Chapter 12

자바스크립트와 jQuery 라이브러리 응용

목차

학습목표

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

Section 01
입력 양식 포커스

01. 입력 양식 포커스

그림 12-1 Yes24 배송 주소 입력 화면

01. 입력 양식 포커스

응용 예제 12-1 입력 양식 포커스 자동 설정

코드 12-1 autofocus.html

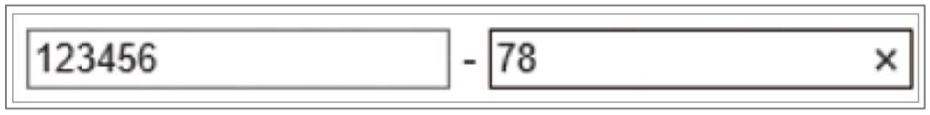
```
<!DOCTYPE html>
<html>
<head>
    <title>Event</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 문서 객체를 선택합니다.
            let input_1 = document.querySelectorAll('input')[0];
            let input_2 = document.querySelectorAll('input')[1];

            // input_1
            input_1.onkeydown = function () {
                // 글자 개수가 6개 이상일 때
                if (6 <= input_1.value.length) {
                    // input_2 문서 객체로 포커스를 이동합니다.
                    input_2.focus();
                }
            };
```

01. 입력 양식 포커스

응용 예제 12-1 입력 양식 포커스 자동 설정

```
            // input_2
            input_2.onkeydown = function (event) {
                // 이벤트 객체를 추출합니다.
                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>
```



01. 입력 양식 포커스

NOTE keyCode 확인

코드 12-2 note_keyCode.html

```
window.onload = function () {
     window.onkeydown = function (event) {
          alert(event.keyCode);
     };
};
```

Section 02
프레임 애니메이션

02. 프레임 애니메이션

- 타이머 함수는 특정 시간마다 코드를 실행.

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

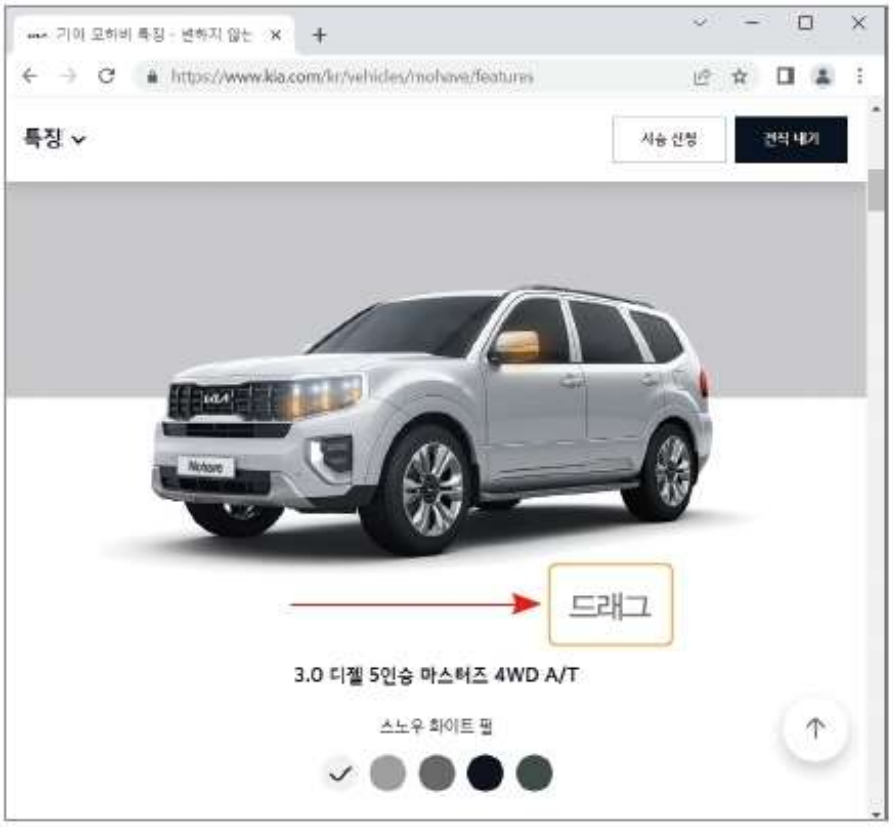
- 실시간 검색어는 일정 시간마다

서버에 데이터를 요청해 사용자에게 보여 주는데,

여기에 setInterval( ) 함수가 사용됨.

