



# 9 문서 객체 모델

모던 웹을 위한 Javascript jQuery 입문

## ❖ 문서 객체 모델 (DOM)

- 넓은 의미로 웹 브라우저가 HTML 페이지 인식하는 방식
- 좁은 의미로는 document 객체와 관련된 객체 집합
- 사용시 HTML 페이지에 태그를 추가, 수정, 제거할 수 있음
- '태그' - HTML 페이지에 존재하는 html이나 body 태그
- 문서 객체 - 태그를 자바스크립트에서 이용할 수 있는 객체로 만든 것

코드 9-1 간단한 HTML 페이지

```
<!DOCTYPE html>
<html>
<head>
  <title>INDEX</title>
  <scriptX/script>
</head>
<body>
  <h1 id="header">HEADER</h1>
</body>
</html>
```



# 9.1 DOM 관련 용어

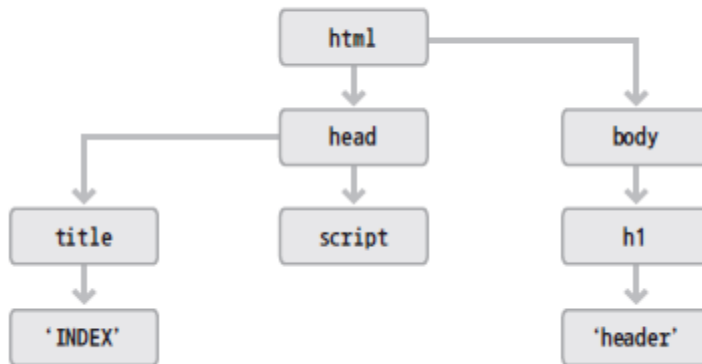
## ❖ 예제 코드 9-2

- document 객체의 getElementById() 메서드
  - 문서 객체를 자바스크립트로 가져와 조작 가능

코드 9-2 문서 객체

```
<script>
  window.onload = function () {
    var header = document.getElementById('header');
  };
</script>
```

그림 9-1 코드 9-2의 DOM



# 9.1 DOM 관련 용어

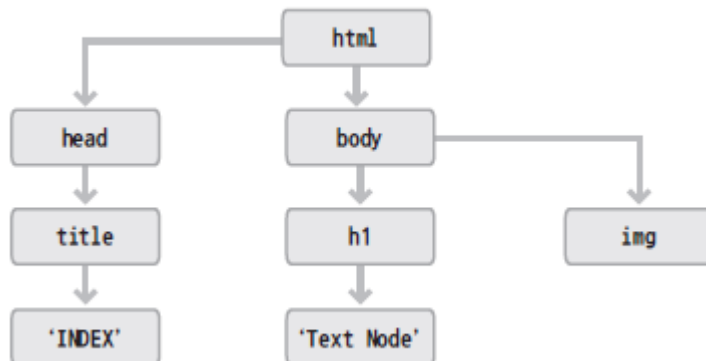
## ❖ 노드

- 요소 노드 (Element Node, HTML xorm)
- 텍스트 노드 (Text Node, 요소 노드 안의 글자)

코드 9-3 요소 노드와 텍스트 노드

```
<!DOCTYPE html>
<html>
<head>
  <title>INDEX</title>
</head>
<body>
  <h1>Text Node</h1>
  
</body>
</html>
```

그림 9-2 코드 9-3의 DOM



## 9.2 문서 객체 만들기 (1)

### ❖ 문서 객체

- 텍스트 노드를 갖는 문서 객체
- 텍스트 노드를 갖지 않는 문서 객체로 구분
- 예제]
  - 텍스트 노드를 갖는 h1 태그를 생성하면서 문서 객체 생성
  - body 태그 구성

코드 9-4 body 태그 구성

```
<body>
```

```
</body>
```

표 9-1 document 객체의 노드 생성 메서드

메서드 이름	설명
createElement(tagName)	요소 노드를 생성합니다.
createTextNode(text)	텍스트 노드를 생성합니다.



