| **Name** | **Type** | **Default** | **Description** |
| --- | --- | --- | --- |
| [a11y](https://swiperjs.com/swiper-api#param-a11y) | any |  | Объект с параметрами a11y или логическим значением true включить с настройками по умолчанию.  const swiper = new Swiper('.swiper', {  a11y: {  prevSlideMessage: 'Previous slide',  nextSlideMessage: 'Next slide',  },  }); |
| [allowSlideNext](https://swiperjs.com/swiper-api#param-allowSlideNext) | boolean | true | Установлено значение false чтобы отключить переход к следующему направлению слайда (вправо или вниз) |
| [allowSlidePrev](https://swiperjs.com/swiper-api#param-allowSlidePrev) | boolean | true | Установлено значение false чтобы отключить прокрутку в предыдущем направлении слайда (влево или вверх) |
| [allowTouchMove](https://swiperjs.com/swiper-api#param-allowTouchMove) | boolean | true | Если false, тогда единственным способом переключения слайда является использование внешних функций API , таких как slidePrev или slideNext |
| [autoHeight](https://swiperjs.com/swiper-api#param-autoHeight) | boolean | false | Установлено значение true а оболочка слайдера адаптирует свою высоту к высоте текущего активного слайда |
| [autoplay](https://swiperjs.com/swiper-api#param-autoplay) | any |  | Объект с параметрами автозапуска или логическим значением true для включения с настройками по умолчанию  const swiper = new Swiper('.swiper', {  autoplay: {  delay: 5000,  },  }); |
| [breakpoints](https://swiperjs.com/swiper-api#param-breakpoints) | object |  | Позволяет устанавливать разные параметры для разных точек останова реагирования (размеров экрана). Не все параметры могут быть изменены в точках останова, только те, которые не требуют другого расположения и логики, например slidesPerView, slidesPerGroup, spaceBetween grid.rows, loop effect не сработает  const swiper = new Swiper('.swiper', {  // Default parameters  slidesPerView: 1,  spaceBetween: 10,  // Responsive breakpoints  breakpoints: {  // when window width is >= 320px  320: {  slidesPerView: 2,  spaceBetween: 20  },  // when window width is >= 480px  480: {  slidesPerView: 3,  spaceBetween: 30  },  // when window width is >= 640px  640: {  slidesPerView: 4,  spaceBetween: 40  }  }  })  const swiper = new Swiper('.swiper', {  slidesPerView: 1,  spaceBetween: 10,  // using "ratio" endpoints  breakpoints: {  '@0.75': {  slidesPerView: 2,  spaceBetween: 20,  },  '@1.00': {  slidesPerView: 3,  spaceBetween: 40,  },  '@1.50': {  slidesPerView: 4,  spaceBetween: 50,  },  }  }); |
| [breakpointsBase](https://swiperjs.com/swiper-api#param-breakpointsBase) | string | 'window' | База для контрольных точек (бета-версия). Может быть window или container. Если установлено значение window (по умолчанию) тогда клавиши точки останова означают ширину окна. Если установлено значение container затем клавиши точки останова обрабатываются как ширина контейнера swiper |
| [cardsEffect](https://swiperjs.com/swiper-api#param-cardsEffect) | any |  | Объект с карточками-параметры эффекта  const swiper = new Swiper('.swiper', {  effect: 'cards',  cardsEffect: {  // ...  },  }); |
| [centerInsufficientSlides](https://swiperjs.com/swiper-api#param-centerInsufficientSlides) | boolean | false | При включении он центрирует слайды, если количество слайдов меньше slidesPerView. Не предназначен для использования loop режим и grid.rows |
| [centeredSlides](https://swiperjs.com/swiper-api#param-centeredSlides) | boolean | false | Если true, тогда активный слайд будет располагаться по центру, не всегда с левой стороны. |
| [centeredSlidesBounds](https://swiperjs.com/swiper-api#param-centeredSlidesBounds) | boolean | false | Если true, тогда активный слайд будет центрирован без добавления пробелов в начале и конце слайдера. Требуемый centeredSlides: true. Не предназначен для использования с loop или pagination |
| [containerModifierClass](https://swiperjs.com/swiper-api#param-containerModifierClass) | string | 'swiper-' | Начало CSS-класса-модификатора, который может быть добавлен в контейнер swiper в зависимости от различных параметров |
