

# 12 자바스크립트와 jQuery 라이브러리 응용

웹 프로그래밍 입문

# Contents

## ❖ 학습목표

- 자바스크립트의 타이머 함수를 알아봅니다.
- 자바스크립트와 jQuery 라이브러리의 이벤트 활용하는 방법을 알아봅니다.
- jQuery 플러그인을 사용하는 방법을 알아봅니다.
- jQuery 라이브러리를 사용해 갤러리를 만드는 방법을 알아봅니다.

## ❖ 내용

- 응용 예제 12-1 입력 양식 자동 초점 설정
- 응용 예제 12-2 프레임 애니메이션 만들기
- 응용 예제 12-3 문서 객체 생성
- 응용 예제 12-4 무한 스크롤 기능 구현
- 응용 예제 12-5 LightBox 플러그인 사용
- 응용 예제 12-6 Masonry 플러그인 사용
- 응용 예제 12-7 jQuery 라이브러리로 갤러리 구현



# 자바스크립트와 jQuery 라이브러리 응용

## ❖ 응용 예제 12-1 입력 양식 자동 초점 설정

- 주민등록번호를 입력할 때 앞자리를 입력하자마자 자동으로 뒷자리를 입력하는 칸으로 이동하는 프로그램

코드 12-1 입력 양식 자동 초점 설정

autofocus.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Event</title>
  <script>
    // 이벤트를 연결합니다.
    window.onload = function () {
      // 문서 객체를 선택합니다.
      var input_1 = document.querySelectorAll('input')[0];
      var input_2 = document.querySelectorAll('input')[1];

      // input_1
      input_1.onkeydown = function () {
        // 글자 개수가 여섯 개 이상일 때
        if (6 <= input_1.value.length) {
          // input_2 문서 객체로 초점을 이동합니다.
          input_2.focus();
        }
      };
    };
  </script>
</head>
</html>
```



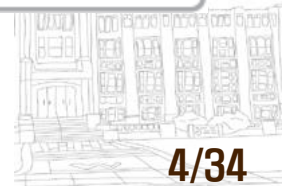
# 자바스크립트와 jQuery 라이브러리 응용

```
// input_2
input_2.onkeydown = function (event) {
    // 이벤트를 객체로 추출합니다.
    var event = event || window.event;

    // 사용자의 입력이 '백 스페이스'이고 입력된 글자가 없을 때
    if (event.keyCode == 8 && input_2.value.length == 0) {
        input_1.focus();
    }
};

</script>
</head>
<body>
    <input type="text" maxlength="6" />
    <span>-</span>
    <input type="text" maxlength="7" />
</body>
</html>
```

123456	-	78	×
--------	---	----	---



# 자바스크립트와 jQuery 라이브러리 응용

표 12-1 타이머 함수

함수	설명
setInterval(함수, 시간)	특정한 시간마다 함수 실행
setTimeout(함수, 시간)	특정한 시간 후에 함수 실행
clearInterval(식별번호)	setInterval() 함수로 설정한 타이머 제거
clearTimeout(식별번호)	setTimeout() 함수로 설정한 타이머 제거

- setInterval ( ) 함수 : 프레임 애니메이션

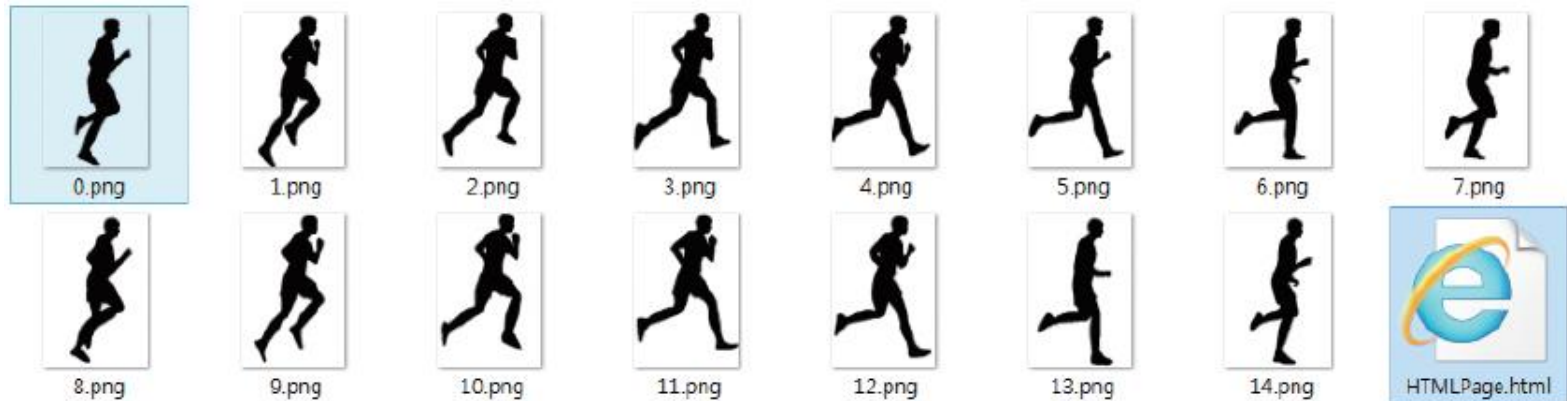
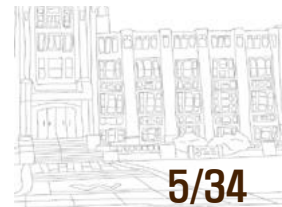


그림 12-1 폴더 구성



# 자바스크립트와 jQuery 라이브러리 응용

- 이미지가 1초에 20번 전환되며 사람이 달리는 애니메이션 만들기

코드 12-3 프레임 애니메이션

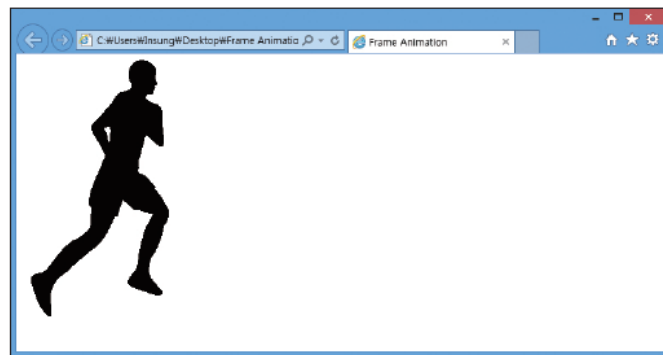
frameanimation.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Frame Animation</title>
  <script>
    window.onload = function () {
      // 변수를 선언합니다.
      var count = 0;
      var image = document.getElementById('image');
      var frames = [
        '0.png', '1.png', '2.png', '3.png', '4.png',
        '5.png', '6.png', '7.png', '8.png', '9.png',
        '10.png', '11.png', '12.png', '13.png', '14.png',
      ];
    };
  </script>
</head>
<body>
  <img alt="A person running" data-bbox="137 278 750 793"/>
</body>
</html>
```