## 9.2 문서 객체 만들기 (1)

### ❖ 문서 객체 동적 생성

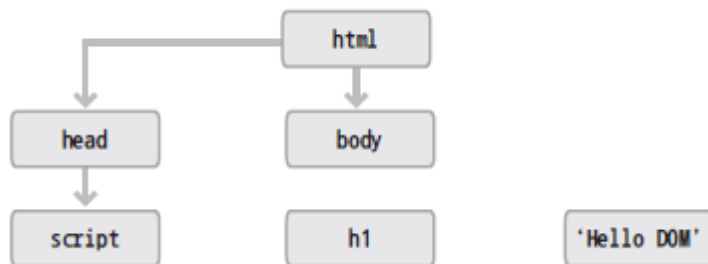
- 화면에 출력하기 위해서는 문서 객체를 body 문서 객체에 연결 필요

코드 9-5 문서 객체 생성

```
<script>
  window.onload = function () {
    // 변수를 선언합니다.
    var header = document.createElement('h1');
    var textNode = document.createTextNode('Hello DOM');

  };
</script>
```

그림 9-3 현재 상황



## 9.2 문서 객체 만들기 (1)

### ❖ 화면에 문서 객체 출력

- 생성한 문서 객체를 body 문서 객체에 연결
- 생성한 요소 노드와 텍스트 노드도 연결
  - 노드와 노드를 연결할 때는 메서드 사용

표 9-2 문서 객체의 연결 메서드

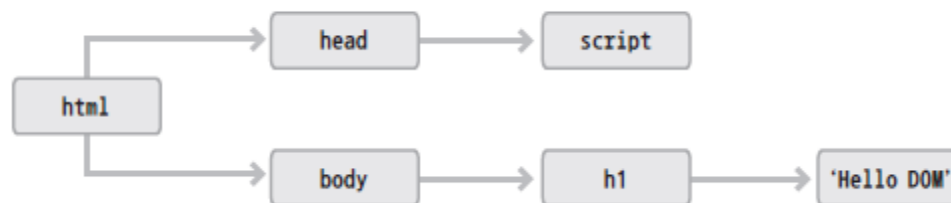
메서드 이름	설명
appendChild(node)	객체에 노드를 연결합니다.

코드 9-6 문서 객체 연결

```
<script>
  window.onload = function () {
    // 변수를 선언합니다.
    var header = document.createElement('h1');
    var textNode = document.createTextNode('Hello DOM');

    // 노드를 연결합니다.
    header.appendChild(textNode);
    document.body.appendChild(header);
  };
</script>
```

그림 9-4 HTML 페이지의 DOM 형태



## 9.3 문서 객체 만들기 (2)

### ❖ 텍스트 노드 갖지 않는 노드 생성하는 방법

- 대표적인 HTML 태그는 img

코드 9-7 문서 객체 생성

```
<script>
    window.onload = function () {
        // 변수를 선언합니다.
        var img = document.createElement('img');

        // 노드를 연결합니다.
        document.body.appendChild(img);
    };
</script>
```

코드 9-8 문서 객체의 속성 지정 (1)

```
<script>
    window.onload = function () {
        // 변수를 선언합니다.
        var img = document.createElement('img');
        img.src = 'Penguins.jpg';
        img.width = 500;
        img.height = 350;

        // 노드를 연결합니다.
        document.body.appendChild(img);
    };
</script>
```

그림 9-5 크롬을 사용한 요소 검사

```
▼<body>
  
</body>
```





## 9.3 문서 객체 만들기 (2)

### ❖ 웹 브라우저가 지원하지 않는 속성

- 표 9-3의 메서드 사용해서 속성을 집어 넣음

표 9-3 문서 객체의 속성 메서드

메서드 이름	설명
setAttribute(name, value)	객체의 속성을 지정합니다.
getAttribute(name)	객체의 속성을 가져옵니다.

