



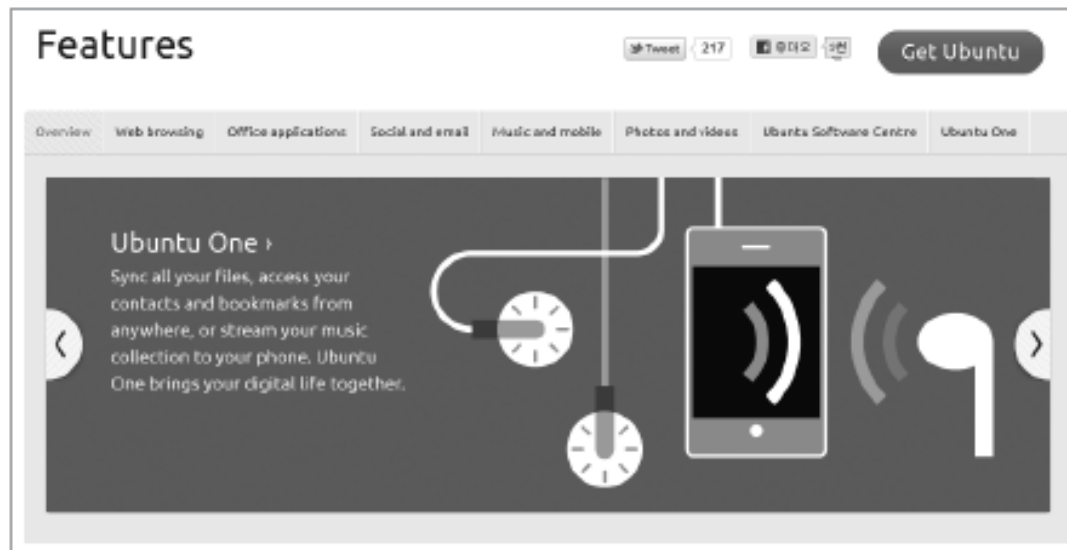
# 18 이미지 슬라이더

모던 웹을 위한 Javascript jQuery 입문

## ❖ 슬라이더란?

- 이미지나 글들이 주기적 또는 선택적으로 넘어가는 요소

그림 18-1 <http://ubuntu.com>의 슬라이더

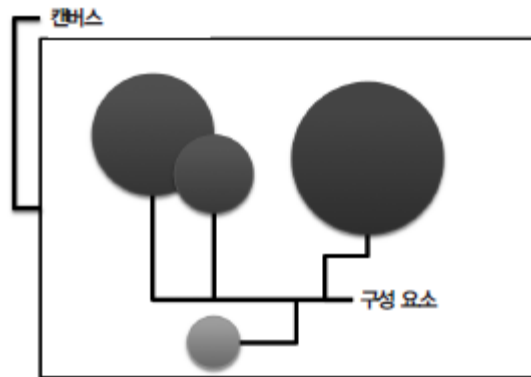


# 18.1 jQuery 애니메이션을 위한 준비

## ❖ 자바스크립트 사용해 플래시나 실버라이트의 시각적 효과 구현

- 세 가지 사항을 CSS로 미리 지정해야
  - 캔버스의 width 스타일 속성과 height 스타일 속성 반드시 지정
  - 캔버스의 position 스타일 속성 Relative로 지정
  - 캔버스의 overflow 스타일 속성 hidden으로 지정
- 구성 요소의 position 스타일 속성은 Absolute로 지정

그림 18-2 캔버스와 구성 요소



# 18.1 jQuery 애니메이션을 위한 준비

## ❖ body 태그 구성

- 외부의 div 태그가 캔버스이고 내부의 div 태그들이 구성 요소
- h1 태그는 캔버스와 애니메이션 구성 요소가 제 위치 있는지 확인하기 위해 놓은 marker

코드 18-1 body 태그 구성

```
<body>
  <h1>Test Header</h1>
  <div id="canvas">
    <div class="inner_box"></div>
    <div class="inner_box"></div>
    <div class="inner_box"></div>
    <div class="inner_box"></div>
  </div>
  <h1>Test Header</h1>
</body>
```