# 자바스크립트와 jQuery 라이브러리 응용

```
// 타이머 반복을 시작합니다.  
setInterval(function () {  
    image.src = frames[count % frames.length];  
    count = count + 1;  
}, 1000 / 20);  
  
};  
</script>  
</head>  
<body>  
    <img id="image" />  
</body>  
</html>
```



실시간 급상승 검색어		
1	김아린	NEW
2	싸이 윤아	↑ 828
3	황정음	↑ 138
4	홍록기	↑ 954
5	한혜진	↑ 114
6	윤일봉	↑ 108
7	이유비	↑ 69
8	다니엘라 세만	NEW
9	송지호	↑ 84
10	파브레가스	↑ 435
더보기		

그림 12-2 setInterval() 함수 활용



# 자바스크립트와 jQuery 라이브러리 응용

## ❖ 응용 예제 12-3 문서 객체 생성

- jQuery 라이브러리를 사용해 문서 객체를 생성

코드 12-4 문서 객체 생성

jqueryCreate.html

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery Create</title>
  <script src="https://code.jquery.com/jquery-3.1.1.js"></script>
  <script>
    // 이벤트를 연결합니다.
    $(document).ready(function () {
      // 문서 객체를 생성합니다.
      $('<h1>Create Document Object By jQuery</h1>')
    });
  </script>
</head>
<body>