코드 9-9 문서 객체의 속성 지정 (2)

```
<script>
  window.onload = function () {
    // 변수를 선언합니다.
    var img = document.createElement('img');
    img.setAttribute('src', 'Penguins.jpg');
    img.setAttribute('width', 500);
    img.setAttribute('height', 350);

    // setAttribute() 메서드를 사용하지 않으면 불가능합니다.
    img.setAttribute('data-property', 350);

    // 노드를 연결합니다.
    document.body.appendChild(img);
  };
</script>
```

그림 9-6 크롬을 사용한 요소 검사

```
▼<body>
  
</body>
```

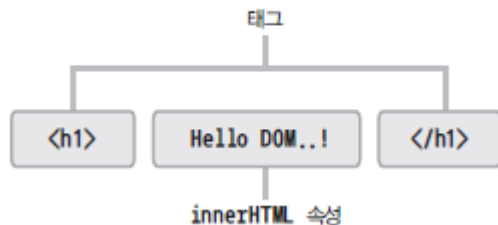


## 9.4 문서 객체 만들기 (3)

### ❖ 문서 객체의 innerHTML 속성 사용해 객체 생성

- 문자열 선언하고 body 문서 객체의 innerHTML 객체에 넣으면 생성

그림 9-7 문서 객체의 innerHTML 속성



코드 9-10 문서 객체의 innerHTML 속성

```
<script>
    window.onload = function () {
        // 변수를 선언합니다.
        var output = '';

        // innerHTML 속성에 문자열을 할당합니다.
        document.body.innerHTML = output;
    };
</script>
```



## 9.4 문서 객체 만들기 (3)

### ❖ 문서 객체의 innerHTML 속성 사용해 객체 생성

- 문자열에 HTML 태그 직접 입력
- body 태그에 내용을 추가하고 싶을 때
  - innerHTML 속성에 복합 대입 연산자 사용해 문자열 추가
    - **document.body.innerHTML += '<h1>Document Object Model</h1>';**

코드 9-11 innerHTML 속성을 사용한 문서 객체 생성

```
<script>
window.onload = function () {
    // 변수를 선언합니다.
    var output = '';
    output += '<ul>';
    output += '    <li>JavaScript</li>';
    output += '    <li>jQuery</li>';
    output += '    <li>Ajax</li>';
    output += '</ul>';

    // innerHTML 속성에 문자열을 할당합니다.
    document.body.innerHTML = output;
};
</script>
```

그림 9-8 동적으로 생성된 태그

- JavaScript
- jQuery
- Ajax

### Document Object Model



## 9.5 문서 객체 가져오기 (1)

### ❖ HTML 태그를 자바스크립트로 가져오는 방법

- 이미 웹 페이지에 존재하는 객체 가져오기

표 9-4 document 객체의 노드 추출 메서드

메서드 이름	설명
getElementById(id)	태그의 id 속성이 id와 일치하는 문서 객체를 가져옵니다.

코드 9-12 body 태그 구성

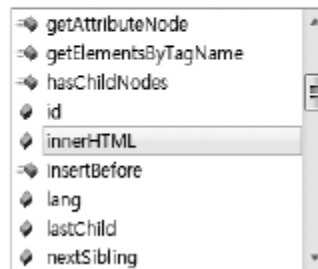
```
<body>
  <h1 id="header_1">Header</h1>
  <h1 id="header_2">Header</h1>
</body>
```

코드 9-13 getElementById() 메서드

```
<script>
  window.onload = function () {
    // 문서 객체를 가져옵니다.
    var header1 = document.getElementById('header_1');
    var header2 = document.getElementById('header_2');
  };
</script>
```

그림 9-9 header1 객체의 속성

```
window.onload = function () {
  var header1 = document.getElementById('header_1');
  var header2 = document.getElementById('header_2');
  header1.innerHTML
};
```



## 9.5 문서 객체 가져오기 (1)

### ❖ 문서 객체의 innerHTML 속성 변경

코드 9-14 문서 객체의 innerHTML 속성 변경

```
<script>
  window.onload = function () {
    // 문서 객체를 가져옵니다.
    var header1 = document.getElementById('header_1');
    var header2 = document.getElementById('header_2');

    // 문서 객체의 속성을 변경합니다.
    header1.innerHTML = 'with getElementById()';
    header2.innerHTML = 'with getElementById()';
  };
</script>
```