| [controller](https://swiperjs.com/swiper-api#param-controller) | any |  | Объект с параметрами контроллера или логическим значением true для включения с настройками по умолчанию  const swiper = new Swiper('.swiper', {  controller: {  inverse: true,  },  }); |
| [coverflowEffect](https://swiperjs.com/swiper-api#param-coverflowEffect) | any |  | Объект с параметрами эффекта Coverflow.  const swiper = new Swiper('.swiper', {  effect: 'coverflow',  coverflowEffect: {  rotate: 30,  slideShadows: false,  },  }); |
| [createElements](https://swiperjs.com/swiper-api#param-createElements) | boolean | false | Когда Swiper включен, он автоматически оборачивает слайды элементом swiper-wrapper и создает необходимые элементы для навигации, разбивки на страницы и полосы прокрутки, если они включены (с соответствующим объектом params или с логическим значением true)) |
| [creativeEffect](https://swiperjs.com/swiper-api#param-creativeEffect) | any |  | Объект с параметрами креативного эффекта  const swiper = new Swiper('.swiper', {  effect: 'creative',  creativeEffect: {  prev: {  // will set `translateZ(-400px)` on previous slides  translate: [0, 0, -400],  },  next: {  // will set `translateX(100%)` on next slides  translate: ['100%', 0, 0],  },  },  }); |
| [cssMode](https://swiperjs.com/swiper-api#param-cssMode) | boolean | false | Когда он включен, он будет использовать современный CSS Scroll Snap API. Он не поддерживает все функции Swiper, но потенциально должен обеспечить гораздо более высокую производительность в простых конфигурациях.  Это то, что не поддерживается, когда оно включено:   * Эффект куба * speed параметр может не иметь никакого эффекта * Все события, связанные с началом / завершением перехода (используйте slideChange вместо этого) * slidesPerGroup имеет ограниченную поддержку * simulateTouch не имеет эффекта, и "перетаскивание" мышью не работает * resistance не оказывает никакого эффекта * allowSlidePrev/Next * swipeHandler   В случае, если вы используете его с другими эффектами, особенно с 3D-эффектами, необходимо обернуть содержимое слайда <div className="swiper-slide-transform"> элемент. И если вы используете какие-либо пользовательские стили на слайдах (например, цвета фона, радиус границы, бордюр и т.д.), они должны быть установлены на swiper-slide-transform элемент вместо этого.  <div class="swiper">  <div class="swiper-wrapper">  <div class="swiper-slide">  <!-- wrap slide content with transform element -->  <div class="swiper-slide-transform">  ... slide content ...  </div>  </div>  ...  </div>  </div>  <script>  const swiper = new Swiper('.swiper', {  effect: 'flip',  cssMode: true,  });  </script> |
| [cubeEffect](https://swiperjs.com/swiper-api#param-cubeEffect) | any |  | Объект с параметрами кубического эффекта  const swiper = new Swiper('.swiper', {  effect: 'cube',  cubeEffect: {  slideShadows: false,  },  }); |
| [direction](https://swiperjs.com/swiper-api#param-direction) | 'horizontal' | 'vertical' | 'horizontal' | Может быть 'horizontal' или 'vertical' (для вертикального слайдера). |
| [edgeSwipeDetection](https://swiperjs.com/swiper-api#param-edgeSwipeDetection) | string | boolean | false | Включите выпуск событий Swiper для работы с возвратом в приложении. Если установлено значение 'prevent' тогда вместо этого это предотвратит системную навигацию с помощью свайпа назад. Эта функция работает только с событиями "касания" (а не с событиями указателя), поэтому она будет работать на устройствах iOS / Android и не будет работать на устройствах Windows с событиями указателя (касания). |
| [edgeSwipeThreshold](https://swiperjs.com/swiper-api#param-edgeSwipeThreshold) | number | 20 | Область (в пикселях) от левого края экрана для разблокировки событий касания для возврата в приложение |
| [effect](https://swiperjs.com/swiper-api#param-effect) | string | 'slide' | Эффект перехода. Может быть 'slide', 'fade' 'cube', 'coverflow' 'flip' или 'creative' |