</body>
</html>
```





# 자바스크립트와 jQuery 라이브러리 응용

표 12-2 jQuery 문서 객체 추가 메소드(1)

메소드	설명
<code>\$(객체).prependTo(대상)</code>	객체를 대상의 앞부분에 추가
<code>\$(객체).appendTo(대상)</code>	객체를 대상의 뒷부분에 추가
<code>\$(객체).beforeTo(대상)</code>	객체를 대상의 앞쪽에 추가
<code>\$(객체).afterTo(대상)</code>	객체를 대상의 뒤쪽에 추가

- `appendTo( )` 메소드를 사용해 문서 객체를 `body` 태그에 추가

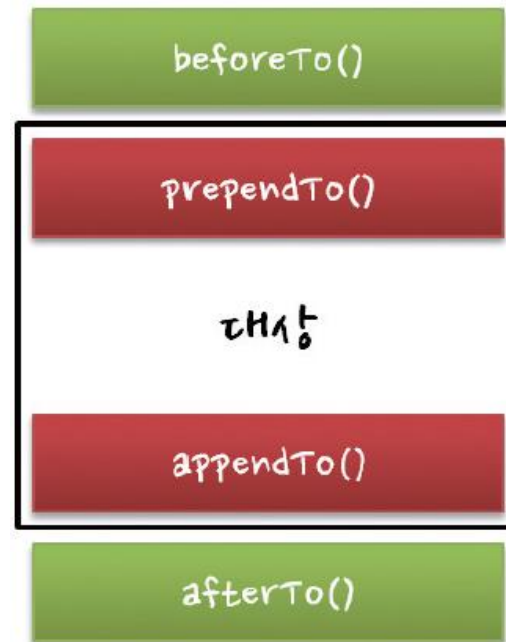


그림 12-3 문서 객체 추가 메소드

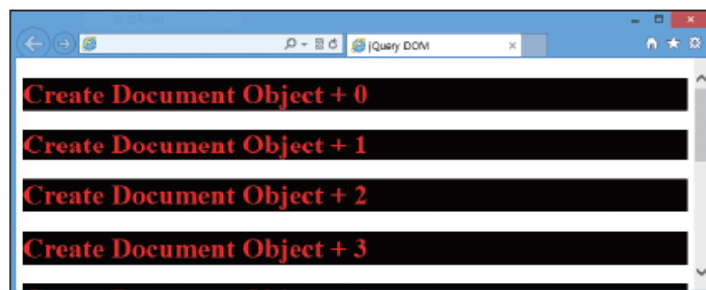


# 자바스크립트와 jQuery 라이브러리 응용

코드 12-5 문서 객체 추가(1)

jqueryCreateAppend\_1.html

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery DOM</title>
  <script src="https://code.jquery.com/jquery-3.1.1.js"></script>
  <script>
    // 이벤트를 연결합니다.
    $(document).ready(function () {
      // 10회 반복합니다.
      for (var i = 0; i < 10; i++) {
        // 문서 객체를 생성합니다.
        $('<h1>Create Document Object + ' + i + '</h1>').css({
          backgroundColor: 'black',
          color: 'red'
        }).appendTo('body');
      }
    });
  </script>
</head>
<body>
</body>
</html>
```



# 자바스크립트와 jQuery 라이브러리 응용

표 12-3 jQuery 문서 객체 추가 메소드(2)

메소드	설명
\$(대상).prepend(객체)	객체를 대상의 앞부분에 추가
\$(대상).append(객체)	객체를 대상의 뒷부분에 추가
\$(대상).before(객체)	객체를 대상의 앞쪽에 추가
\$(대상).after(객체)	객체를 대상의 뒤쪽에 추가

코드 12-6 문서 객체 추가(2)

jqueryCreateAppend\_2.html

```
<script>
• // 이벤트를 연결합니다.
$(document).ready(function () {
    // 10회 반복합니다.
    for (var i = 0; i < 10; i++) {
        // 문서 객체를 생성합니다.
        var $dynamic = $('<h1>Create Document Object + ' + i + '</h1>')
            .css({
                backgroundColor: 'black',
                color: 'red'
            });

        // 문서 객체를 추가합니다.
        $('body').append($dynamic);
    }
});
</script>
```



