마우스 휠 및 팝업 이벤트

이 코드는 크게 5가지 주요 기능으로 나뉘며, 웹사이트에서 흔히 사용되는 **풀페이지 스크롤(Full-page Scroll) 효과** 및 **팝업 제어** 기능을 구현하고 있습니다.

**1. 초기 설정 및 스크롤 이벤트 정의**

| 코드 | 설명 |
| --- | --- |
| $(function(){...}); | **Document Ready:** HTML 문서가 완전히 로드되면 내부의 코드를 실행합니다. |
| var f\_top = $('#float\_div').offset().top; | #float\_div 요소의 **초기 수직 위치**를 f\_top 변수에 저장합니다. 이 값은 뒤에서 요소가 스크롤을 따라다니게 만드는 데 사용됩니다. |
| $(window).scroll(function(){...}); | **스크롤 이벤트:** 사용자가 창을 스크롤할 때마다 이 안에 있는 모든 코드가 반복적으로 실행됩니다. |
| let sct = $(this).scrollTop(); | 현재 창의 \*\*수직 스크롤 위치 (Scroll Top)\*\*를 sct 변수에 저장합니다. |
| $('.s\_Top').text(sct); | 현재 스크롤 위치(sct) 값을 .s\_Top이라는 요소에 텍스트로 표시합니다. |

**2. 핵심 스크롤 기반 기능 (Scroll-Driven Features)**

**A. 플로팅 요소 (스크롤 따라다니기)**

JavaScript

$('#float\_div').stop().animate({top:f\_top+sct},300);

* #float\_div 요소를 **스크롤을 따라다니도록** 만듭니다. top 값을 \*\*초기 위치(f\_top) + 현재 스크롤 위치(sct)\*\*로 계속 업데이트하여, 스크롤을 내려도 요소가 뷰포트 내의 같은 위치에 떠 있는 것처럼 보이게 합니다.

**B. 특정 스크롤 위치 기반 애니메이션**

특정 스크롤 영역에 도달하면 CSS 클래스를 추가(addClass('on'), addClass('active'))하여 애니메이션이나 스타일을 적용합니다.

| 스크롤 범위 | 대상 요소 | 동작 |
| --- | --- | --- |
| sct >= 450 이고 sct <= 1000 | .left1 | on 클래스를 추가/제거 (등장/사라짐 효과) |
| sct >= 1000 이고 sct <= 1800 | .right1 | on 클래스를 추가/제거 |
| sct >= 2500 | .s4\_1, .s4\_2, .s4\_3, .s4\_4 | active 클래스를 **0ms, 400ms, 800ms, 1200ms**의 지연시간을 두고 순차적으로 추가하여 애니메이션을 만듭니다 (캐스케이딩 효과). |

**C. 내비게이션 활성화 (현재 섹션 표시)**

JavaScript

$('.container > div').each(function(i){...});

* 페이지의 모든 메인 섹션(.container > div)을 반복 확인합니다.
* **현재 스크롤 위치가 섹션의 시작 위치를 지났다면**, 해당 섹션과 연결된 \*\*상단 메뉴 (nav ul li)\*\*와 \*\*플로팅 메뉴 (#float\_div ul li)\*\*에 on 클래스를 추가하여 현재 보고 있는 섹션을 표시합니다.

**D. 고정 내비게이션 (Sticky Navigation)**

JavaScript

if(sct >=300){ $('nav').addClass('fixed'); } else { $('nav').removeClass('fixed'); }

* 스크롤 위치가 **300px 이상**이 되면 <nav> 요소에 fixed 클래스를 추가하여 상단에 고정되게 만듭니다.

**3. 내비게이션 클릭 이벤트 (부드러운 스크롤 이동)**

JavaScript

$('nav ul li').click(function(){...});

$('#float\_div ul li').click(function(){...});

* 상단 메뉴(nav ul li) 또는 플로팅 메뉴 (#float\_div ul li) 항목을 클릭하면 실행됩니다.
* 클릭한 메뉴 항목의 순서(i)를 가져와 해당 순서에 맞는 콘텐츠 섹션 (.container > div)의 위치를 계산합니다.
* $('html,body').stop().animate({scrollTop:offset\_t},1000);를 사용하여 페이지를 **1초 동안 부드럽게(animate)** 해당 섹션의 시작 위치로 이동시킵니다.

**4. 마우스 휠 이벤트 (섹션별 스크롤)**

JavaScript

$('.container > div').mousewheel(function(event,d){...});

* **jquery.mousewheel.js** 라이브러리를 사용하여 콘텐츠 섹션 위에서 마우스 휠이 작동할 때의 이벤트를 처리합니다.
* **d > 0 (휠 올림)**: 이전 섹션의 위치로 이동합니다.
* **d < 0 (휠 내림)**: 다음 섹션의 위치로 이동합니다.
* 'easeOutBounce' 효과를 사용하여 **튕기는 듯한** 독특한 애니메이션으로 섹션 간에 이동합니다.

**5. 팝업 및 공지사항 제어 (쿠키 사용)**

이 코드는 \*\*$.cookie\*\*라는 jQuery 플러그인을 사용하여 사용자가 팝업을 닫았는지 여부를 쿠키에 저장하고 하루 또는 며칠 동안 다시 팝업이 뜨지 않도록 제어합니다.

**A. 메인 팝업 (#popup)**

* $('#popup').draggable(); : 팝업을 **드래그**할 수 있게 합니다.
* if($.cookie('pop')!='no'): **pop 쿠키의 값이 'no'가 아니라면** 팝업을 보여줍니다.
* $('#popup area:eq(1)').click: 두 번째 area 버튼(보통 "하루 동안 보지 않기")을 클릭하면, **pop 쿠키를 1일 동안 'no'로 저장**한 뒤 팝업을 닫습니다.

**B. 공지사항 (#notice\_wrap)**

* if($.cookie('popup')=='none'): **popup 쿠키의 값이 'none'이라면** 공지사항을 숨깁니다.
* closePop 함수:
  + **체크박스(chk)가 체크되어 있다면**, popup 쿠키를 **3일 동안 'none'으로 저장**합니다. (즉, "3일간 다시 보지 않기" 기능).
  + 그 후 공지사항을 닫습니다.