그림 12-2 setInterval( ) 함수 활용

02. 프레임 애니메이션



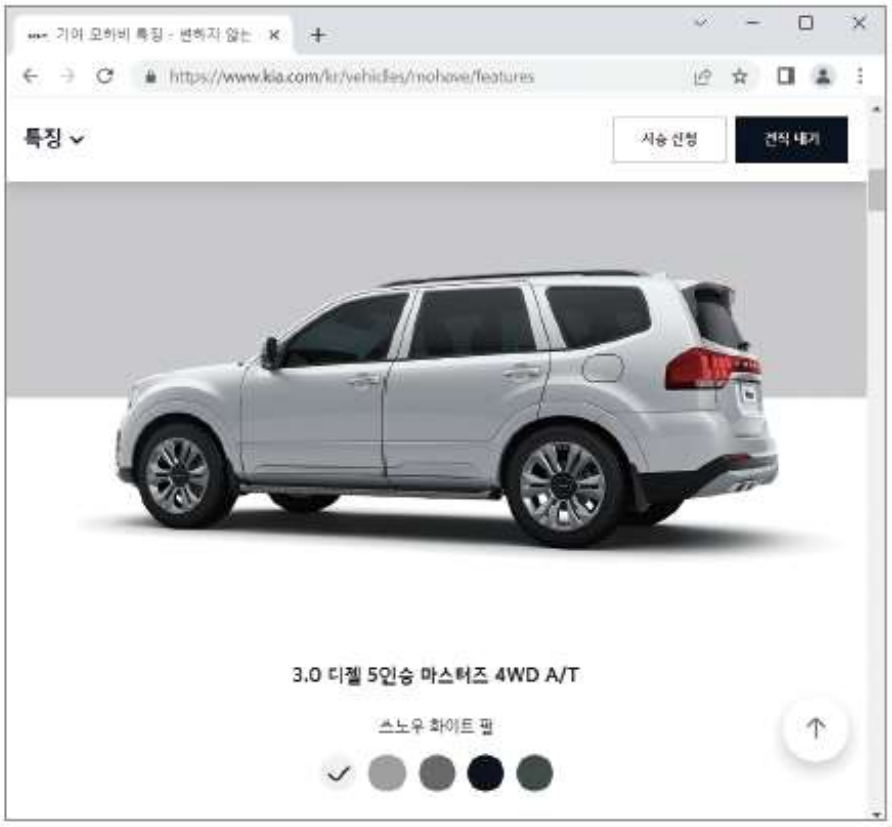


그림 12-3 기아자동차 웹 사이트의 마우스를 활용한 프레임 애니메이션

02. 프레임 애니메이션

응용 예제 12-2 프레임 애니메이션

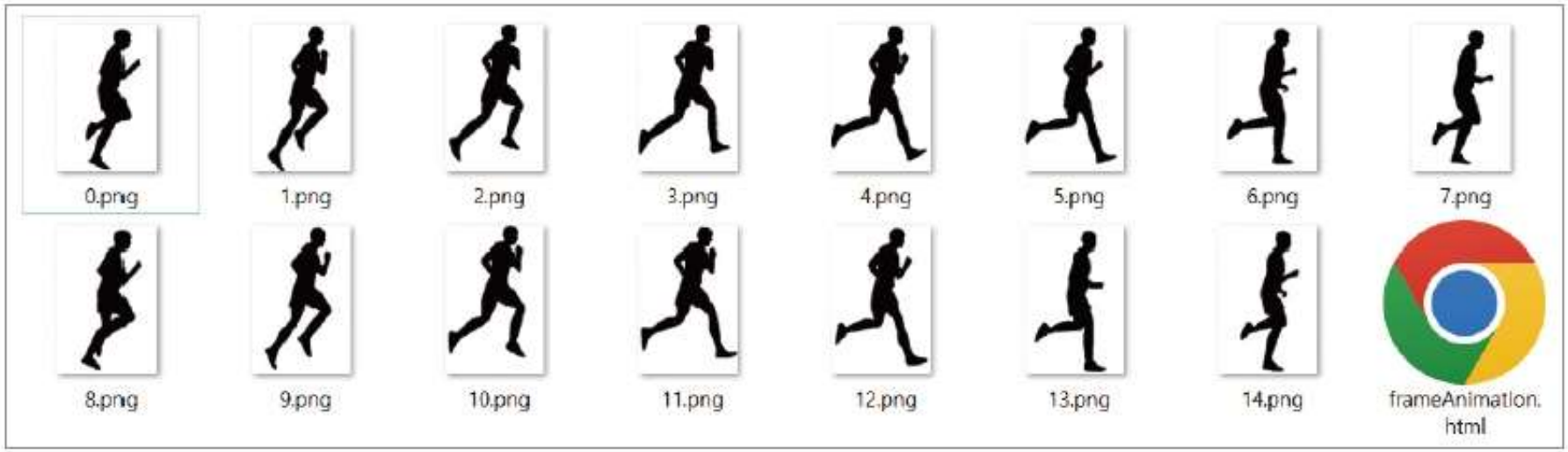


그림 12-4 프레임 애니메이션 폴더 구성

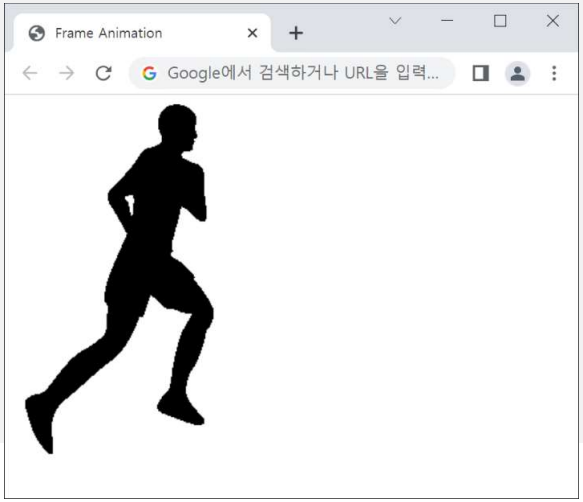
코드 12-3 frameAnimation.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Frame Animation</title>
    <script>
        window.onload = function () {
```

02. 프레임 애니메이션

응용 예제 12-2 프레임 애니메이션

```
            // 변수를 선언합니다.
            let count = 0;
            let image = document.getElementById('image');
            let 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'
            ];
            // 타이머 반복을 시작합니다.
            setInterval(function () {
                image.src = frames[count % frames.length];
                count = count + 1;
            }, 1000 / 20);
        };
    </script>
</head>
<body>
    <img id="image">
</body>
</html>
```



Section 03
문서 객체 생성과 추가

03. 문서 객체 생성과 추가

표 12-2 jQuery 문서 객체 추가 메서드

메서드	설명
$(객체).prependTo(대상)	객체를 대상의 앞부분에 추가
$(객체).appendTo(대상)	객체를 대상의 뒷부분에 추가
$(객체).beforeTo(대상)	객체를 대상의 앞쪽에 추가
$(객체).afterTo(대상)	객체를 대상의 뒤쪽에 추가

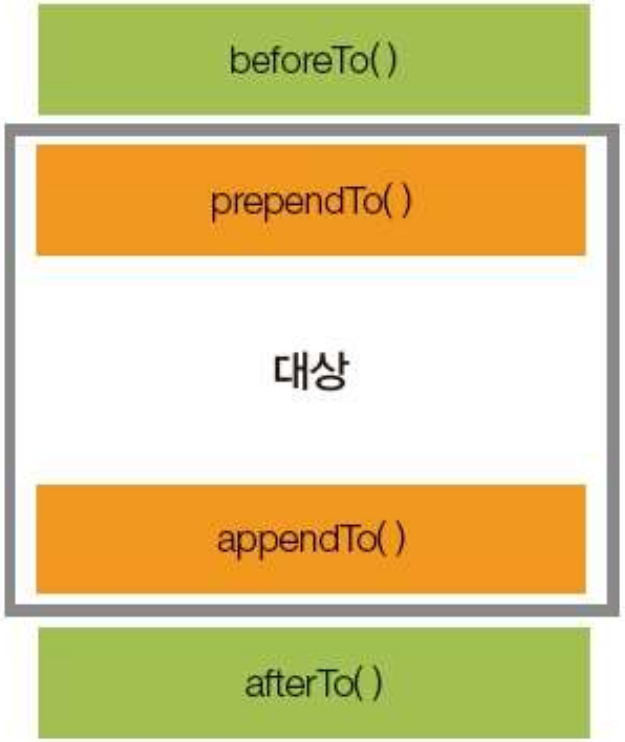


표 12-3 기타 jQuery 문서 객체 추가 메서드

메서드	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

03. 문서 객체 생성과 추가

응용 예제 12-3 문서 객체 생성

1 문서 객체 생성하기

코드 12-4 jqueryCreate.html

```
<!DOCTYPE html>
<html>
<head>
    <title>jQuery Create</title>
    <script src="https://code.jquery.com/jquery-3.6.4.js"></script>
    <script>
        // 이벤트를 연결합니다.
        $(document).ready(function () {
            // 문서 객체를 생성합니다.
            $('<h1>Create Document Object By jQuery</h1>')
        });
    </script>
</head>
<body>