그림 9-10 innerHTML 속성

**with getElementById()**  
**with getElementById()**



## 9.6 문서 객체 가져오기 (2)

### ❖ 문서 객체 가져오기

- document 객체의 getElementById() 메서드
  - 한 번에 한 가지 문서 객체만 가져올 수 있음
- document 객체가 가지는 표 9-5의 메서드 사용
  - 한 번에 여러 개의 문서 객체를 가져올 수 있음

표 9-5 document 객체의 노드 추출 메서드

메서드 이름	설명
getElementsByName(name)	태그의 name속성이 name과 일치하는 문서 객체를 배열로 가져옵니다.
getElementsByTagName(tagName)	tagName과 일치하는 문서 객체를 배열로 가져옵니다.



## 9.6 문서 객체 가져오기 (2)

### ❖ 문서 객체 가져오기 예제

- 변수 headers는 문서 객체 가지는 배열

코드 9-15 document.getElementsByTagName() 메서드

```
<!DOCTYPE html>
<html>
<head>
  <title>Index</title>
  <script>
    window.onload = function () {
      // 문서 객체를 가져옵니다.
      var headers = document.getElementsByTagName('h1');
    };
  </script>
</head>
<body>
  <h1>Header</h1>
  <h1>Header</h1>
</body>
</html>
```

코드 9-16 문서 객체 배열의 사용

```
<script>
  window.onload = function () {
    // 문서 객체를 가져옵니다.
    var headers = document.getElementsByTagName('h1');
    headers[0].innerHTML = 'With getElementsByTagName()';
    headers[1].innerHTML = 'With getElementsByTagName()';
  };
</script>
```



## 9.6 문서 객체 가져오기 (2)

### ❖ 문서 객체 가져오기 예제

#### ■ 문서 객체 속성 변경

코드 9-17 getElementByTagName() 메서드

```
<script>
  window.onload = function () {
    // 문서 객체를 가져옵니다.
    var headers = document.getElementsByTagName('h1');

    for (var i = 0; i < headers.length; i++) {
      // 문서 객체의 속성을 변경합니다.
      headers[i].innerHTML = 'With getElementByTagName()';
    }
  };
</script>
```

그림 9-11 getElementByTagName() 메서드를 사용한 문서 객체 선택

**With getElementByTagName()**

**With getElementByTagName()**





## 9.6 문서 객체 가져오기 (2)

### ❖ 문서 객체 가져오기 예제

- 잘못된 문서 객체 배열 사용의 예
  - 문서 객체 배열에 for in 반복문 사용
    - 문서 객체 이외의 속성에도 접근

코드 9-18 잘못된 문서 객체 배열의 사용

```
<script>
  window.onload = function () {
    // 문서 객체를 가져옵니다.
    var headers = document.getElementsByTagName('h1');

    // 출력합니다.
    var output = '';
    for (var i in headers) {
      output += i + '\n';
    }
    alert(output);
  };
</script>
```

그림 9-12 실행 결과



## ❖ 문서 객체의 style 속성 사용

- 해당 문서 객체의 스타일 변경 가능

코드 9-19 body 태그 구성

```
<body>
  <h1 id="header">Header</h1>
</body>
```

코드 9-20 문서 객체의 스타일 지정

```
<script>
  window.onload = function () {
    // 문서 객체를 가져옵니다.
    var header = document.getElementById('header');

    // 문서 객체의 스타일을 바꿔줍니다.
    header.style.border = '2px Solid Black';
    header.style.color = 'Orange';
    header.style.fontFamily = 'Helvetica';

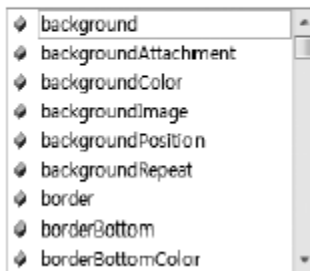
  };
</script>
```

그림 9-14 오페라를 사용한 요소 검사

```
<!DOCTYPE html>
<html>
  <head>
  <body>
    <h1 id="header" style="border-top-color: black; border-left-color: black; border-right-color: black;
border-bottom-color: black; border-top-width: 2px; border-left-width: 2px; border-right-width: 2px;
border-bottom-width: 2px; border-top-style: solid; border-left-style: solid; border-right-style: solid;
border-bottom-style: solid; color: orange; font-family: "Helvetica"">
  </body>
</html>
```

그림 9-13 문서 객체의 style 속성

// 문서 객체를 가져옵니다.  
var header = document.getElementById('header');  
header.style.



