// Grab elements

const selectElement = (selector) => {

const element = document.querySelector(selector);

if(element) return element;

throw new Error(`Something went wrong! Make sure that ${selector} exists/is typed correctly.`);

};

//Nav styles on scroll

const scrollHeader = () =>{

const navbarElement = selectElement('#header');

if(this.scrollY >= 15) {

navbarElement.classList.add('activated');

} else {

navbarElement.classList.remove('activated');

}

}

window.addEventListener('scroll', scrollHeader);

// Open menu & search pop-up

const menuToggleIcon = selectElement('#menu-toggle-icon');

const formOpenBtn = selectElement('#search-icon');

const formCloseBtn = selectElement('#form-close-btn');

const searchContainer = selectElement('#search-form-container');

const toggleMenu = () =>{

const mobileMenu = selectElement('#menu');

mobileMenu.classList.toggle('activated');

menuToggleIcon.classList.toggle('activated');

}

menuToggleIcon.addEventListener('click', toggleMenu);

// Open/Close search form popup

formOpenBtn.addEventListener('click', () => searchContainer.classList.add('activated'));

formCloseBtn.addEventListener('click', () => searchContainer.classList.remove('activated'));

// -- Close the search form popup on ESC keypress

window.addEventListener('keyup', (event) => {

if(event.key === 'Escape') searchContainer.classList.remove('activated');

});

// Switch theme/add to local storage

const body = document.body;

const themeToggleBtn = selectElement('#theme-toggle-btn');

const currentTheme = localStorage.getItem('currentTheme');

// Check to see if there is a theme preference in local Storage, if so add the ligt theme to the body

if (currentTheme) {

body.classList.add('light-theme');

}

themeToggleBtn.addEventListener('click', function () {

// Add light theme on click

body.classList.toggle('light-theme');

// If the body has the class of light theme then add it to local Storage, if not remove it

if (body.classList.contains('light-theme')) {

localStorage.setItem('currentTheme', 'themeActive');

} else {

localStorage.removeItem('currentTheme');

}

});

// Swiper

const swiper = new Swiper(".swiper", {

// How many slides to show

slidesPerView: 1,

// How much space between slides

spaceBetween: 20,

// Make the next and previous buttons work

navigation: {

nextEl: '.swiper-button-next',

prevEl: '.swiper-button-prev',

},

// Make the pagination indicators work

pagination: {

el: '.swiper-pagination'

},

//Responsive breakpoints for how many slides to show at that view

breakpoints: {

// 700px and up shoes 2 slides

700: {

slidesPerView: 2

},

// 1200px and up shoes 3 slides

1200: {

slidesPerView: 3

}

}

});