# 18.1 jQuery 애니메이션을 위한 준비

## ❖ style 태그

- 코드 18-2처럼 구성해 각 요소를 구분할 수 있게
- 각 구성 요소 순서대로 배치
  - 내부에 있는 구성 요소에 의해 캔버스 크기 자동 지정

코드 18-2 style 태그 구성

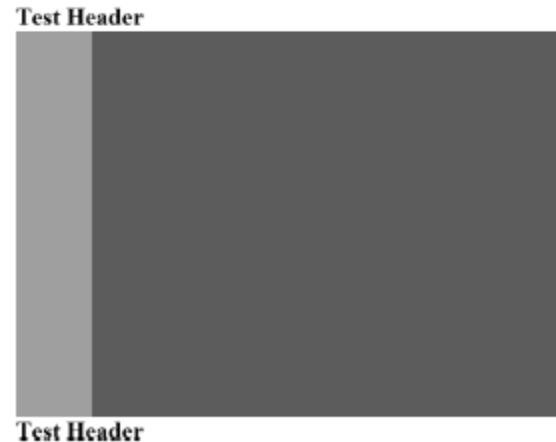
```
<style>
  * { margin:0px; padding:0px; }

  #canvas {
    background:Gray;
  }

  .inner_box {
    background:Orange;

    width:100px;
    height:100px;
  }
</style>
```

그림 18-3 실행 결과



# 18.1 jQuery 애니메이션을 위한 준비

## ❖ Style 태그 사용

- animate() 메서드 사용해 left 속성이나 top 속성 지정
  - position 스타일 속성이 Absolute나 Relative여야
  - 절대 좌표 사용해 애니메이션을 구현하는 것이 일반적
  - 구성 요소는 대부분 Absolute 속성 사용
    - 캔버스의 width 속성과 height 속성이 사라짐
    - 그림 18-4처럼 캔버스 뒤에 있는 h1태그가 위로 올라오게 됨

코드 18-3 style 태그 구성

```
.inner_box
{
    background:orange;

    width:100px;
    height:100px;

    position:absolute;
}
```

그림 18-4 절대 좌표를 놓으면



# 18.1 jQuery 애니메이션을 위한 준비

## ❖ Style 태그 사용 (2)

- 캔버스의 width 스타일 속성과 height 스타일 속성은 반드시 지정

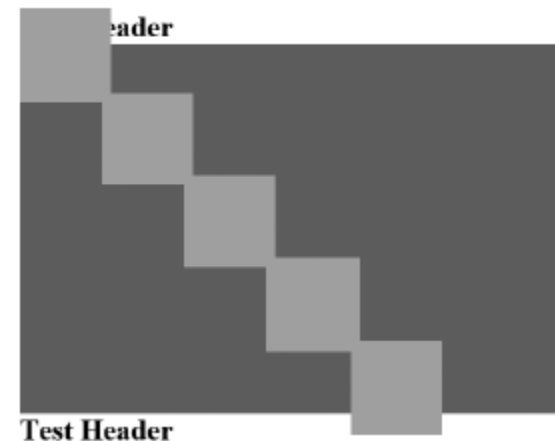
코드 18-4 style 태그 구성

```
#canvas
{
    background:Gray;
    width:600px;
    height:400px;
}
```

코드 18-5 script 태그 구성

```
<script>
$(document).ready(function () {
    $('.inner_box').each(function (index) {
        $(this).css({
            left: index * 90,
            top: index * 90
        });
    });
});
</script>
```

그림 18-5 절대 좌표로 인해 발생하는 문제



## ❖ Style 태그 사용 (3)

- 부모의 position 스타일 속성이 Absolute나 Relative일 때
  - 자식 구성 요소가 부모의 위치 기준으로 위치 잡음
  - 구성 요소뿐만 아니라 캔버스에도 position 스타일 속성 지정해야 함
  - 일반적으로 캔버스의 position 스타일 속성에는 Relative 부여
- overflow 스타일 속성
  - 내용이 사용할 수 있는 범위를 초과하면 어떻게 처리하는지 나타냄
  - overflow 스타일 속성에 hidden 값 입력
    - 캔버스의 공간을 넘어가는 내용은 보이지 않게 함