# 자바스크립트와 jQuery 라이브러리 응용

## ❖ 응용 예제 12-4 무한 스크롤 기능 구현

- 스크롤을 맨 아래쪽까지 내려도 무한 스크롤을 활용해 정보를 계속 제공

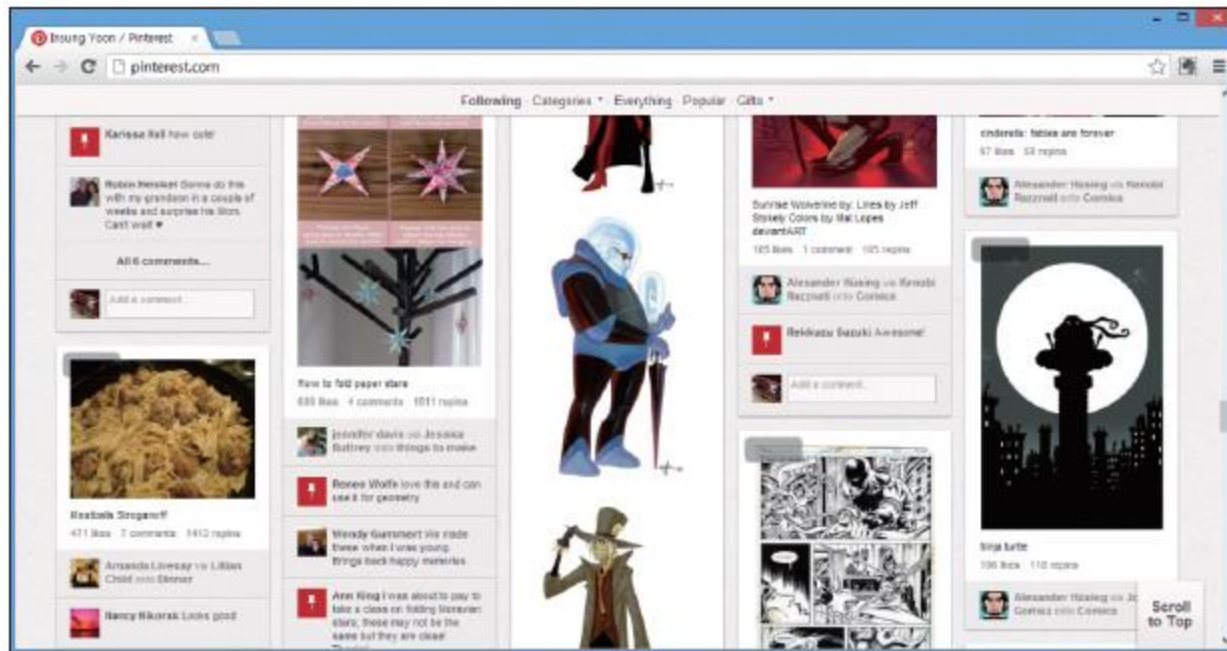


그림 12-4 무한 스크롤



# 자바스크립트와 jQuery 라이브러리 응용

- 전체 문서의 높이가 스크롤 높이와 웹 브라우저 창의 높이를 더한 값과 같은지 조건을 확인. 이 조건을 충족할 때만 데이터를 추가함



그림 12-5 무한 스크롤에 사용되는 변수



# 자바스크립트와 jQuery 라이브러리 응용

코드 12-7 무한 스크롤

infinityScroll.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Infinity Scroll</title>
  <script src="https://code.jquery.com/jquery-3.1.1.js"></script>
  <script>
    // 이벤트를 연결합니다.
    $(document).ready(function () {
      // 문서 객체 추가 함수
      var appendDocument = function () {
        for (var i = 0; i < 20; i++) {
          // 문서 객체를 생성합니다.
          $('<h1>Infinity Scroll</h1>').appendTo('body');
        }
      };
    });
  </script>
</head>
</html>
```



# 자바스크립트와 jQuery 라이브러리 응용

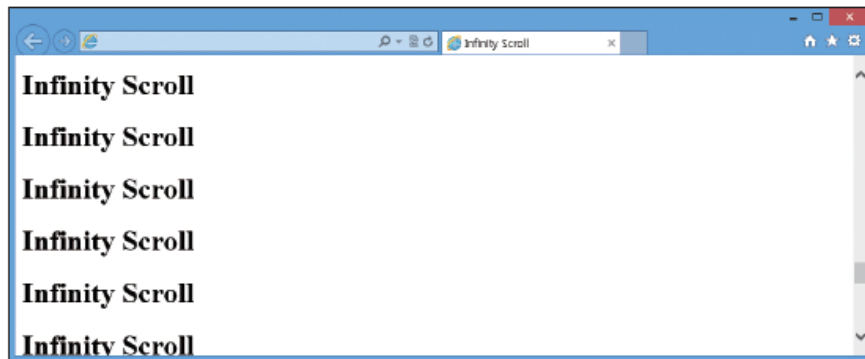
```
// 초기에 문서 객체를 추가합니다.
appendDocument();

// 이벤트를 연결합니다.
$(window).scroll(function () {
    // 변수를 선언합니다.
    var scrollHeight = $(window).scrollTop() + $(window).height();
    var documentHeight = $(document).height();

    // 검사합니다.
    if (scrollHeight == documentHeight) {
        appendDocument();
    }
});

</script>
</head>
<body>

</body>
</html>
```



# 자바스크립트와 jQuery 라이브러리 응용

## ❖ 응용 예제 12-5 LightBox 플러그인 사용

- 웹 페이지 내부에서 특정한 정보를 보여 주려고 배경은 어둡게 하고 정보는 밝게 보여 주는 플러그인



그림 12-6 라이트박스





# 자바스크립트와 jQuery 라이브러리 응용

- jQuery LightBox 플러그인

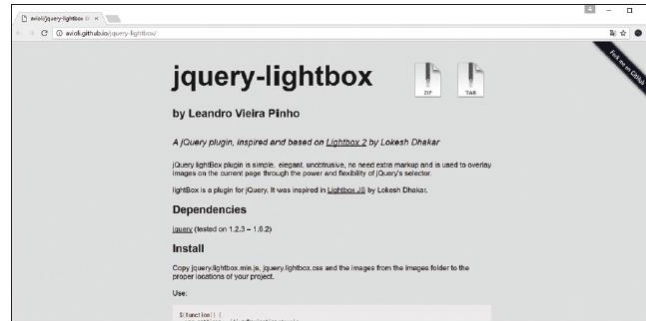


그림 12-7 jQuery LightBox 플러그인(<http://avioli.github.io/jquery-lightbox/>)

- 플러그인 압축 파일을 다운로드



그림 12-8 압축 파일 구성

- 폴더 구성

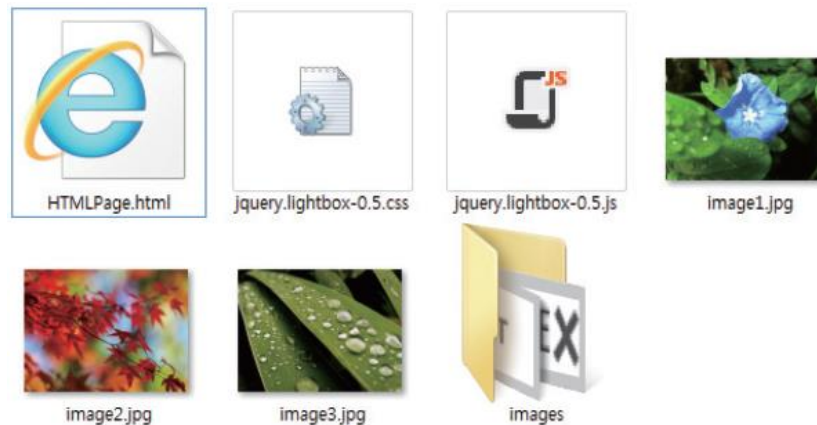


그림 12-9 폴더 구성



# 자바스크립트와 jQuery 라이브러리 응용

코드 12-8 jQuery LightBox 플러그인 설정

lightbox\_ready.html