</body>
</html>
```

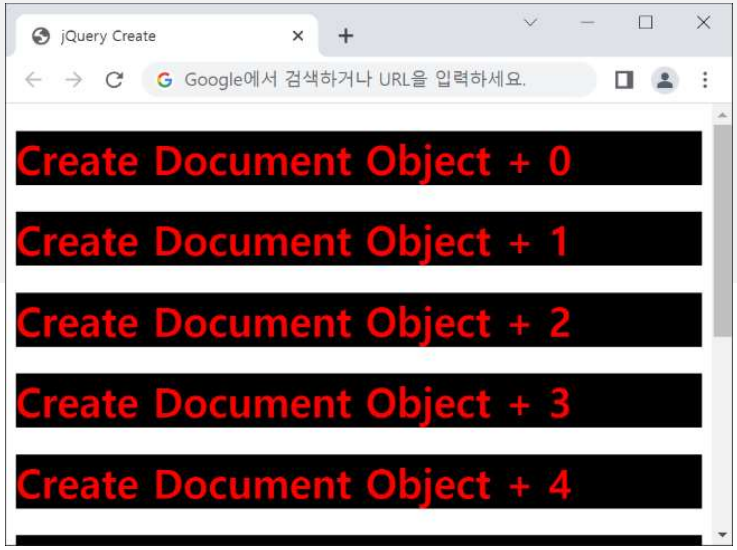
03. 문서 객체 생성과 추가

응용 예제 12-3 문서 객체 생성

2 appendTo( ) 메서드를 사용해 문서 객체 추가하기

코드 12-5 jqueryCreateAppend_1.html

```
<script src="https://code.jquery.com/jquery-3.6.4.js"></script>
<script>
    // 이벤트를 연결합니다.
    $(document).ready(function () {
        // 10회 반복합니다.
        for (let i = 0; i < 10; i++) {
            // 문서 객체를 생성합니다.
            $('<h1>Create Document Object + ' + i + '</h1>').css({
                backgroundColor: 'black',
                color: 'red'
            }).appendTo('body');
        }
    });
</script>
```



03. 문서 객체 생성과 추가

응용 예제 12-3 문서 객체 생성

3 append( ) 메서드를 사용해 문서 객체 추가하기

코드 12-6 jqueryCreateAppend_2.html

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

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