| [enabled](https://swiperjs.com/swiper-api#param-enabled) | boolean | true | Включен ли Swiper изначально. Когда Swiper отключен, он скроет все элементы навигации и не будет реагировать ни на какие события и взаимодействия |
| [eventsPrefix](https://swiperjs.com/swiper-api#param-eventsPrefix) | string | `swiper` | Префикс имени события для всех событий DOM, генерируемых элементом Swiper (веб-компонентом) |
| [fadeEffect](https://swiperjs.com/swiper-api#param-fadeEffect) | any |  | Объект с параметрами эффекта затухания  const swiper = new Swiper('.swiper', {  effect: 'fade',  fadeEffect: {  crossFade: true  },  }); |
| [flipEffect](https://swiperjs.com/swiper-api#param-flipEffect) | any |  | Объект с параметрами перевернутого эффекта  const swiper = new Swiper('.swiper', {  effect: 'flip',  flipEffect: {  slideShadows: false,  },  }); |
| [focusableElements](https://swiperjs.com/swiper-api#param-focusableElements) | string | 'input, select, option, textarea, button, video, label' | CSS-селектор для фокусируемых элементов. Пролистывание будет отключено для таких элементов, если они "сфокусированы". |
| [followFinger](https://swiperjs.com/swiper-api#param-followFinger) | boolean | true | Если этот параметр отключен, то слайдер будет анимироваться только тогда, когда вы его отпустите, он не будет двигаться, пока вы держите на нем палец |
| [freeMode](https://swiperjs.com/swiper-api" \l "param-freeMode) | any |  | Включает функциональность свободного режима. Объект со свободными параметрами режима или логическим значением true включить с настройками по умолчанию.  const swiper = new Swiper('.swiper', {  freeMode: true,  });  const swiper = new Swiper('.swiper', {  freeMode: {  enabled: true,  sticky: true,  },  }); |
| [grabCursor](https://swiperjs.com/swiper-api#param-grabCursor) | boolean | false | Эта опция может немного улучшить удобство использования рабочего стола. Если true, пользователь увидит курсор "захватить" при наведении курсора на Swiper |
| [grid](https://swiperjs.com/swiper-api#param-grid) | any |  | Объект с параметрами сетки для включения ползунка "многорядность".  const swiper = new Swiper('.swiper', {  grid: {  rows: 2,  },  }); |
| [hashNavigation](https://swiperjs.com/swiper-api#param-hashNavigation) | any |  | Включает навигацию по хэш-URL для слайдов. Объект с параметрами навигации по хэшу или логическим значением true для включения с настройками по умолчанию  const swiper = new Swiper('.swiper', {  hashNavigation: {  replaceState: true,  },  }); |
| [height](https://swiperjs.com/swiper-api#param-height) | null | number | null | Высота свипера (в пикселях). Параметр позволяет принудительно изменять высоту свипера. Полезно, только если вы инициализируете Swiper, когда он скрыт, а также в SSR и тестовых средах для корректной инициализации Swiper  Установка этого параметра приведет к тому, что Swiper перестанет реагировать |
| [history](https://swiperjs.com/swiper-api#param-history) | any |  | Включает состояние "просмотр истории", в котором каждый слайд будет иметь свой собственный URL-адрес. В этом параметре вы должны указать URL-адрес основных слайдов, например "slides" и укажите URL-адрес каждого слайда, используя data-history атрибут.  Объект с параметрами навигации по истории или логическим значением true для включения с настройками по умолчанию  const swiper = new Swiper('.swiper', {  history: {  replaceState: true,  },  });  <!-- will produce "slides/slide1" url in browser history -->  <div class="swiper-slide" data-history="slide1"></div> |
| [init](https://swiperjs.com/swiper-api#param-init) | boolean | true | Должен ли Swiper инициализироваться автоматически при создании экземпляра. Если отключен, то вам нужно инициализировать его вручную, вызвав swiper.init() |
| [initialSlide](https://swiperjs.com/swiper-api#param-initialSlide) | number | 0 | Порядковый номер начального слайда. |
| [injectStyles](https://swiperjs.com/swiper-api#param-injectStyles) | string[] |  | Добавьте текстовые стили в shadow DOM. Только для использования с элементом Swiper |