# 18.1 jQuery 애니메이션을 위한 준비

## ❖ Style 태그 사용 (4)

코드 18-6

```
#canvas
{
    background:Gray;
    width:600px;
    height:400px;
    position:relative;
    overflow:hidden;
}
```

그림 18-6 적절하게 자리잡은 캔버스 내부의 구성 요소

Test Header



Test Header



## 18.2 이미지 슬라이더 (1) - HTML 구성

### ❖ 그림 18-7]

- 이미지 슬라이더는 사용자의 눈에 보이는 것은 일부분
- 보이지 않는 곳에서 모든 이미지가 미리 준비하고 있어야 함

그림 18-7 이미지 슬라이더의 기본 구성



# 18.2 이미지 슬라이더 (1) - HTML 구성

## ❖ 텍스트도 함께 움직이는 이미지 슬라이더

### ■ 1) HTML 페이지 구성

코드 18-7 HTML 페이지 구성

```
<!DOCTYPE html>
<html>
<head>
  <style>

  </style>
  <script src="http://code.jquery.com/jquery-1.7.js"></script>
  <script>

  </script>
</head>
<body>

</body>
</html>
```



# 18.2 이미지 슬라이더 (1) - HTML 구성

## ❖ 텍스트도 함께 움직이는 이미지 슬라이더

- 캔버스를 구성할 div 태그
  - 애니메이션 요소가 위치를 제대로 차지하는지 확인하려고 h1 태그 입력

코드 18-8

```
<body>
  <h1>Test Header</h1>
  <div class="animation_canvas">

  </div>
  <h1>Test Header</h1>
</body>
```

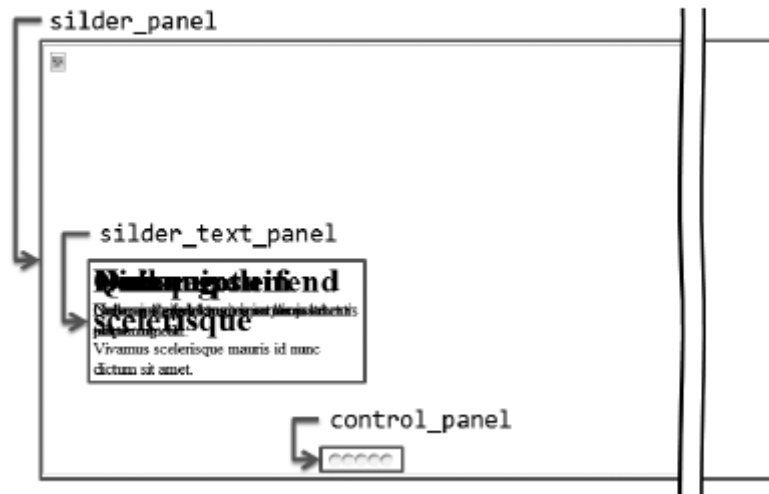


# 18.2 이미지 슬라이더 (1) - HTML 구성

## ❖ 캔버스 내부 분할

- slider\_panel - 세 가지 id 갖는 공간 만들기
  - 이미지를 담는 패널로, 움직이며 이미지가 전환되는 효과를 만듦
  - slider\_text\_panel: 글자를 담는 패널
  - control\_panel: 컨트롤 버튼 담는 패널

그림 18-8 이미지슬라이더의 HTML 태그 구성



# 18.2 이미지 슬라이더 (1) - HTML 구성

## ❖ body 태그 구성

코드 18-9 body 태그 구성

```
<body>
  <h1>Test Header</h1>
  <div class="animation_canvas">

    <div class="slider_panel">

      </div>
      <div class="slider_text_panel">

        </div>
        <div class="control_panel">

          </div>
        </div>
      <h1>Test Header</h1>
</body>
```