Section 04
무한 스크롤

04. 무한 스크롤

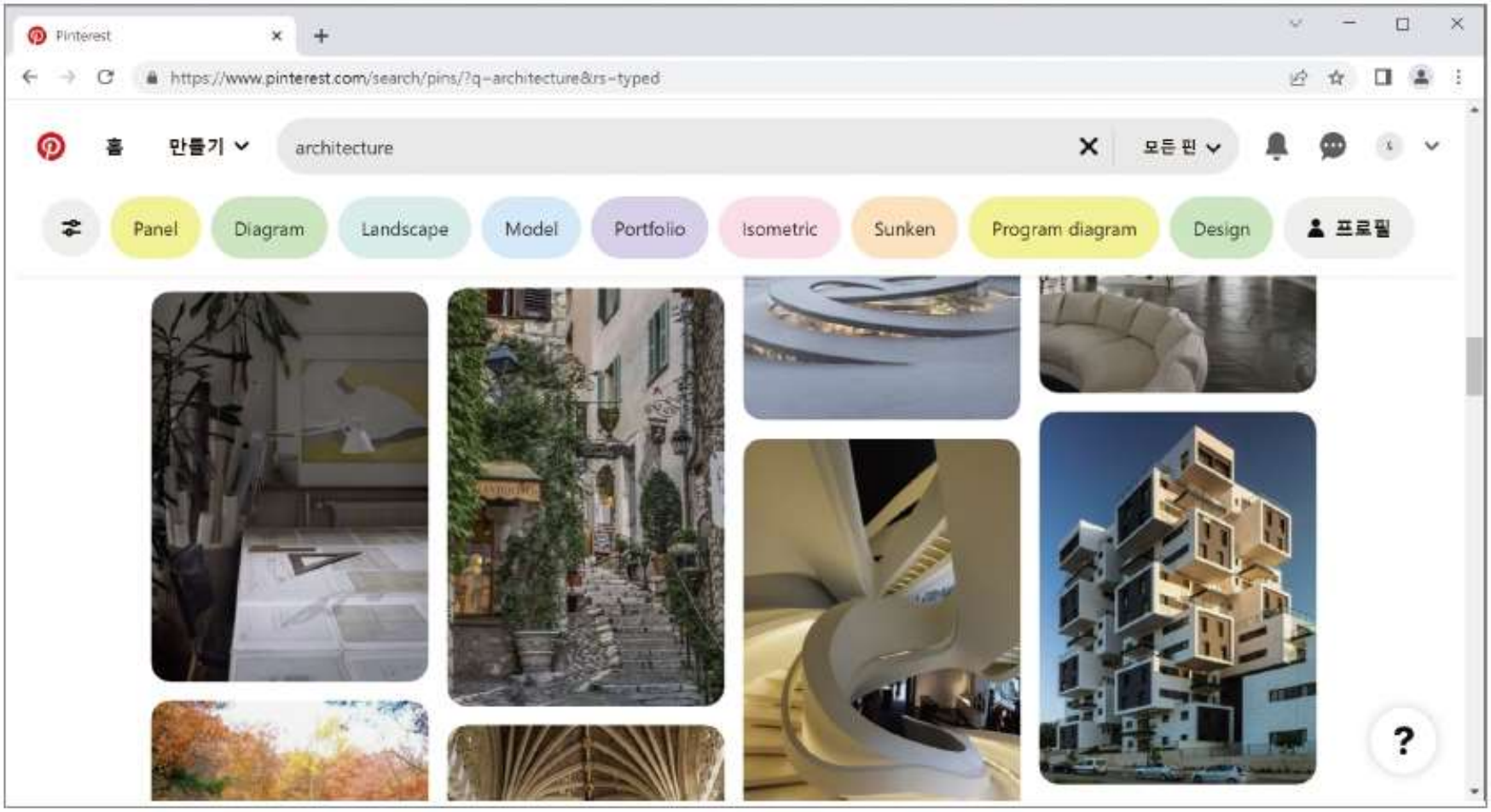


그림 12-5 무한 스크롤

04. 무한 스크롤

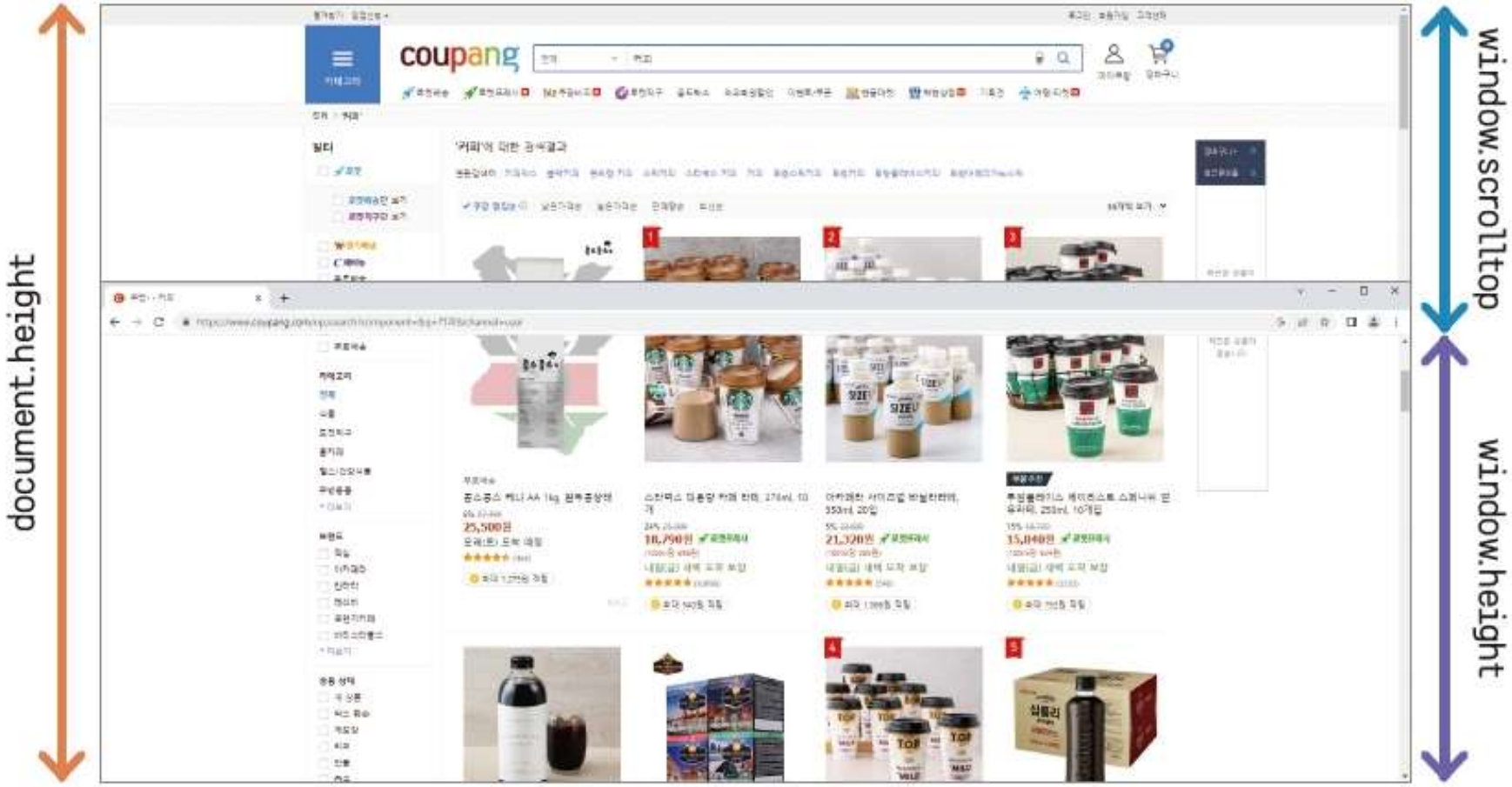


그림 12-6 무한 스크롤에 사용되는 변수

04. 무한 스크롤

응용 예제 12-4 무한 스크롤

코드 12-7 infinityScroll.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Infinity Scroll</title>
    <script src="https://code.jquery.com/jquery-3.6.4.js"></script>
    <script>
        // 이벤트를 연결합니다.
        $(document).ready(function () {
            // 문서 객체 추가 함수
            let appendDocument = function () {
                for (var i = 0; i < 20; i++) {
                    // 문서 객체를 생성합니다.
                    $('<h1>Infinity Scroll</h1>').appendTo('body');
                }
            };
            // 초기에 문서 객체를 추가합니다.
            appendDocument();
```