| [injectStylesUrls](https://swiperjs.com/swiper-api#param-injectStylesUrls) | string[] |  | Ввод стилей <link>отправляйся в ДОМ теней. Только для использования с элементом Swiper |
| [keyboard](https://swiperjs.com/swiper-api#param-keyboard) | any |  | Позволяет осуществлять навигацию по слайдам с помощью клавиатуры. Объект с параметрами клавиатуры или логическим значением true для включения с настройками по умолчанию  const swiper = new Swiper('.swiper', {  keyboard: {  enabled: true,  onlyInViewport: false,  },  }); |
| [lazyPreloadPrevNext](https://swiperjs.com/swiper-api#param-lazyPreloadPrevNext) | number | 0 | Количество следующих и предыдущих слайдов для предварительной загрузки. Применимо только при использовании отложенной загрузки. |
| [lazyPreloaderClass](https://swiperjs.com/swiper-api#param-lazyPreloaderClass) | string | 'swiper-lazy-preloader' | Имя CSS-класса ленивого предварительного загрузчика |
| [longSwipes](https://swiperjs.com/swiper-api#param-longSwipes) | boolean | true | Установлено значение false если вы хотите отключить длинные пролистывания |
| [longSwipesMs](https://swiperjs.com/swiper-api#param-longSwipesMs) | number | 300 | Минимальная длительность (в мс) для перехода к следующему / предыдущему слайду при длительном пролистывании |
| [longSwipesRatio](https://swiperjs.com/swiper-api#param-longSwipesRatio) | number | 0.5 | Соотношение для запуска свайпа к следующему / предыдущему слайду при длительных свайпах |
| [loop](https://swiperjs.com/swiper-api#param-loop) | boolean | false | Установлено значение true чтобы включить режим непрерывного цикла  Из-за особенностей работы режима цикла (он будет переставлять слайды) общее количество слайдов должно быть:   * больше или равно slidesPerView + slidesPerGroup * даже для slidesPerGroup (или использовать loopAddBlankSlides параметр) * даже для grid.rows (или использовать loopAddBlankSlides параметр) |
| [loopAddBlankSlides](https://swiperjs.com/swiper-api#param-loopAddBlankSlides) | boolean | false | Автоматически добавляет пустые слайды, если вы используете сетку или slidesPerGroup и общее количество слайдов даже не дотягивает до slidesPerGroup или к grid.rows |
| [loopAdditionalSlides](https://swiperjs.com/swiper-api#param-loopAdditionalSlides) | number | 0 | Позволяет увеличить количество закольцованных слайдов |
| [loopPreventsSliding](https://swiperjs.com/swiper-api#param-loopPreventsSliding) | boolean | true | Если включено, то slideNext / Prev ничего не будет делать, пока слайдер анимируется в режиме цикла |
| [maxBackfaceHiddenSlides](https://swiperjs.com/swiper-api#param-maxBackfaceHiddenSlides) | number | 10 | Если общее количество слайдов меньше указанного здесь значения, то Swiper включит backface-visibility: hidden на элементах слайда, чтобы уменьшить визуальное "мерцание" в Safari.  Не рекомендуется включать его на большом количестве слайдов, так как это снизит производительность |
| [modules](https://swiperjs.com/swiper-api#param-modules) | any[] |  | Array with Swiper modules  import Swiper from 'swiper';  import { Navigation, Pagination } from 'swiper/modules';  const swiper = new Swiper('.swiper', {  modules: [ Navigation, Pagination ],  }); |
| [mousewheel](https://swiperjs.com/swiper-api#param-mousewheel) | any |  | Позволяет перемещаться по слайдам с помощью колесика мыши. Объект с параметрами колеса мыши или логическим значением true для включения с настройками по умолчанию  const swiper = new Swiper('.swiper', {  mousewheel: {  invert: true,  },  }); |
| [navigation](https://swiperjs.com/swiper-api#param-navigation) | any |  | Объект с навигационными параметрами или логическим значением true включить с настройками по умолчанию.  const swiper = new Swiper('.swiper', {  navigation: {  nextEl: '.swiper-button-next',  prevEl: '.swiper-button-prev',  },  }); |
| [nested](https://swiperjs.com/swiper-api#param-nested) | boolean | false | Установлено значение true включите Swiper для правильного перехвата событий касания. Используйте только для свайперов, которые используют то же направление, что и родительский |