```
<head>
  <title>LightBox</title>
  <link rel="stylesheet" href="jquery.lightbox-0.5.css" />
  <script src="https://code.jquery.com/jquery-3.1.1.js"></script>
  <script src="jquery.lightbox-0.5.js"></script>
</head>
```

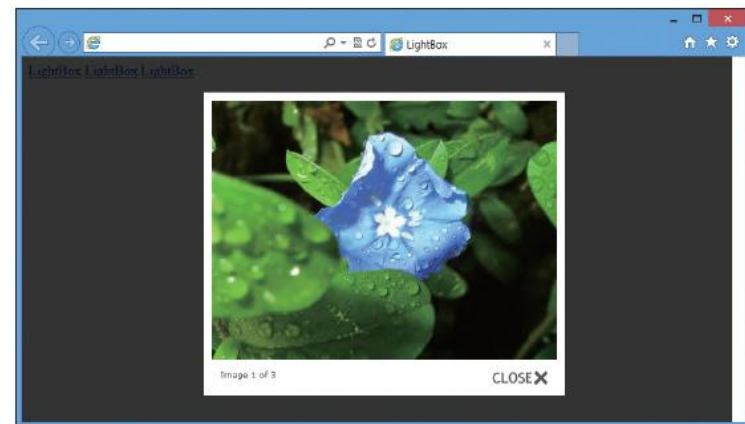


# 자바스크립트와 jQuery 라이브러리 응용

코드 12-9 jQuery LightBox 플러그인 사용

lightbox.html