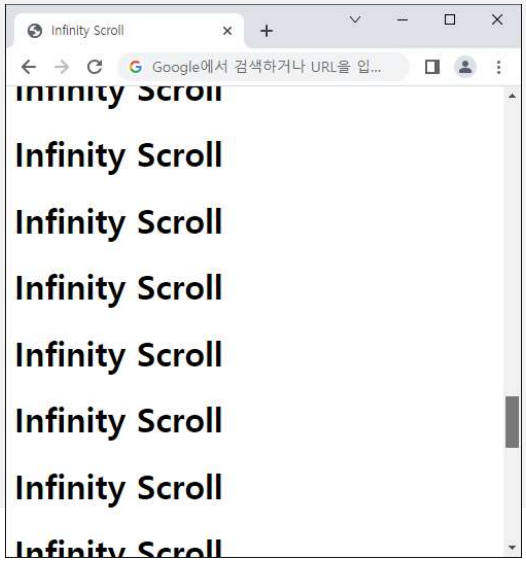
04. 무한 스크롤

응용 예제 12-4 무한 스크롤

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

                // 검사합니다.
                if (scrollHeight == documentHeight) {
                appendDocument();
                }
            });
        });
    </script>
</head>

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



Section 05
플러그인 사용

05. 플러그인 사용

1. 라이트박스

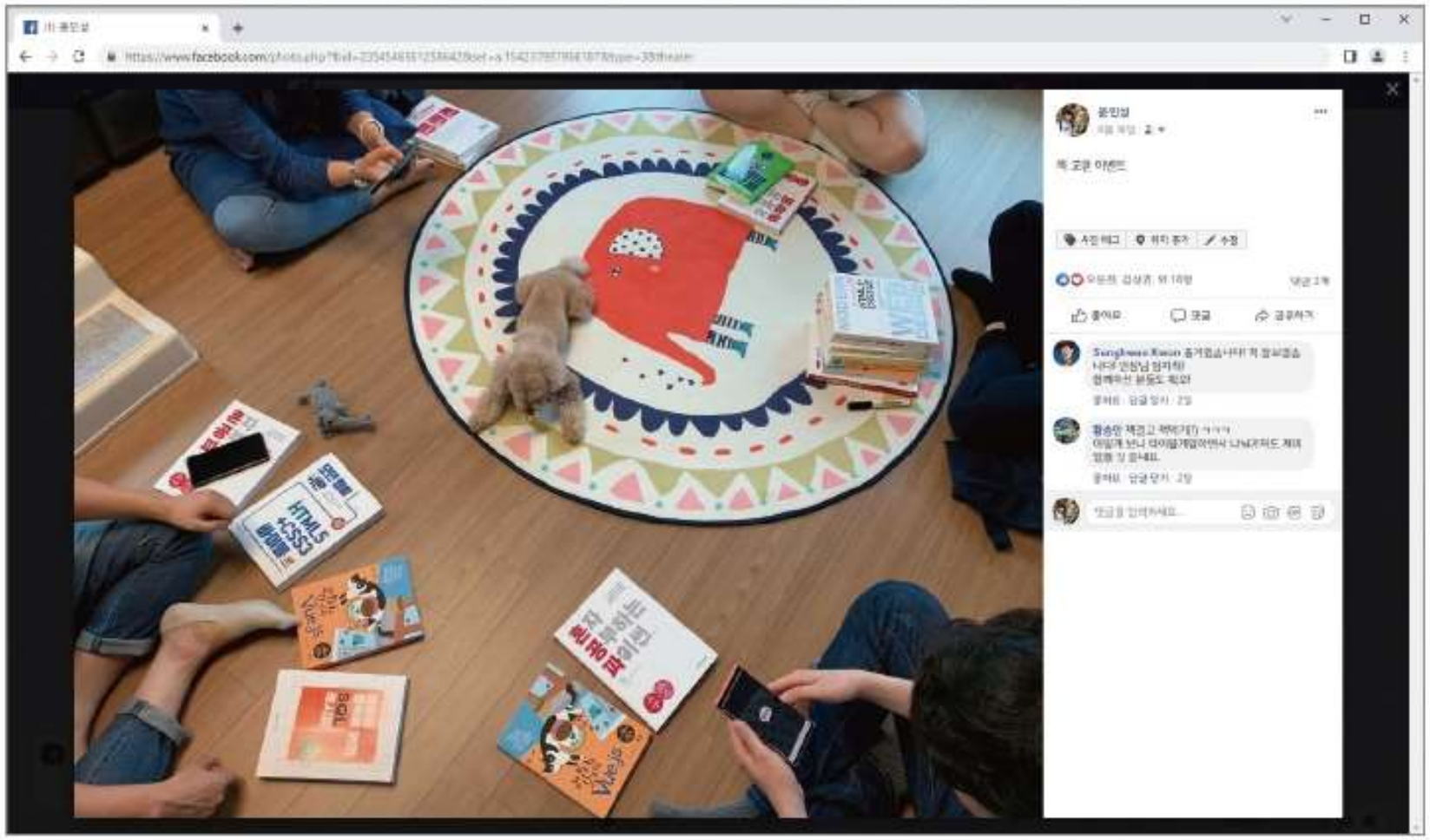


그림 12-7 라이트박스

05. 플러그인 사용

1. 라이트박스

응용 예제 12-5 LightBox 플러그인 사용

- 플러그인 압축 파일을 다운로드.

(a) jQuery LightBox 플러그인 압축 파일 구성

(b) 시작 폴더

그림 12-9 라이트박스 시작 폴더 구성

05. 플러그인 사용

1. 라이트박스

코드 12-8 lightbox_ready.html

```
<head>
<title>LightBox</title>
    <link rel="stylesheet" href="jquery.lightbox.css">
    <script src="https://code.jquery.com/jquery-3.6.4.js"></script>
    <script src="jquery.lightbox.js"></script>
</head>
```

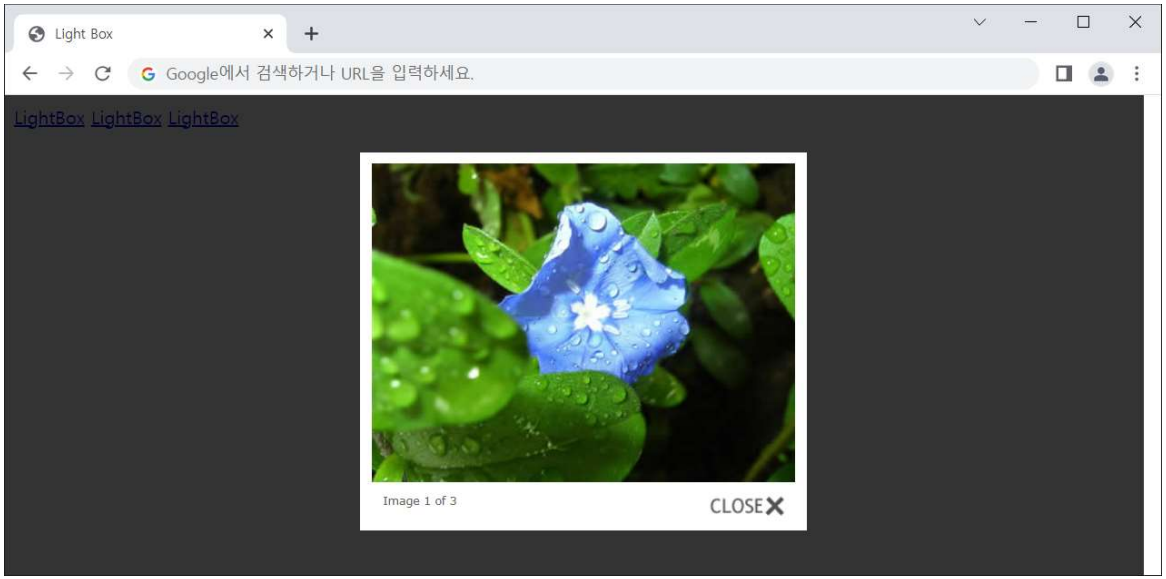
코드 12-9 lightbox.html

```
<!DOCTYPE html>
<html>
<head>
    <title>LightBox</title>
    <link rel="stylesheet" href="jquery.lightbox.css">
    <script src="https://code.jquery.com/jquery-3.6.4.js"></script>
    <script src="jquery.lightbox.js"></script>
    <script>
        // 이벤트를 연결합니다.
        $(document).ready(function () {
                $('a.light').lightBox();
        });
    </script>
```