| [noSwiping](https://swiperjs.com/swiper-api#param-noSwiping) | boolean | true | Включать / отключать пролистывание элементов, соответствующих классу, указанному в noSwipingClass |
| [noSwipingClass](https://swiperjs.com/swiper-api#param-noSwipingClass) | string | 'swiper-no-swiping' | Указать noSwipingcss-класс элемента 's |
| [noSwipingSelector](https://swiperjs.com/swiper-api#param-noSwipingSelector) | string |  | Может использоваться вместо noSwipingClass чтобы указать элементы, по которым нужно отключить пролистывание. Например 'input' отключит пролистывание на всех входных данных |
| [normalizeSlideIndex](https://swiperjs.com/swiper-api#param-normalizeSlideIndex) | boolean | true | Нормализовать индекс слайда. |
| [observeParents](https://swiperjs.com/swiper-api#param-observeParents) | boolean | false | Установлено значение true если вам также нужно отслеживать мутации для родительских элементов Swiper |
| [observeSlideChildren](https://swiperjs.com/swiper-api#param-observeSlideChildren) | boolean | false | Установлено значение true если вам также нужно понаблюдать за мутациями для дочерних элементов Swiper slide |
| [observer](https://swiperjs.com/swiper-api#param-observer) | boolean | false | Установлено значение true чтобы включить наблюдение за мутациями в Swiper и его элементах. В этом случае Swiper будет обновляться (повторно инициализироваться) каждый раз, если вы измените его стиль (например, скройте / покажите) или измените его дочерние элементы (например, добавьте / удалите слайды) |
| [on](https://swiperjs.com/swiper-api#param-on) | object |  | Регистрировать обработчики событий |
| [onAny](https://swiperjs.com/swiper-api#param-onAny) | function(handler) |  | Добавьте прослушиватель событий, который будет запускаться при всех событиях  const swiper = new Swiper('.swiper', {  onAny(eventName, ...args) {  console.log('Event: ', eventName);  console.log('Event data: ', args);  }  }); |
| [oneWayMovement](https://swiperjs.com/swiper-api#param-oneWayMovement) | boolean | false | При включении слайды будут прокручиваться только вперед (в одну сторону) независимо от направления свайпа |
| [pagination](https://swiperjs.com/swiper-api#param-pagination) | any |  | Объект с параметрами разбивки на страницы или логическим значением true включить с настройками по умолчанию.  const swiper = new Swiper('.swiper', {  pagination: {  el: '.swiper-pagination',  type: 'bullets',  },  }); |
| [parallax](https://swiperjs.com/swiper-api#param-parallax) | any |  | Объект с параметрами параллакса или логическим значением true включить с настройками по умолчанию.  const swiper = new Swiper('.swiper', {  parallax: true,  }); |
| [passiveListeners](https://swiperjs.com/swiper-api#param-passiveListeners) | boolean | true | Пассивные прослушиватели событий будут использоваться по умолчанию там, где это возможно, для повышения производительности прокрутки на мобильных устройствах. Но если вам нужно использовать e.preventDefault если у вас с ним возник конфликт, то вам следует отключить этот параметр |
| [preventClicks](https://swiperjs.com/swiper-api#param-preventClicks) | boolean | true | Установлено значение true для предотвращения случайных нежелательных переходов по ссылкам во время пролистывания |
| [preventClicksPropagation](https://swiperjs.com/swiper-api#param-preventClicksPropagation) | boolean | true | Установлено значение true чтобы остановить распространение событий кликов по ссылкам во время пролистывания |
| [preventInteractionOnTransition](https://swiperjs.com/swiper-api#param-preventInteractionOnTransition) | boolean | false | Если он включен, он не позволит изменять слайды, проводя пальцем по экрану или нажимая кнопки навигации / разбивки на страницы во время перехода |
| [resistance](https://swiperjs.com/swiper-api#param-resistance) | boolean | true | Установлено значение false если вы хотите отключить устойчивые границы |
