다.
	hostname	URL의 호스트 이름이 저장됩니다.
	href	전체 URL입니다. 이 값을 변경하면 해당 주소로 이동할 수 있습니다.
	pathname	URL 경로가 저장됩니다.
	port	URL의 포트 번호를 담고 있습니다.
	protocol	URL의 프로토콜을 저장합니다.
	password	도메인 이름 앞에 username과 password를 함께 입력해서 접속하는 사이트의 URL 일 경우에 password 정보를 저장합니다.
	search	URL 중에서 ?로 시작하는 검색 내용을 저장합니다.
	username	도메인 이름 앞에 username을 함께 입력해서 접속하는 사이트의 URL일 경우에 username 정보를 저장합니다.
메서드	assign()	현재 문서에 새 문서 주소를 할당해서 새 문서를 가져옵니다.
	reload()	현재 문서를 다시 불러옵니다.
	replace()	현재 문서의 URL을 지우고 다른 URL의 문서로 교체합니다.
	toString()	현재 문서의 URL을 문자열로 반환합니다.

screen 객체

사용자의 화면 크기, 정보

구분		설명
프로퍼티	availHeight	UI 영역(예를 들어 윈도우의 작업 표시줄이나 Mac의 독)을 제외한 영역의 높이를 나타냅니다.
	availWidth	UI 영역을 제외한 내용 표시 영역의 너비를 나타냅니다.
	colorDepth	화면에서 픽셀을 렌더링할 때 사용하는 색상 수를 나타냅니다.
	height	UI 영역을 포함한 화면의 높이를 나타냅니다.
	orientation	화면의 현재 방향을 나타냅니다.
	pixelDepth	화면에서 픽셀을 렌더링할 때 사용하는 비트 수를 나타냅니다.
	width	UI 영역을 포함한 화면의 너비를 나타냅니다.
메서드	lockOrientation()	화면 방향을 잠급니다.
	unlockOrientation()	화면 방향 잠금을 해제합니다.

문자열

원시 유형에서 메서드 사용하기

- 자료형 중 단순히 값만 가지고 있을 때 → 원시 유형(primitive type)
- 원시 유형에서도 프로퍼티와 메서드를 사용할 수 있다.

```
str = "hello"  
str.length
```

str은 문자열(string)인데,
length 프로퍼티는 어디에서 온 것일까?

숫자형과 논리형, 문자열 유형은 별도로 객체가 만들어져 있다.

Number, Boolean, String 객체 ← 래퍼 객체라고 부름

boolean → Boolean 객체

number → Number 객체

string → String 객체

오토박싱(autoboxing)

number나 boolean, string 같은 원시 유형에서 프로퍼티나 메서드를 사용하면 일시적으로 원시 유형을 해당 객체로 변환한다.

프로퍼티나 메서드의 사용이 끝나면 해당 객체는 메모리에서 사라진다.

필요할 때에만 임시로 객체로 바꿔 사용하고 사용이 끝나면 다시 원시 유형으로 되돌아오는데, 이것을 오토박싱이라고 한다.

문자열

문자열의 길이 - length

문자열의 길이를 찾을 때에는 length 프로퍼티 사용

`문자열.length`

```
let str = "Good morning!";  
let greeting = "안녕하세요?"  
str.length // 13  
greeting.length // 6
```

특정 위치의 문자에 접근하기 - charAt()

- 방법 1) 대괄호 사용 (ES6 이후)
- 방법 2) charAt() 메서드 사용 `문자열.charAt(위치)`

```
str = "Good morning!";  
str.charAt(3) // "d"  
str[5] // "m"
```

워드 카운트

```
<body>
  <p id="content">Lorem, ipsum dolor sit amet consectetur adipisicing elit. Nobis,
voluptates quibusdam facilis vitae, aliquid omnis possimus incidunt sed perspiciatis
nihil cum mollitia fuga dignissimos voluptas voluptatibus, beatae iste consectetur
animi!</p>
  <p id="word_count"> </p>
  <script>
    let text = document.querySelector('#content').innerText;
    text = text.trim().toLowerCase();
    let curIndex = 0;
    let wordCount = 0; // 단어의 카운트 변수
    let findWord = "ipsum";
    console.log(text);

    // 반복 문을 이용해 찾을 문자열을 검사한다.
    while(text.includes(findWord, curIndex) ){
      var findIndex = text.indexOf(findWord, curIndex);
      curIndex += findIndex + findWord.length;
      wordCount++;
    }
    document.querySelector('#word_count').innerText = wordCount;
  </script>
</body>
```