05. 플러그인 사용

1. 라이트박스

```
</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>
```



05. 플러그인 사용

2. 카드 타일

- Masonry 플러그인

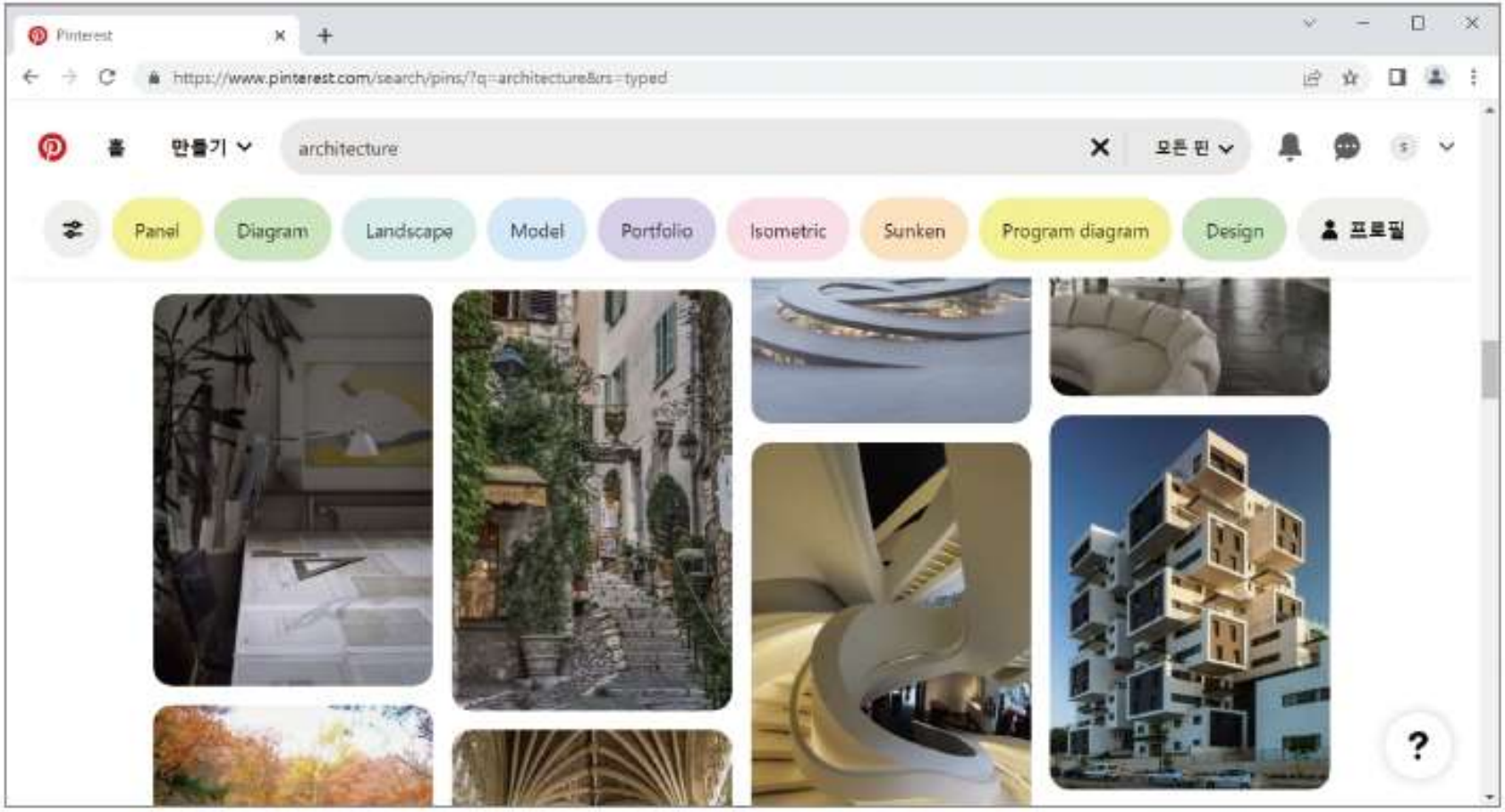


그림 12-10 핀터레스트(http://www.pinterest.co.kr)

05. 플러그인 사용

2. 카드 타일

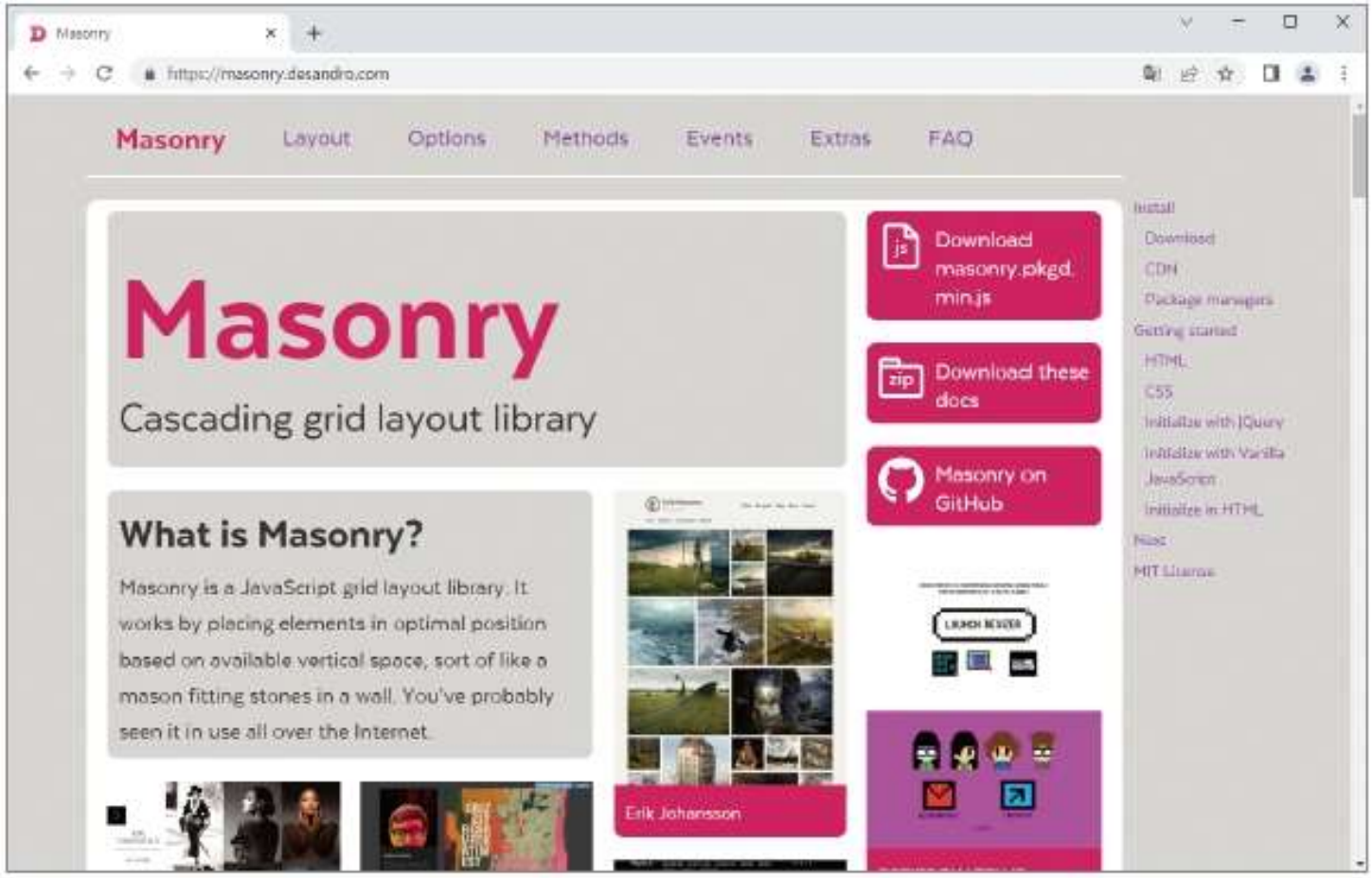


그림 12-11 Masonry 플러그인(http://masonry.desandro.com)