# 18.2 이미지 슬라이더 (1) - HTML 구성

## ❖ Body 태그 완성

코드 18-10 body 태그 구성

```
<body>
  <h1>Test Header</h1>
  <div class="animation_canvas">
    <div class="slider_panel">
      <img class="slider_image"/>
      <img class="slider_image"/>
      <img class="slider_image"/>
      <img class="slider_image"/>
      <img class="slider_image"/>
    </div>
    <div class="slider_text_panel">
      <div class="slider_text">
        <h1>Lorem ipsum</h1>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
      </div>
      <div class="slider_text">
        <h1>Nulla eget</h1>
        <p>Nulla eget sapien mauris, ornare elementum elit.</p>
      </div>
      <div class="slider_text">
        <h1>Quisque eleifend</h1>
        <p>Quisque eleifend augue nec lacus lobortis porta.</p>
      </div>
      <div class="slider_text">
        <h1>Donec</h1>
        <p>Donec a ligula lectus, eu iaculis justo.</p>
      </div>
    </div>
  </div>
```



# 18.2 이미지 슬라이더 (1) - HTML 구성

## ❖ Body 태그 완성 (2)

```
</div>
<div class="slider_text">
  <h1>Vivamus scelerisque</h1>
  <p>Vivamus scelerisque mauris id nunc dictum sit amet.</p>
</div>
</div>
<div class="control_panel">
  <div class="control_button"></div>
  <div class="control_button"></div>
  <div class="control_button"></div>
  <div class="control_button"></div>
  <div class="control_button"></div>
</div>
</div>
<h1>Test Header</h1>
</body>
```





# 18.3 이미지 슬라이더 (2) - CSS 구성

## ❖ Style 태그 구성

코드 18-11 style 태그 구성

```
<style>
  * { margin:0px; padding:0px; }

  /* Animation Canvas */
  .animation_canvas {
    /*overflow:hidden;*/
    position:relative;
    width:600px; height:400px;
  }

  /* Slider Panel */
  .slider_panel { width:3000px; position:relative; }
  .slider_image { float:left; width:600px; height:400px; }

  /* Slider Text Panel */
  .slider_text_panel { position:absolute; top:200px; left:50px; }
  .slider_text { position:absolute; width:250px; height:150px; }

  /* Control Panel */
  .control_panel {
    position:absolute; top:380px;
    left:270px; height:13px;
    overflow:hidden;
  }
}
```



## ❖ Style 태그 구성 (2)

```
.control_button {  
    width:12px; height:46px;  
    position:relative;  
  
    float:left; cursor:pointer;  
    background:url('point_button.png');  
}  
.control_button:hover { top:-16px; }  
.control_button.active { top:-31px; }  
</style>
```

- **control\_button** 클래스의 **hover**와 **active**에 **top** 속성 지정
  - point\_button.png 파일은 그림 18-9 참조해 이해
  - 이미지가 변하는 것과 비슷한 효과 나타냄

그림 18-9 point\_button.png

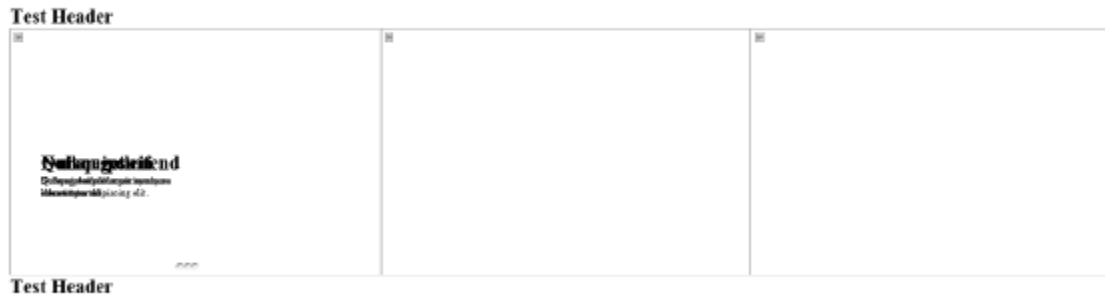


# 18.3 이미지 슬라이더 (2) - CSS 구성

## ❖ Style 태그 구성 (3)

- 현재까지 구현된 페이지

그림 18-10 현재까지의 결과



## ❖ Script 태그 구성 단계

- 초기 텍스트 위치 지정
- 이벤트 연결
- 각 텍스트와 컨트롤 버튼의 위치를 알 수 있게 data-index 할당
- 이동 함수 제작
- 초기 슬라이더 위치 설정



