

jquery.smoothState.js

개요

- 부드러운 페이지 전환효과를 적용할 수 있는 jQuery Plugin
- smoothState 에 대한 소개 및 동작원리

Get started with smoothState.js

Why use smoothState.js?

- hard cut, white flash 등의 현상은 유저의 집중력을 흐트러뜨리고 레이아웃 변경이나 엘리먼트 재배열 같은 혼란을 야기
- smoothState.js를 사용하면 위와 같은 현상을 제거하는 트랜지션을 추가할 수 있다.
- 아래 요소들이 사용?됨
 - Progressive enhancement
 - jQuery
 - history.pushState()
 - Ajax
- smoothState.js는 웹사이트 페이지 로드가 보다 더 SPA framework과 같이 동작하도록 겸손하게 향상(Unobtrusive JavaScript)시켜줄 것이다.
- 페이지 트랜지션을 추가하고 더 나은 UX를 제공할 수 있다.

Requirements

- ~~링크가 아닌~~ 컨테이너에 초기화
- 요구사항
 - 모든 컨테이너는 id 필요
 - 모든 링크는 전체 레이아웃 리턴(not just an HTML fragment)
- 에러 발생 시 중단하고 레다이렉트를 허용하기 때문에 웹사이트가 보다 탄력적이게 만든다.
- 페이지 트랜지션을 보장하기 위해 각 링크는 정규화된 페이지를 반환하도록 한다.
- smoothState.js는 업데이트되는 컨테이너가 서버로부터의 응답 내에 존재하는 한 제대로 동작할 것이다.
- 개발모드 활성화를 통해 콘솔창으로 경고 메시지를 확인할 수 있다.

A barebones example

- 시작을 위해 필요한 것
 - 페이지에 jQuery, jquery.smoothState.js 포함
 - 새로운 js 파일 생성 및 \$('#main').smoothState() 실행
 - 내부 링크들을 포함한 컨테이너('#main') 추가

The HTML

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Home - My Site</title>
  </head>
  <body>
    <!-- Every smoothState container needs an id -->
    <div id="main">
      <!-- Every link should return a full layout -->
      <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="about.html">About</a></li>
      </ul>
      <!-- Contents of the page... -->
    </div>
    <!-- Scripts -->
    <script src="jquery.js"></script>
    <script src="jquery.smoothState.js"></script>
    <script src="functions.js"></script>
  </body>
</html>
```

The JS

```
// Contents of functions.js
;(function ($) {
  $('#main').smoothState();
})(jQuery);
```

- [데모 페이지](#)

Add page transitions to your site

What makes this a 'typical' example?

- 애니메이션은 완성시간이 오래걸리며 각 레이아웃은 페이지 상의 엘리먼트들이 나타나고 사라지는 방법에 대한 주의 깊은 choreography 이 요구된다.
- 간단한 fade 효과는 약간의 노력으로 매끄럽게 동작할 수 있도록 해주는 완벽한 방법

Adding page transitions

- 트랜지션 추가를 위해선, 페이지에 (컨텐츠가)삽입되는 방식을 처리하는 기본 callback을 재정의 해야한다.

The HTML

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Home - My Site</title>
    <link href="keyframes.css">
    <link href="pageTransitions.css">
  </head>
  <body>
    <!-- Animation class -->
    <div id="main" class="m-scene">
      <!-- Classes that define element animations -->
      <div class="scene_element scene_element--fadein">
        <!-- contents... -->
      </div>
    </div>
    <!-- Scripts -->
    <script src="jquery.js"></script>
    <script src="jquery.smoothState.js"></script>
    <script src="functions.js"></script>
  </body>
</html>
```

The CSS

```
/*
 * CSS Animations
 * Don't forget to add vendor prefixes!
 */
.m-scene .scene_element {
  animation-duration: 0.25s;
  transition-timing-function: ease-in;
  animation-fill-mode: both;
}

.m-scene .scene_element--fadein {
  animation-name: fadeIn;
}

.m-scene.is-exiting .scene_element {
  animation-direction: alternate-reverse;
}

/*
 * Keyframes
 */
@keyframes fadeIn {
  0% { opacity: 0; }
  100% { opacity: 1; }
}
```

The JS

```
// Contents of functions.js
;(function($) {
  'use strict';
  var $body = $('html, body'),
      content = $('#main').smoothState({
        // Runs when a link has been activated
        onStart: {
          duration: 250, // Duration of our animation
          render: function (url, $container) {
            // toggleAnimationClass() is a public method
            // for restarting css animations with a class
            content.toggleAnimationClass('is-exiting');
            // Scroll user to the top
            $body.animate({
              scrollTop: 0
            });
          }
        }
      })
  })
  .data('smoothState');
//.data('smoothState') makes public methods available
})(jQuery);
```

- [데모 페이지](#)

Other animation effects

- @keyframes 정의를 통해 더 복잡한 애니메이션 적용 가능
- Daniel Eden이 [animate.css](#) 라는 css 애니메이션 라이브러리 작성

Boost your site's performance

Quicker response times

- 세가지 주요 callback
 - onStart: 링크가 활성화될 때 실행
 - onProgress: 페이지 요청이 여전히 로딩중이고, onStart는 애니메이션을 완료된 경우 실행
 - onEnd: 요청된 콘텐츠가 페이지에 삽입될 준비가 되었을 때 실행

Prefetch content

- 마우스 오버와 클릭 사이에는 지연시간이 있다.
- prefetch 옵션값을 true로 설정하면, smoothState는 이 지연시간 사이에 url 콘텐츠를 미리 로드하기 시작한다.
- 웹사이트 속도의 극적 향상

```
$('#main').smoothState({ prefetch: true });
```

Caching content

- pageCacheSize가 0보다 큰 값으로 설정되어있는 경우 메모리에 페이지를 저장
- 사용자가 한번 이상 페이지를 요청하는 것을 방지할 수 있다.
- 메모리에 저장된 페이지는 즉시 로드

```
$('#main').smoothState({ pageCacheSize: 4 });
```

Demo and code

```
;(function ($) {  
  'use strict';  
  var $body = $('html, body'),  
      content = $('#main').smoothState({  
    prefetch: true,  
    pageCacheSize: 4,  
    onStart: {  
      duration: 250,  
      render: function (url, $container) {  
        content.toggleAnimationClass('is-exiting');  
        $body.animate({  
          scrollTop: 0  
        });  
      }  
    }  
  }).data('smoothState');  
})(jQuery);
```

References

- <http://weblinc.github.io/jquery.smoothState.js/index.html>