| [resistanceRatio](https://swiperjs.com/swiper-api#param-resistanceRatio) | number | 0.85 | Эта опция позволяет вам контролировать коэффициент сопротивления |
| [resizeObserver](https://swiperjs.com/swiper-api#param-resizeObserver) | boolean | true | Когда он включен, он будет использовать ResizeObserver (если поддерживается браузером) в контейнере swiper для определения размера контейнера (вместо наблюдения за изменением размера окна) |
| [rewind](https://swiperjs.com/swiper-api#param-rewind) | boolean | false | Установлено значение true чтобы включить режим "перемотки назад". Когда функция включена, нажмите навигационную кнопку "Далее" (или вызовите .slideNext()) при включении последнего слайда произойдет возврат к первому слайду. Нажатие навигационной кнопки "предыдущая" (или вызов .slidePrev()) при включении первого слайда произойдет переход к последнему слайду.  Не следует использовать вместе с loop режим |
| [roundLengths](https://swiperjs.com/swiper-api#param-roundLengths) | boolean | false | Установлено значение true округлить значения ширины и высоты слайдов, чтобы предотвратить размытость текста на экранах обычного разрешения (если у вас есть такой). |
| [runCallbacksOnInit](https://swiperjs.com/swiper-api#param-runCallbacksOnInit) | boolean | true | Запускает события перехода / смены слайда / Запуска / Завершения при инициализации swiper. Такие события будут запускаться при инициализации в случае, если ваш initialSlide не равен 0, или вы используете режим цикла |
| [scrollbar](https://swiperjs.com/swiper-api#param-scrollbar) | any |  | Объект с параметрами полосы прокрутки или логическим значением true включить с настройками по умолчанию.  const swiper = new Swiper('.swiper', {  scrollbar: {  el: '.swiper-scrollbar',  draggable: true,  },  }); |
| [setWrapperSize](https://swiperjs.com/swiper-api#param-setWrapperSize) | boolean | false | Включите эту опцию, и плагин установит ширину / высоту на оболочке swiper равной общему размеру всех слайдов. В основном следует использовать в качестве запасного варианта совместимости для браузеров, которые плохо поддерживают макет flexbox |
| [shortSwipes](https://swiperjs.com/swiper-api#param-shortSwipes) | boolean | true | Установлено значение false если вы хотите отключить короткие свайпы |
| [simulateTouch](https://swiperjs.com/swiper-api#param-simulateTouch) | boolean | true | Если true, Swiper будет принимать события мыши, такие как события касания (щелчок и перетаскивание для изменения слайдов). |
| [slideActiveClass](https://swiperjs.com/swiper-api#param-slideActiveClass) | string | 'swiper-slide-active' | Имя класса CSS текущего активного слайда  Изменяя классы, вам также нужно будет изменить CSS Swiper, чтобы отразить измененные классы  Не поддерживается в компонентах Swiper React / Vue |
| [slideBlankClass](https://swiperjs.com/swiper-api#param-slideBlankClass) | string | 'swiper-slide-blank' | Имя CSS-класса пустого слайда, добавленного в режиме цикла (когда loopAddBlankSlides включен)  Не поддерживается в Swiper React / Vue |
| [slideClass](https://swiperjs.com/swiper-api#param-slideClass) | string | 'swiper-slide' | Имя класса CSS для слайда  Изменяя классы, вам также нужно будет изменить CSS Swiper, чтобы отразить измененные классы  Не поддерживается в компонентах Swiper React / Vue |
| [slideFullyVisibleClass](https://swiperjs.com/swiper-api#param-slideFullyVisibleClass) | string | 'swiper-slide-fully-visible' | Имя класса CSS для полностью (когда весь слайд находится в окне просмотра) видимого слайда  Не поддерживается в Swiper React / Vue |
| [slideNextClass](https://swiperjs.com/swiper-api#param-slideNextClass) | string | 'swiper-slide-next' | CSS-имя класса slide, которое находится сразу после текущего активного слайда  Изменяя классы, вам также нужно будет изменить CSS Swiper, чтобы отразить измененные классы  Не поддерживается в Swiper React / Vue |
| [slidePrevClass](https://swiperjs.com/swiper-api#param-slidePrevClass) | string | 'swiper-slide-prev' | CSS-класс - имя слайда, который находится непосредственно перед текущим активным слайдом  Изменяя классы, вам также нужно будет изменить CSS Swiper, чтобы отразить измененные классы  Не поддерживается в Swiper React / Vue |