## ❖ 9.8 문서 객체 제거

표 9-6 문서 객체의 제거 메서드

메서드 이름	설명
removeChild(child)	문서 객체의 자식 노드를 제거합니다.

코드 9-21 body 태그 구성

```
<body>
  <h1 id="will_remove">Header</h1>
</body>
```

코드 9-22 문서 객체의 제거

```
<script>
  window.onload = function () {
    // 문서 객체를 가져옵니다.
    var willRemove = document.getElementById('will_remove');

    // 문서 객체를 제거합니다.
    document.body.removeChild(willRemove);
  };
</script>
```



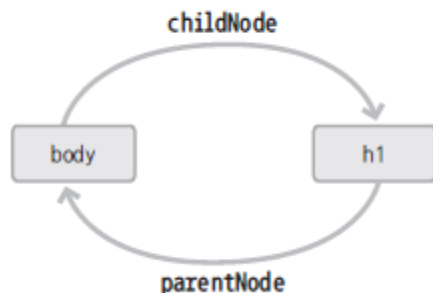
### ❖ 문서 객체 제거방법

#### ■ 예제 코드의 문제점

- body 문서 객체 바로 아래에 제거하고자 하는 문서 객체 있어 가능
- 일반적인 문서 객체 제거 코드

- `willRemove.parentNode.removeChild(willRemove);`
- h1 태그에서 부모 노드로 이동한 후
- 부모 노드에서 자식 노드 삭제

그림 9-15 parentNode와 childNode의 관계



## 9.9 문서 객체를 사용한 시계

### ❖ 문서 객체를 사용해 현재 시간 표시하는 시계 작성

코드 9-23 body 태그 구성

```
<body>
  <h1 id="clock">/h1>
</body>
```

코드 9-24 문서 객체 설정

```
<script>
  window.onload = function () {
    // 변수를 선언합니다.
    var clock = document.getElementById('clock');
  };
</script>
```

코드 9-25 현재 시간 표시

```
<script>
  window.onload = function () {
    // 변수를 선언합니다.
    var clock = document.getElementById('clock');

    // 1초마다 함수를 실행합니다.
    setInterval(function () {
      var now = new Date();
      clock.innerHTML = now.toString();
    }, 1000);
  };
</script>
```

그림 9-16 1초마다 시간이 변화

**Mon Jul 4 00:42:18 UTC+0900 2011**



# 9.10 문서 객체를 사용한 움직임 구현

## ❖ 궤도 따라 이동하는 지구와 달 구현

코드 9-26 body 태그 구성

```
<body>
  <h1 id="sun">☉</h1>
  <h1 id="earth">🌍</h1>
  <h1 id="moon">🌙</h1>
</body>
```

코드 9-27 문서 객체

```
<script>
  window.onload = function () {
    // 변수를 선언합니다.
    var sun = document.getElementById('sun');
    var earth = document.getElementById('earth');
    var moon = document.getElementById('moon');

  };
</script>
```

코드 9-28 문서 객체의 스타일 속성 설정

```
// 문서 객체를 가져오고 설정합니다.
var sun = document.getElementById('sun');
var earth = document.getElementById('earth');
var moon = document.getElementById('moon');

sun.style.position = 'absolute';
earth.style.position = 'absolute';
moon.style.position = 'absolute';

sun.style.left = 250 + 'px';
sun.style.top = 200 + 'px';
```