05. 플러그인 사용

2. 카드 타일

응용 예제 12-6 Masonry 플러그인 사용

코드 12-10 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-3.6.4.js"></script>
    <script src= "https://masonry.desandro.com/masonry.pkgd.min.js"></script>
    <script>
        // 이벤트를 연결합니다.
        $(document).ready(function () {
```

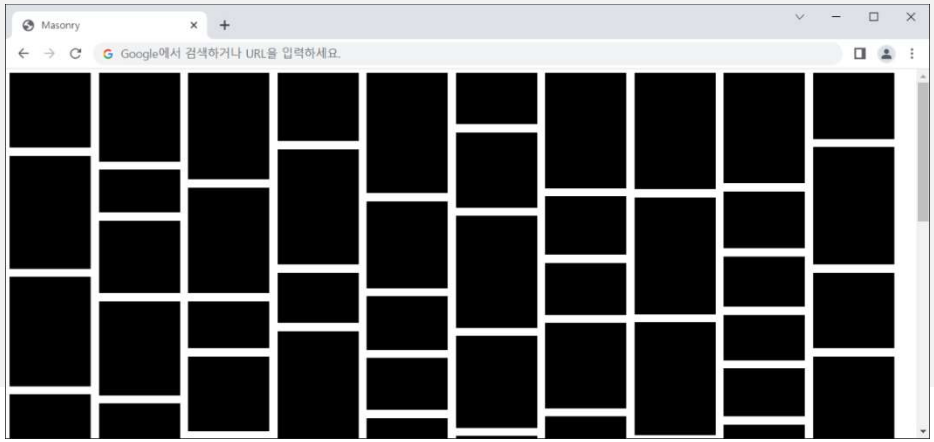
05. 플러그인 사용

2. 카드 타일

응용 예제 12-6 Masonry 플러그인 사용

```
            // 문서 객체를 생성합니다.
            for (let 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>
```



Section 06
슬라이더

06. 슬라이더

그림 12-12 삼성 갤럭시 홍보 페이지의 거대한 슬라이더 갤러리(http://www.samsung.com/global/galaxy)

06. 슬라이더

응용 예제 12-7 jQuery 라이브러리를 사용한 슬라이더 갤러리 구현

1 HTML 페이지 만들기

코드 12-11 gallery_ready.html

```
<script src="https://code.jquery.com/jquery-3.6.4.js"></script>
<script>
    // 첫 번째 ready 이벤트
    $(document).ready(function () {
    });

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

06. 슬라이더

응용 예제 12-7 jQuery 라이브러리를 사용한 슬라이더 갤러리 구현

2 body 태그 구성하기

코드 12-12 body 태그

```
<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>
     <button id="left-button">←</button>
     <button id="right-button">→</button>
</body>
```

06. 슬라이더

응용 예제 12-7 jQuery 라이브러리를 사용한 슬라이더 갤러리 구현

3 갤러리 구분 색상 적용하기

코드 12-13 div.item 태그에 스타일 적용

```
<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 적용하기

코드 12-14 첫 번째 ready 이벤트에 스타일 적용

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

06. 슬라이더

응용 예제 12-7 jQuery 라이브러리를 사용한 슬라이더 갤러리 구현

```
    // 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'
    });
});
```

06. 슬라이더

응용 예제 12-7 jQuery 라이브러리를 사용한 슬라이더 갤러리 구현

5 갤러리 이벤트 연결하기

코드 12-15 두 번째 ready 이벤트

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

06. 슬라이더

응용 예제 12-7 jQuery 라이브러리를 사용한 슬라이더 갤러리 구현

6 갤러리 클릭 이벤트 구현하기

코드 12-16 클릭 이벤트 구현

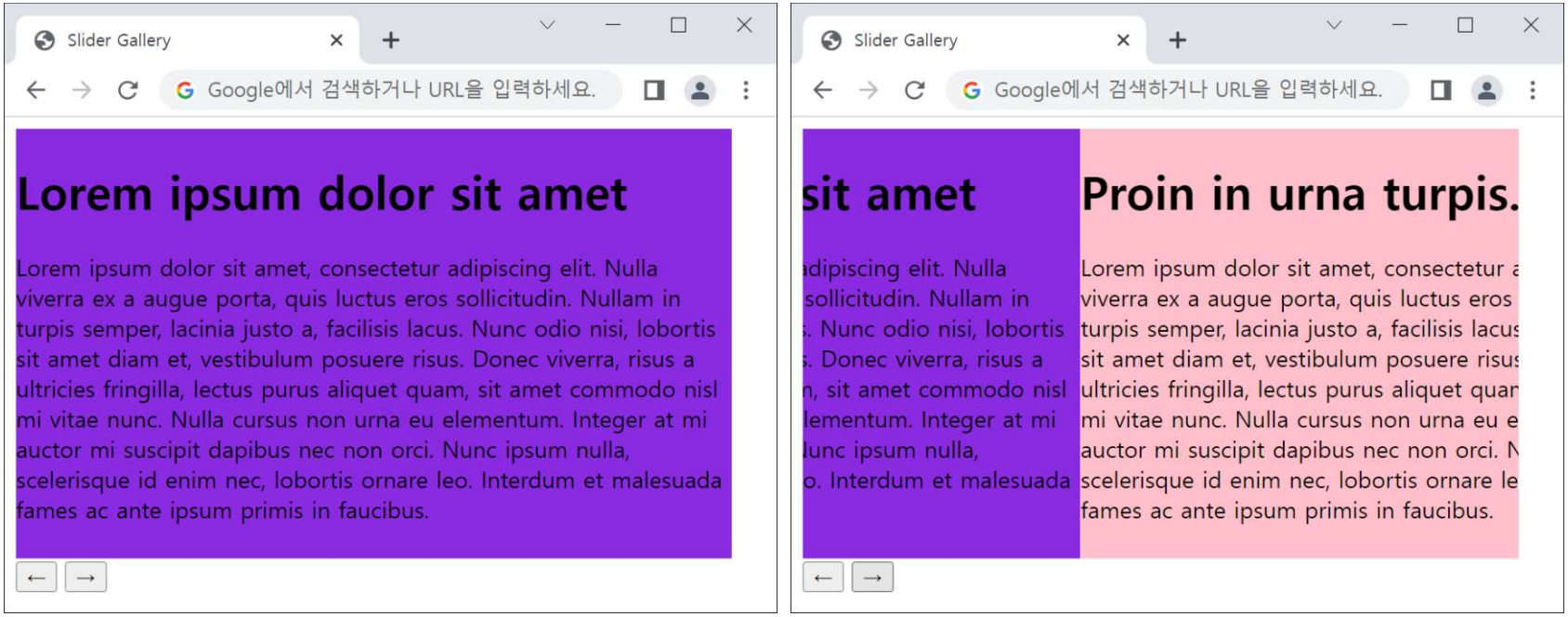
```
// 이벤트를 연결합니다.
$('#left-button').click(function () {
    if (currentPage > 0) {
        // 왼쪽으로 이동
        currentPage = currentPage - 1;
        changePage();
    }
});