| [slideToClickedSlide](https://swiperjs.com/swiper-api#param-slideToClickedSlide) | boolean | false | Установлено значение true и щелчок по любому слайду произведет переход к этому слайду |
| [slideVisibleClass](https://swiperjs.com/swiper-api#param-slideVisibleClass) | string | 'swiper-slide-visible' | Имя класса CSS текущего / частично видимого слайда  Изменяя классы, вам также нужно будет изменить CSS Swiper, чтобы отразить измененные классы  Не поддерживается в Swiper React / Vue |
| [slidesOffsetAfter](https://swiperjs.com/swiper-api#param-slidesOffsetAfter) | number | 0 | Добавьте (в пикселях) дополнительное смещение слайда в конце контейнера (после всех слайдов) |
| [slidesOffsetBefore](https://swiperjs.com/swiper-api#param-slidesOffsetBefore) | number | 0 | Добавьте (в пикселях) дополнительное смещение слайда в начале контейнера (перед всеми слайдами) |
| [slidesPerGroup](https://swiperjs.com/swiper-api#param-slidesPerGroup) | number | 1 | Задайте количество слайдов, чтобы определить и включить групповое скольжение. Полезно использовать с slidesPerView > 1 |
| [slidesPerGroupAuto](https://swiperjs.com/swiper-api#param-slidesPerGroupAuto) | boolean | false | Этот параметр предназначен для использования только с slidesPerView: 'auto' и slidesPerGroup: 1. Когда он включен, он будет пропускать все слайды, видимые на .slideNext() & .slidePrev() вызовы методов, щелчки по навигационным "кнопкам" и в автозапуске. |
| [slidesPerGroupSkip](https://swiperjs.com/swiper-api#param-slidesPerGroupSkip) | number | 0 | Параметр работает следующим образом: Если slidesPerGroupSkip равно 0 (по умолчанию) из группировки не исключаются слайды, и результирующее поведение остается таким же, как и без этого изменения.  Если slidesPerGroupSkip равно или больше, чем 1 первые X слайдов рассматриваются как отдельные группы, в то время как все последующие слайды сгруппированы по slidesPerGroup ценность. |
| [slidesPerView](https://swiperjs.com/swiper-api#param-slidesPerView) | number | 'auto' | 1 | Количество слайдов в просмотре (слайды видны одновременно в контейнере слайдера).  slidesPerView: 'auto' в настоящее время несовместим с многорядным режимом, когда grid.rows > 1 |
| [spaceBetween](https://swiperjs.com/swiper-api#param-spaceBetween) | string | number | 0 | Расстояние между слайдами в пикселях.  Если вы используете css-свойство "margin" для элементов, которые входят в Swiper, в который вы передаете "Пробел между", навигация может работать неправильно. |
| [speed](https://swiperjs.com/swiper-api#param-speed) | number | 300 | Длительность перехода между слайдами (в мс) |
| [swipeHandler](https://swiperjs.com/swiper-api#param-swipeHandler) | any | null | Строка с CSS-селектором или HTML-элементом контейнера с разбиением на страницы, который будет работать как единственный доступный обработчик для пролистывания |
| [threshold](https://swiperjs.com/swiper-api#param-threshold) | number | 5 | Пороговое значение в пикселях. Если "расстояние касания" будет меньше этого значения, то swiper не будет двигаться |
| [thumbs](https://swiperjs.com/swiper-api#param-thumbs) | any |  | Объект с параметрами компонента thumbs  const swiper = new Swiper('.swiper', {  ...  thumbs: {  swiper: thumbsSwiper  }  }); |
| [touchAngle](https://swiperjs.com/swiper-api#param-touchAngle) | number | 45 | Допустимый угол (в градусах) для запуска сенсорного перемещения |
| [touchEventsTarget](https://swiperjs.com/swiper-api#param-touchEventsTarget) | 'container' | 'wrapper' | 'wrapper' | Целевой элемент для прослушивания сенсорных событий. Может быть 'container' (для прослушивания событий касания в swiper) или 'wrapper'(для прослушивания событий касания в swiper-оболочке) |
| [touchMoveStopPropagation](https://swiperjs.com/swiper-api#param-touchMoveStopPropagation) | boolean | false | Если включено, то распространение "touchmove" будет остановлено |