## 9.10 문서 객체를 사용한 움직임 구현

### ❖ 궤도 따라 이동하는 지구와 달 구현

- 회전 위한 변수 선언과 함수 사용 – 삼각함수 사용

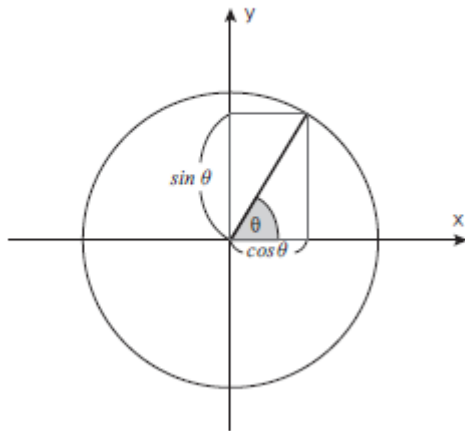
코드 9-29 setInterval() 함수 사용

```
<script>
  window.onload = function () {
    // 문서 객체를 가져오고 설정합니다.
    /* 생략 */
    // 변수를 선언합니다.
    var earthAngle = 0;
    var moonAngle = 0;

    // 애니메이션을 시작합니다.
    setInterval(function () {

      }, 1000 / 30);
  };
</script>
```

그림 9-17 각도를 알면 sin 값과 cos 값으로 궤도의 좌표를 구할 수 있습니다.



## 9.10 문서 객체를 사용한 움직임 구현

### ❖ 궤도 따라 이동하는 지구와 달 구현

#### ■ 문서 객체 회전 구현

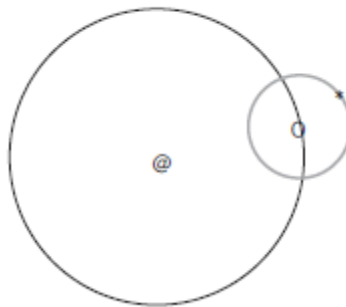
코드 9-31 문서 객체의 회전

```
// 애니메이션을 시작합니다.
setInterval(function () {
    // 각도를 사용해 지구와 달의 좌표를 구합니다.
    var earthLeft = 250 + 150 * Math.cos(earthAngle);
    var earthTop = 200 + 150 * Math.sin(earthAngle);
    var moonLeft = earthLeft + 50 * Math.cos(moonAngle);
    var moonTop = earthTop + 50 * Math.sin(moonAngle);

    // 위치를 이동합니다.
    earth.style.left = earthLeft + 'px';
    earth.style.top = earthTop + 'px';
    moon.style.left = moonLeft + 'px';
    moon.style.top = moonTop + 'px';

    // 각도를 변경합니다.
    earthAngle += 0.1;
    moonAngle += 0.3;
}, 1000 / 30);
```

그림 9-18 선을 따라 회전하는 글자





## 9.10 문서 객체를 사용한 움직임 구현

### ❖ 궤도 따라 이동하는 지구와 달 구현 – 전체 코드

코드 9-32 전체 코드

```
<!DOCTYPE html>
<html>
<head>
  <script>
    window.onload = function () {
      // 문서 객체를 가져오고 설정합니다.
      var sun = document.getElementById('sun');
      var earth = document.getElementById('earth');
      var moon = document.getElementById('moon');

      sun.style.position = 'absolute';
      earth.style.position = 'absolute';
      moon.style.position = 'absolute';

      sun.style.left = 250 + 'px';
      sun.style.top = 200 + 'px';

      // 변수를 선언합니다.
      var earthAngle = 0;
      var moonAngle = 0;
```



## 9.10 문서 객체를 사용한 움직임 구현

### ❖ 궤도 따라 이동하는 지구와 달 구현 – 전체 코드 (2)

```
// 애니메이션을 시작합니다.
setInterval(function () {
    // 각도를 사용해 지구와 달의 좌표를 구합니다.
    var earthLeft = 250 + 150 * Math.cos(earthAngle);
    var earthTop = 200 + 150 * Math.sin(earthAngle);
    var moonLeft = earthLeft + 50 * Math.cos(moonAngle);
    var moonTop = earthTop + 50 * Math.sin(moonAngle);

    // 위치를 이동합니다.
    earth.style.left = earthLeft + 'px';
    earth.style.top = earthTop + 'px';
    moon.style.left = moonLeft + 'px';
    moon.style.top = moonTop + 'px';

    // 각도를 변경합니다.
    earthAngle += 0.1;
    moonAngle += 0.3;
}, 1000 / 30);
};
</script>
</head>
<body>
    <h1 id="sun">☉</h1>
    <h1 id="earth">🌍</h1>
    <h1 id="moon">🌙</h1>
</body>
</html>
```