# 18.4 이미지 슬라이더 (3) - 자바스크립트 구성

## ❖ 초기 텍스트 위치 지정

- 스타일 시트에서 처리도 가능

코드 18-12 초기 텍스트의 위치 지정

```
<script>
$(document).ready(function () {
    // 초기 텍스트의 위치를 지정합니다.
    $('.slider_text').css('left', -300);
});
</script>
```



# 18.4 이미지 슬라이더 (3) - 자바스크립트 구성

## ❖ 초기 텍스트 위치 지정

- 각 요소에 each() 메서드를 사용해 data-index 부여
- 사용자가 추가로 속성 입력할 때는 data-로 시작하는 것
  - 웹 표준 권고 사항

코드 18-13 data-index의 속성 부여

```
<script>
$(document).ready(function () {
    // 초기 텍스트 위치 지정과 data-index 할당
    $('.slider_text').css('left', -300).each(function (index) {
        $(this).attr('data-index', index);
    });

    // 컨트롤 버튼의 클릭 핸들러 지정과 data-index 할당
    $('.control_button').each(function (index) {
        $(this).attr('data-index', index);
    });
});
</script>
```

그림 18-11 크롬 요소 검사를 통해 data-index 속성이 차례대로 들어가 있음을 확인

```
<div class="slider_text_panel">
  <div class="slider_text" style="left: -300px; " data-index="0">.</div>
  <div class="slider_text" style="left: -300px; " data-index="1">.</div>
  <div class="slider_text" style="left: -300px; " data-index="2">.</div>
  <div class="slider_text" style="left: -300px; " data-index="3">.</div>
  <div class="slider_text" style="left: -300px; " data-index="4">.</div>
</div>
<div class="control_panel">
  <div class="control_button" data-index="0"></div>
  <div class="control_button" data-index="1"></div>
  <div class="control_button" data-index="2"></div>
  <div class="control_button" data-index="3"></div>
  <div class="control_button" data-index="4"></div>
</div>
```



## 18.4 이미지 슬라이더 (3) - 자바스크립트 구성

### ❖ 이벤트 연결

- control\_button을 클릭했을 때 moveSlider() 함수 호출
- 처음 시작할 때 moveSlider() 함수 호출해 랜덤한 위치로 이동

코드 18-14 이벤트 연결

```
<script>
    $(document).ready(function () {
        // 슬라이더를 움직여주는 함수
        function moveSlider(index) {

        }

        // 초기 텍스트의 위치 지정과 data-index 할당
        $('.slider_text').css('left', -300).each(function (index) {
            $(this).attr('data-index', index);
        });

        // 컨트롤 버튼의 클릭 핸들러 지정과 data-index 할당
        $('.control_button').each(function (index) {
            $(this).attr('data-index', index);
        }).click(function () {
            var index = $(this).attr('data-index');
            moveSlider(index);
        });

        // 초기 슬라이더 위치 지정
        var randomNumber = Math.round(Math.random() * 5);
        moveSlider(randomNumber);
    });
</script>
```



# 18.4 이미지 슬라이더 (3) - 자바스크립트 구성

## ❖ moveSlider() 함수

코드 18-15 moveSlider() 함수

```
// 슬라이더를 움직여주는 함수
function moveSlider(index) {
    // 슬라이더를 이동합니다.
    var willMoveLeft = -(index * 600);
    $('.slider_panel').animate({ left: willMoveLeft }, 'slow');
    // control_button에 active 클래스를 부여/제거합니다.
    $('.control_button[data-index=' + index + ']').addClass('active');
    $('.control_button[data-index!=' + index + ']').removeClass('active');

    // 글자를 이동합니다.
    $('.slider_text[data-index=' + index + ']').show().animate({
        left: 0
    }, 'slow');
    $('.slider_text[data-index!=' + index + ']').hide('slow', function () {
        $(this).css('left', -300);
    });
}
```

그림 18-12 최종 완성 결과

Test Header



Test Header







# Thank You !

모던 웹을 위한 Javascript jQuery 입문