| [touchRatio](https://swiperjs.com/swiper-api#param-touchRatio) | number | 1 | Коэффициент соприкосновения |
| [touchReleaseOnEdges](https://swiperjs.com/swiper-api#param-touchReleaseOnEdges) | boolean | false | Включите активацию событий касания в положении края ползунка (начало, конец), чтобы обеспечить дальнейшую прокрутку страницы. Эта функция работает только с событиями "касания" (а не с событиями указателя), поэтому она будет работать на устройствах iOS / Android и не будет работать на устройствах Windows с событиями указателя. Также threshold параметр должен быть установлен в значение 0 |
| [touchStartForcePreventDefault](https://swiperjs.com/swiper-api#param-touchStartForcePreventDefault) | boolean | false | Принудительно всегда предотвращать использование по умолчанию для touchstart (pointerdown) |
| [touchStartPreventDefault](https://swiperjs.com/swiper-api#param-touchStartPreventDefault) | boolean | true | Если отключен, pointerdown событие не будет предотвращено |
| [uniqueNavElements](https://swiperjs.com/swiper-api#param-uniqueNavElements) | boolean | true | Если включено (по умолчанию), и параметры элементов навигации передаются в виде строки (например ".pagination") затем Swiper сначала будет искать такие элементы через дочерние элементы. Применяется для разбивки на страницы, кнопок "До" / "Далее" и элементов полосы прокрутки |
| [updateOnWindowResize](https://swiperjs.com/swiper-api#param-updateOnWindowResize) | boolean | true | Swiper пересчитает положение слайдов при изменении размера окна (orientationchange) |
| [url](https://swiperjs.com/swiper-api#param-url) | null | string | null | Требуется для активного обнаружения слайдов при рендеринге на стороне сервера и включенной истории |
| [userAgent](https://swiperjs.com/swiper-api#param-userAgent) | null | string | null | Строка пользовательского агента. Требуется для обнаружения браузера / устройства при визуализации на стороне сервера |
| [virtual](https://swiperjs.com/swiper-api#param-virtual) | any |  | Включает функциональность виртуальных слайдов. Объект с параметрами виртуальных слайдов или логическим значением true включить с настройками по умолчанию.  const swiper = new Swiper('.swiper', {  virtual: {  slides: ['Slide 1', 'Slide 2', 'Slide 3', 'Slide 4', 'Slide 5'],  },  }); |
| [virtualTranslate](https://swiperjs.com/swiper-api#param-virtualTranslate) | boolean | false | Включите эту опцию, и swiper будет работать как обычно, за исключением того, что он не будет перемещаться, реальные значения translate для wrapper не будут установлены. Полезно, когда вам может понадобиться создать пользовательский переход к слайду |
| [watchOverflow](https://swiperjs.com/swiper-api#param-watchOverflow) | boolean | true | При включении Swiper будет отключен и скроет кнопки навигации на случай, если слайдов недостаточно для скольжения. |
| [watchSlidesProgress](https://swiperjs.com/swiper-api#param-watchSlidesProgress) | boolean | false | Включите эту функцию для расчета хода выполнения каждого слайда и его видимости (слайды в окне просмотра будут иметь дополнительный класс visible). |
| [width](https://swiperjs.com/swiper-api#param-width) | null | number | null | Ширина свипера (в пикселях). Параметр позволяет принудительно изменять ширину свипера. Полезно, только если вы инициализируете Swiper, когда он скрыт, а также в SSR и тестовых средах для корректной инициализации Swiper  Установка этого параметра приведет к тому, что Swiper перестанет реагировать |
| [wrapperClass](https://swiperjs.com/swiper-api#param-wrapperClass) | string | 'swiper-wrapper' | Имя CSS-класса оболочки слайдов  Изменяя классы, вам также нужно будет изменить CSS Swiper, чтобы отразить измененные классы  Не поддерживается в Swiper React / Vue |
| [zoom](https://swiperjs.com/swiper-api#param-zoom) | any |  | Включает функцию масштабирования. Объект с параметрами масштабирования или логическим значением true для включения с настройками по умолчанию  const swiper = new Swiper('.swiper', {  zoom: {  maxRatio: 5,  },  }); |