## ❖ HTML 페이지 구성

코드 9-33 HTML 페이지 구성

```
<!DOCTYPE html>
<html>
<head>
  <!-- 보조 함수 -->
  <script>

  </script>
  <!-- 생성자 함수 -->
  <script>

  </script>
  <!-- window.onload -->
  <script>

  </script>
</head>
<body>

</body>
</html>
```



## ❖ 보조 함수 구성

### ■ randomSpeed() 함수

- $\text{Math.random()} * \text{maxSpeed} - \text{Math.random()} * \text{maxSpeed}$

– 양수와 음수 값으로 랜덤한 값 발생시키려 사용

코드 9-34 보조 함수 선언

```
<!-- 보조 함수 -->
<script>
    // 랜덤한 정수를 생성합니다.
    function nextRandomInteger(limit) {
        return Math.round(Math.random() * limit);
    }

    // 랜덤한 알파벳을 리턴하는 함수입니다.
    var alphabet = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
    function randomAlphabet() {
        return alphabet.charAt(nextRandomInteger(25));
    }

    // 양음으로 랜덤한 속도를 생성하는 함수입니다.
    function randomSpeed(maxSpeed) {
        return Math.random() * maxSpeed - Math.random() * maxSpeed;
    }
</script>
```



## ❖ 움직이는 글자 표현할 생성자 함수 MovingText()

코드 9-35 MovingText 생성자 함수 선언

```
<!-- 생성자 함수 -->
<script>
  // MovingText의 생성자 함수
  var canvasWidth = 700;
  var canvasHeight = 500;
  function MovingText() {
    // 위치와 속도 속성
    this.x = nextRandomInteger(canvasWidth);
    this.y = nextRandomInteger(canvasHeight);
    this.vX = randomSpeed(10);
    this.vY = randomSpeed(10);

    // 문서 객체를 생성합니다.
    this.body = document.createElement('h1');
    this.body.innerHTML = randomAlphabet();
    this.body.style.position = 'absolute';

    // 문서 객체를 document.body에 추가합니다.
    document.body.appendChild(this.body);
  }
</script>
```



## ❖ MovingText() 메서드 객체 추가

코드 9-36 MovingText 객체의 메서드 추가

```
<!-- 생성자 함수 -->
<script>
    // MovingText의 생성자 함수
    var canvasWidth = 700;
    var canvasHeight = 500;
    function MovingText() {
        /* 생략 */
    }

    MovingText.prototype.move = function () {
        // 범위 검사
        if (this.x < 0 || this.x > canvasWidth) { this.vX *= -1; }
        if (this.y < 0 || this.y > canvasHeight) { this.vY *= -1; }

        // 이동
        this.x += this.vX;
        this.y += this.vY;

        // 화면에 이동 표시
        this.body.style.left = this.x + 'px';
        this.body.style.top = this.y + 'px';
    };
</script>
```



## ❖ window.onload

- 객체를 생성해 객체 움직임
  - 배열을 만들고 MovingText 객체 생성해 배열에 넣음

코드 9-37 window.onload 부분

```
<script>
  window.onload = function () {
    // 변수를 선언합니다.
    var movingTexts = [];

    // 배열에 MovingText 객체 100개를 생성합니다.
    for (var i = 0; i < 100; i++) {
      movingTexts.push(new MovingText());
    }
  };
</script>
```



# 9.11 문서 객체와 객체지향을 이용한 움직임 구현

## ❖ 60 FPS 구현

코드 9-38 애니메이션

```
<!-- window.onload -->
<script>
    window.onload = function () {
        // 변수를 선언합니다.
        /* 생략 */
        // 배열에 MovingText 객체 100개를 생성합니다.
        /* 생략 */
        // 움직입니다.
        setInterval(function () {
            for (var i in movingTexts) { movingTexts[i].move(); }
        }, 1000 / 60);
    };
</script>
```