```
<!DOCTYPE html>
<html>
<head>
  <title>LightBox</title>
  <link rel="stylesheet" href="jquery.lightbox-0.5.css" />
  <script src="https://code.jquery.com/jquery-3.1.1.js"></script>
  <script src="jquery.lightbox-0.5.js"></script>
  <script>
    // 이벤트를 연결합니다.
    $(document).ready(function () {
      $('a.light').lightbox();
    });
  </script>
</head>
<body>
  <a class="light" href="image1.jpg">LightBox</a>
  <a class="light" href="image2.jpg">LightBox</a>
  <a class="light" href="image3.jpg">LightBox</a>
</body>
</html>
```



# 자바스크립트와 jQuery 라이브러리 응용

## ❖ 응용 예제 12-6 Masonry 플러그인 사용

- 타일형 웹 페이지를 만들 때 사용

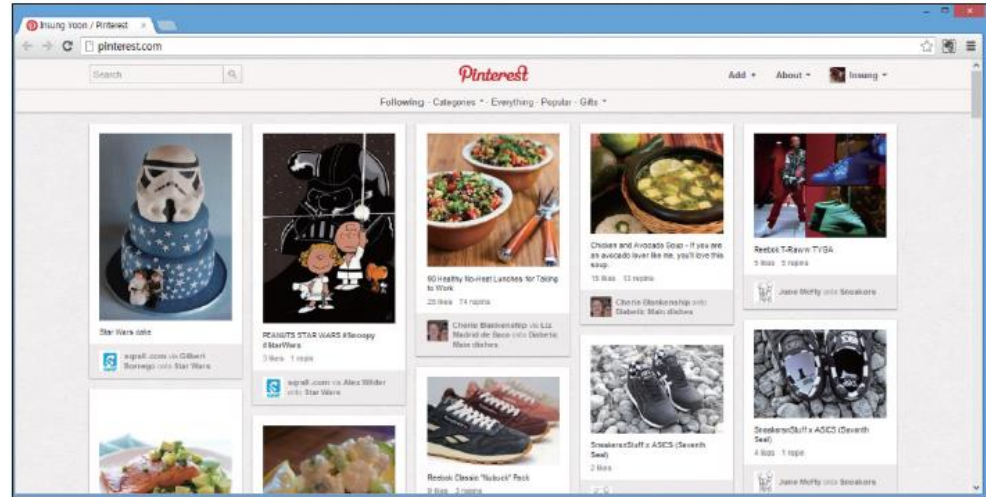


그림 12-10 핀터레스트(<http://pinterest.com/>)

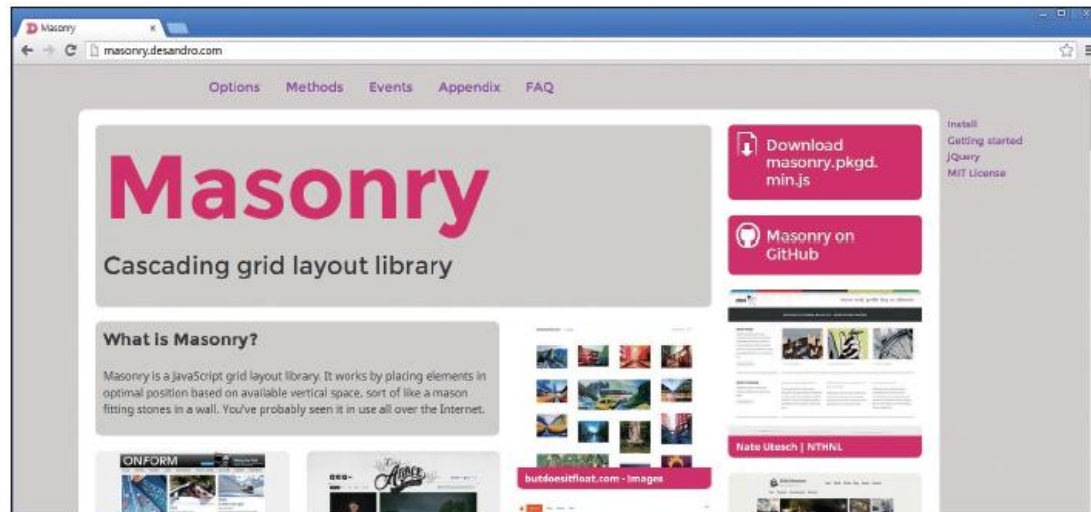


그림 12-11 Masonry 플러그인(<http://masonry.desandro.com/>)



# 자바스크립트와 jQuery 라이브러리 응용

- Masonry 플러그인의 CDN을 사용

코드 12-10 플러그인

jqueryMasonry.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Masonry</title>
  <style>
    * { margin: 0; padding: 0; }
    .box {
      background: black;
      margin: 5px;
    }
  </style>
  <script src="https://code.jquery.com/jquery-3.1.1.js"></script>
  <script src="http://masonry.desandro.com/masonry.pkgd.min.js"></script>
  <script>
```

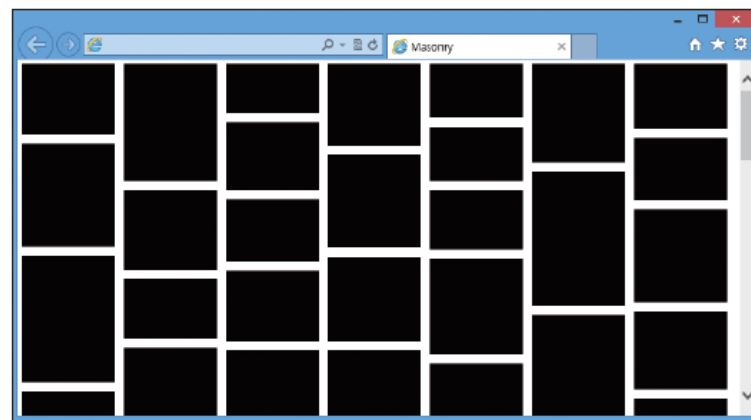


# 자바스크립트와 jQuery 라이브러리 응용

```
// 이벤트를 연결합니다.
$(document).ready(function () {
    // 문서 객체를 생성합니다.
    for (var i = 0; i < 100; i++) {
        $('<div></div>').addClass('box').css({
            width: 100,
            height: Math.floor(Math.random() * 100) + 50
        }).appendTo('body');
    }

    // Masonry 플러그인을 적용합니다.
    $('body').masonry({
        columnWidth: 110
    });
});
</script>
</head>
<body>

</body>
</html>
```



# 자바스크립트와 jQuery 라이브러리 응용

## ❖ 응용 예제 12-7 jQuery 라이브러리로 갤러리 구현

- 1 HTML 페이지 만들기 : ready 이벤트를 2개로 나눔

코드 12-11 기본 HTML 페이지 생성

gallery\_ready.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Slider</title>
  <style>