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

06. 슬라이더

응용 예제 12-7 jQuery 라이브러리를 사용한 슬라이더 갤러리 구현

7 실행하기



06. 슬라이더

응용 예제 12-7 jQuery 라이브러리를 사용한 슬라이더 갤러리 구현

8 갤러리 전체 코드 확인하기

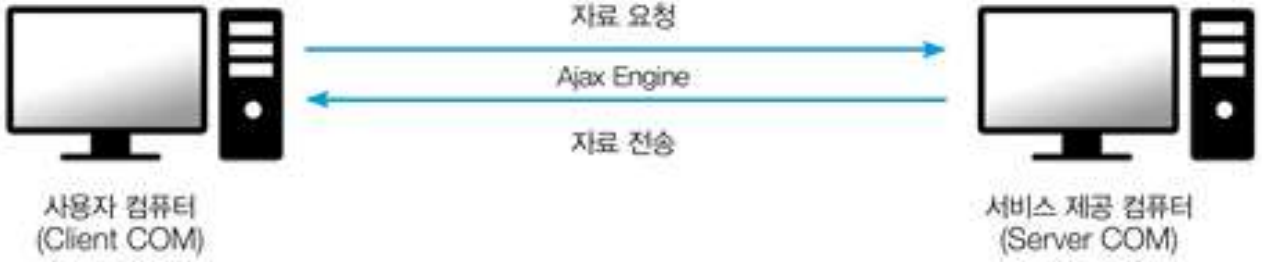
코드 12-17 gallery.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Slider Gallery</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-3.6.4.js"></script>
    <script>

...
```

AJAX

- AJAX
  - **A**synchronous **Ja**vaScript and **X**ML



AJAX

- 장점
  - 웹페이지 속도향상
  - 서버의 처리가 완료될 때까지 기다리지 않고 처리 가능
  - 기존 웹에서 불가능했던 UI를 가능하게 해줌
- 단점
  - 페이지 이동이 없는 통신으로 보안상의 문제가 있을수 있음
  - 연속으로 데이터 요청시 서버 부하가 증가
  - 히스토리 관리가 안된다.
  - Script로 작성되서 디버깅이 용이하지 않음

AJAX

Ajax 관련 메서드의 종류

종류	설명
load( )	외부 콘텐츠를 가져올 때 사용합니다.
$.ajax( )	데이터를 서버에 HTTP POST, GET 방식으로 전송할 수 있으며, HTML, XML, JSON, 텍스트 유형에 데이터를 요청할 수 있는 통합적인 메서드입니다. 이 표에 있는 $.post( ), $.get( ), $.getJSON( ) 메서드의 기능을 하나로 합쳐 놓은 것이라고 보면 됩니다.
$.post( )	데이터를 서버에 HTTP POST 방식으로 전송한 후 서버 측의 응답을 받을 때 사용합니다.
$.get( )	데이터를 서버에 HTTP GET 방식으로 전송한 후 서버 측의 응답을 받을 때 사용합니다.
$.getJSON( )	데이터를 서버에 HTTP GET 방식으로 전송한 후 서버 측의 응답을 JSON 형식으로 받을 때 사용합니다.
$.getScript( )	Ajax를 이용하여 외부 자바스크립트를 불러옵니다. 예 $("button").click(function( ){ $.getScript("demo_ajax_script.js"); });
.ajaxStop(function( ){ ... })	비동기 방식으로 서버에 응답 요청이 완료되었을 때 함수가 실행됩니다.
.ajaxSuccess(function( ){ ... })	ajax 요청이 성공적으로 완료되면 함수가 실행됩니다.
ajaxComplete(function( ){ ... })	Ajax 통신이 완료되면 함수가 실행됩니다.

AJAX

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
   $(document).ready(function(){
     $("button").click(function(){
       $("#div1").load("demo_test.txt");
     });
   });
</script>
</head>
<body>

<div id="div1"><h2>Let jQuery AJAX Change This Text</h2></div>

<button>Get External Content</button>

</body>
</html>
```

AJAX

문법

```
$.ajax({
    type : 'post',            // 타입 (get, post, put 등등)
    url : '/test',            // 요청할 서버url
    async : true,             // 비동기화 여부 (default : true)
    headers : {               // Http header
      "Content-Type" : "application/json",
      "X-HTTP-Method-Override" : "POST"
    },
    dataType : 'text',        // 데이터 타입 (html, xml, json, text 등등)
    data : JSON.stringify({  // 보낼 데이터 (Object , String, Array)
      "no" : no,
      "name" : name,
      "nick" : nick
    }),
    success : function(result) { // 결과 성공 콜백함수
        console.log(result);
    },
    error : function(request, status, error) { // 결과 에러 콜백함수
        console.log(error)
    }
})
```

AJAX

구 분	설 명	값
url	ajax 요청할 매핑된 url 입력	
type	HTTP 통신의 종류를 설정. put, delete는 모든 브라우저에서 지원하는 것이 아니기 때문에 주의가 필요. 기본값(default) : get	get, post, delete, put
dataType	ajax를 통해 리턴받을 데이터의 타입을 설정. 생략했을 경우는, jQuery이 MIME 타입 등을 보면서 자동으로 결정	xml, html, json, script, jsonp, text
contentType	기본값을 가장 많이 사용 기본값(default) : application/x-www-form-urlencoded; charset=UTF-8	
timeout	ajax가 호출된 시점부터 시간을 재서 요청이 초과될 경우 에러 등의 상태로 전환함 $.ajaxSetup()의 global timeout에 의해 override됨.	number(milli secs)
data	URL 파라미터를 통해 보낼 데이터. 종류 : Object or String or Array Object는 key:value set 객체여야 하며 value 영역이 array일 경우 jQuery가 serialize를 해줌. value 영역이 String이 아닌 경우 String으로 변환한 뒤 전송됨	
beforeSend	ajax 실행 시 요청이 전송되기 전에 실행되는 event 입니다. 반환값을 false 또는 jqXHR.abort();로 설정하면 ajax 전송을 취소할 수 있습니다.	
success	ajax 통신이 성공했을 때 실행되는 콜백함수	
error	request가 실패했을 때 실행되는 콜백함수. cross-domain 요청이나 cross-domain의 jsonp 요청에는 작동하지 않는다.	
complete	success나 error가 호출된 이후에 호출되는 콜백함수	

AJAX

```
//    요청한 데이터 : {"member_list":[
//     {"id":"aa1","pw":"bb","addr":"cc","tel":"dd"},
//     {"id":"aa2","pw":"bb","addr":"cc","tel":"dd"},
//     {"id":"aa3","pw":"bb","addr":"cc","tel":"dd"}
//   ]}

<script>
$(document).ready(function(){
    $("#listButton").click(getMemberList); //id="listButton"인 태그에 click하면 function getMemberList() 실행
});
function getMemberList(){
    $.ajax({
        url:"list.jsp",                    //list.jsp에 AJAX요청
        success:function(data){
            let obj=JSON.parse(data);      //data를 받아와서 JSON형태로 변환
            let array=["<ol>"];
            obj["member_list"].forEach(
                    member =>  array.push("<li>"+member.id+"</li>")
                    //JSON에 있는 member.id의 value를 li태그에 넣어서 array에 넣어줌
            );
            array.push("</ol>");

            $("#result").html(array.join(""));
            //array의 요소들을 다 합쳐서 하나로 만든후 id="result"인 태그에 html로 출력
        }
    });
}
</script>
<body>
    <a href="#" id="listButton">회원리스트</a><br/>
    <div id="result">이곳에 회원 목록을 출력하세요</div>
</body>
```

Thank you!

한빛아카데미 Hanbit Academy, Inc.