# 9.11 문서 객체와 객체지향을 이용한 움직임 구현

## ❖ 전체 코드

코드 9-39 전체 코드

```
<!DOCTYPE html>
<html>
<head>
  <!-- 보조 함수 -->
  <script>
    // 랜덤한 정수를 생성합니다.
    function nextRandomInteger(limit) {
      return Math.round(Math.random() * limit);
    }

    // 랜덤한 알파벳을 리턴하는 함수입니다.
    var alphabet = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
    function randomAlphabet() {
      return alphabet.charAt(nextRandomInteger(25));
    }

    // 양음으로 랜덤한 속도를 생성하는 함수입니다.
    function randomSpeed(maxSpeed) {
      return Math.random() * maxSpeed - Math.random() * maxSpeed;
    }
  </script>
  <!-- 생성자 함수 -->
  <script>
    // MovingText의 생성자 함수
    var canvasWidth = 700;
    var canvasHeight = 500;
```



## ❖ 전체 코드 (2)

```
function MovingText() {  
    // 위치와 속도 속성  
    this.x = nextRandomInteger(canvasWidth);  
    this.y = nextRandomInteger(canvasHeight);  
    this.vX = randomSpeed(10);  
    this.vY = randomSpeed(10);  
  
    // 문서 객체를 생성합니다.  
    this.body = document.createElement('h1');  
    this.body.innerHTML = randomAlphabet();  
    this.body.style.position = 'absolute';  
  
    // 문서 객체를 document.body에 추가합니다.  
    document.body.appendChild(this.body);  
}  
  
MovingText.prototype.move = function () {  
    // 범위 검사  
    if (this.x < 0 || this.x > canvasWidth) { this.vX *= -1; }  
    if (this.y < 0 || this.y > canvasHeight) { this.vY *= -1; }  
  
    // 이동  
    this.x += this.vX;  
    this.y += this.vY;  
  
    // 화면에 이동 표시  
    this.body.style.left = this.x + 'px';  
    this.body.style.top = this.y + 'px';  
};  
</script>
```



## ❖ 전체 코드 (3)

```
<!-- window.onload -->
<script>
    window.onload = function () {
        // 변수를 선언합니다.
        var movingTexts = [];

        // 배열에 MovingText 객체 100개를 생성합니다.
        for (var i = 0; i < 100; i++) {
            movingTexts.push(new MovingText());
        }

        // 움직입니다.
        setInterval(function () {
            for (var i in movingTexts) {
                movingTexts[i].move();
            }
        }, 1000 / 60);
    };
</script>
</head>
<body>

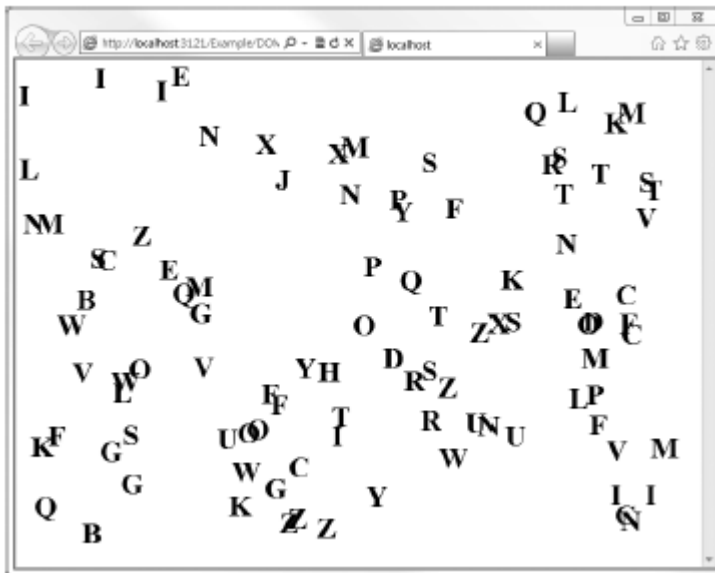
</body>
</html>
```



# 9.11 문서 객체와 객체지향을 이용한 움직임 구현

## ❖ 실행 결과

그림 9-19 이동하는 글자 예제





# Thank You !

모던 웹을 위한 Javascript jQuery 입문