  </style>
  <script src="https://code.jquery.com/jquery-3.1.1.js"></script>
  <script>
    // 첫 번째 ready 이벤트
    $(document).ready(function () {

    });

    // 두 번째 ready 이벤트
    $(document).ready(function () {
    });
  </script>
</head>
<body>

</body>
</html>
```



# 자바스크립트와 jQuery 라이브러리 응용

- 2 body 태그 구성하기 : body 태그 내부에 다음 코드를 작성

코드 12-12 태그 구성

gallery\_body.html

```
<body>
  <div data-role="slider" data-width="500" data-height="300">
    <div class="container">
      <div class="item">
        <h1>Lorem ipsum dolor sit amet</h1>
        <p>Lorem ipsum dolor sit amet, consectetur </p>
      </div>
      <div class="item">
        <h1>Proin in urna turpis.</h1>
        <p>Lorem ipsum dolor sit amet, consectetur </p>
      </div>
      <div class="item">
        <h1>Duis malesuada lorem neque.</h1>
        <p>Lorem ipsum dolor sit amet, consectetur </p>
      </div>
    </div>
  </div>
  <button id="left-button">←</button>
  <button id="right-button">→</button>
</body>
```

‘←’을 입력하고 [한자]를 누르면  
특수 문자를 입력할 수 있습니다.





# 자바스크립트와 jQuery 라이브러리 응용

- 3 갤러리 구분 색상 적용

코드 12-13 갤러리 구분 색상 적용

gallery\_style.html

```
<style>
  div.item:nth-child(1) { background: blueviolet; }
  div.item:nth-child(2) { background: pink; }
  div.item:nth-child(3) { background-color: burlywood; }
</style>
```

- 4 갤러리 스타일 적용

코드 12-14 갤러리 스타일 적용

gallery\_css.html

```
// 첫 번째 ready 이벤트
$(document).ready(function () {
  // 변수를 선언합니다.
  var width = $('[data-role="slider"]').attr('data-width');
  var height = $('[data-role="slider"]').attr('data-height');
  var count = $('[data-role="slider"] div.item').length;

  // 스타일을 적용합니다.
  $('[data-role="slider"]').css({
    position: 'relative',
    overflow: 'hidden',
```



# 자바스크립트와 jQuery 라이브러리 응용

```
        width: width,  
        height: height  
    }).find('.container').css({  
        position: 'absolute',  
        width: count * width,  
        overflow: 'hidden'  
    }).find('.item').css({  
        width: width,  
        height: height,  
        float: 'left'  
    });  
});
```



# 자바스크립트와 jQuery 라이브러리 응용

- 5 갤러리 이벤트 연결 : changePage( ) 함수는 div.container 태그를 이동시킴

코드 12-15 갤러리 이벤트 연결

gallery\_event.html

```
// 두 번째 ready 이벤트
$(document).ready(function () {
    // 변수를 선언합니다.
    var currentPage = 0;
    var changePage = function () {
        $('[data-role="slider"] > .container').animate({
            left: -currentPage * width
        }, 500);
    };

    // 이벤트를 연결합니다.
    $('#left-button').click(function () { });
    $('#right-button').click(function () { });
});
```



# 자바스크립트와 jQuery 라이브러리 응용

- 6 갤러리 클릭 이벤트 구현 : 맨 왼쪽과 오른쪽(시작과 끝)에서는 페이지를 이동하지 않게 만들어줌

코드 12-16 갤러리 클릭 이벤트 구현

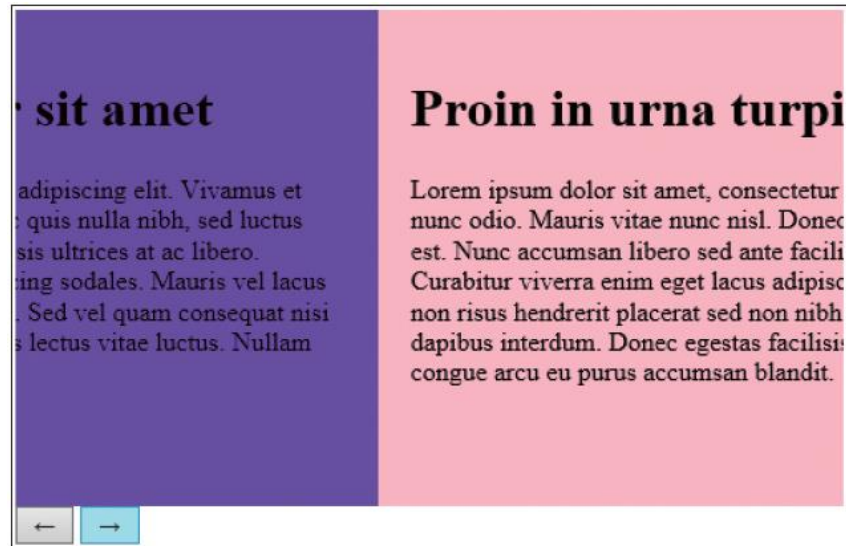
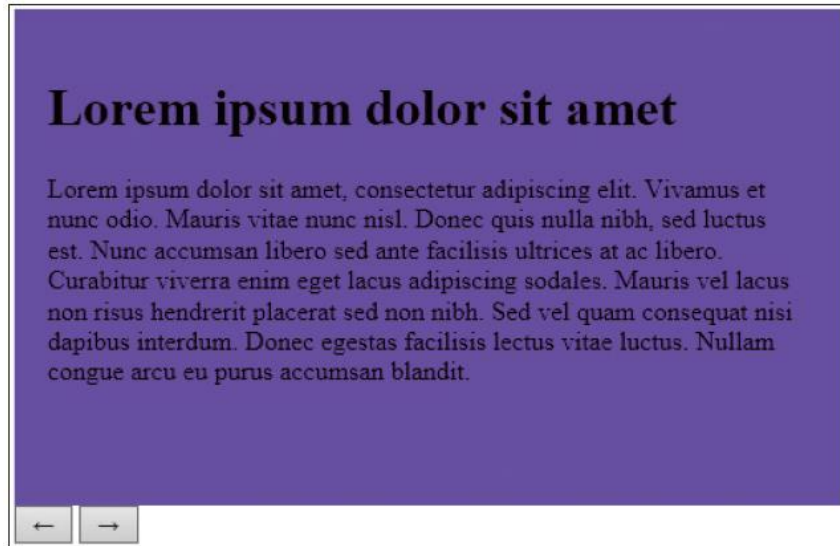
gallery\_complete.html

```
// 이벤트를 연결합니다.
$('#left-button').click(function () {
    if (currentPage > 0) {
        // 왼쪽으로 이동
        currentPage = currentPage - 1;
        changePage();
    }
});

$('#right-button').click(function () {
    if (currentPage < count - 1) {
        // 오른쪽으로 이동
        currentPage = currentPage + 1;
        changePage();
    }
});
```

# 자바스크립트와 jQuery 라이브러리 응용

- 7 실행 : 코드를 실행한 후 버튼을 클릭하면 갤러리가 자연스럽게 움직임



# 자바스크립트와 jQuery 라이브러리 응용

- 8 전체 코드 확인

코드 12-17 갤러리 전체 코드

gallery.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Slider</title>
  <style>
    div.item:nth-child(1) { background: blueviolet; }
    div.item:nth-child(2) { background: pink; }
    div.item:nth-child(3) { background-color: burlywood; }
  </style>
  <script src="https://code.jquery.com/jquery-3.1.1.js"></script>
  <script>
    $(document).ready(function () {
      // 변수를 선언합니다.
      var width = $('[data-role="slider"]').attr('data-width');
      var height = $('[data-role="slider"]').attr('data-height');
      var count = $('[data-role="slider"] div.item').length;
```



# 자바스크립트와 jQuery 라이브러리 응용

```
// 스타일을 적용합니다.
$('[data-role="slider"]').css({
    position: 'relative',
    overflow: 'hidden',
    width: width,
    height: height
}).find('.container').css({
    position: 'absolute',
    width: count * width,
    overflow: 'hidden'
}).find('.item').css({
    width: width,
    height: height,
    float: 'left'
});

// 변수를 선언합니다.
var currentPage = 0;
var changePage = function () {
    $('[data-role="slider"] > .container').animate({
        left: -currentPage * width
    }, 500);
};
```



# 자바스크립트와 jQuery 라이브러리 응용

```
// 이벤트를 연결합니다.
$('#left-button').click(function () {
    if (currentPage > 0) {
        // 왼쪽으로 이동
        currentPage = currentPage - 1;
        changePage();
    }
});

$('#right-button').click(function () {
    if (currentPage < count - 1) {
        // 오른쪽으로 이동
        currentPage = currentPage + 1;
        changePage();
    }
});
});
</script>
</head>
```





# 자바스크립트와 jQuery 라이브러리 응용

```
<body>
  <div data-role="slider" data-width="500" data-height="300">
    <div class="container">
      <div class="item">
        <h1>Lorem ipsum dolor sit amet</h1>
        <p>Lorem ipsum dolor sit amet, consectetur </p>
      </div>
      <div class="item">
        <h1>Proin in urna turpis.</h1>
        <p>Lorem ipsum dolor sit amet, consectetur </p>
      </div>
      <div class="item">
        <h1>Duis malesuada lorem neque.</h1>
        <p>Lorem ipsum dolor sit amet, consectetur </p>
      </div>
    </div>
  </div>
  <button id="left-button">←</button>
  <button id="right-button">→</button>
</body>
</html>
```





Thank You

---