IScroll是移动页面上被使用的一款仿系统滚动插件。IScroll5相对于之前的IScroll4改进了许多，使得大家可以更方便的定制所需的功能了。

做项目的时候正好用到了这个插件，自己做了一下总结，发在这里方便大家学习IScroll5。

官网：<http://iscrolljs.com/>

IScroll最简单的DOM结构:

1. <div id="wrapper">
2. <ul>
3. <li>...</li>
4. <li>...</li>
5. ...
6. </ul>
7. </div>

初始化iscroll

1. **var myScroll = new IScroll('#wrapper',options);**

初始化设置

初始化设置使用实例：

1. **var myScroll = new IScroll('#wrapper', {**
2. mouseWheel: **true,**
3. scrollbars: **true**
4. });

设置列表：

|  |  |  |  |
| --- | --- | --- | --- |
| 所属 | 属性名 | 说明 | 默认值 |
| 核心库  croe | options.useTransform | 是否使用CSS3的Transform属性 | true |
| options.useTransition | 是否使用CSS3的Transition属性，否则使用requestAnimationFram代替 | true |
| options.HWCompositing | 是否启用硬件加速 | true |
| options.bounce | 是否启用弹力动画效果，关掉可以加速 | true |
| 基础特性  Basic features | options.click | 是否启用click事件。建议关闭此选项并启用自定义的tap事件（options.tap） | false |
| options.disableMouse | 是否关闭鼠标事件探测。如知道运行在哪个平台，可以开启它来加速。 | false |
| options.disablePointer | 是否关闭指针事件探测。如知道运行在哪个平台，可以开启它来加速。 | false |
| options.disableTouch | 是否关闭触摸事件探测。如知道运行在哪个平台，可以开启它来加速。 | false |
| options.eventPassthrough | 使用IScroll的横轴滚动时，如想使用系统立轴滚动并在横轴上生效，请开启。  [event passthrough demo](http://lab.cubiq.org/iscroll5/demos/event-passthrough/) | false |
| options.freeScroll | 主要在上下左右滚动都生效时使用，可以向任意方向滚动。  [2D scroll demo](http://lab.cubiq.org/iscroll5/demos/2d-scroll/) | false |
| options.keyBindings | 绑定按键事件。  [Key bindings](http://iscrolljs.com/#key-bindings) | false |
| options.invertWheelDirection | 反向鼠标滚轮。 | false |
| options.momentum | 是否开启动量动画，关闭可以提升效率。 | true |
| options.mouseWheel | 是否监听鼠标滚轮事件。 | false |
| options.preventDefault | 是否屏蔽默认事件。 | true |
| options.scrollbars | 是否显示默认滚动条 | false |
| options.scrollX  options.scrollY | 可以设置是否支持横向/纵向滑动 | scrollX  false  scrollY  true |
| options.tap | 是否启用自定义的tap事件  可以自定义tap事件名 | false |
| 滚动条  Scrollbars | options.scrollbars | 是否显示默认滚动条 | false |
| options.fadeScrollbars | 是否渐隐滚动条，关掉可以加速 | true |
| options.interactiveScrollbars | 用户是否可以拖动滚动条 | false |
| options.resizeScrollbars | 是否固定滚动条大小，建议自定义滚动条时可开启。 | false |
| options.shrinkScrollbars | 滚动超出滚动边界时，是否收缩滚动条。  ‘clip’：裁剪超出的滚动条  ‘scale’:按比例的收缩滚动条（占用CPU资源）  false:不收缩， | false |
|  | options.indicators | 指示IScroll该如何滚动，Scrollbars的底层实现方式。 |  |
| options.indicators.el | 制定滚动条的容器。容器中的第一个元素即为指示器。  例如：  indicators: {      el: document.getElementById('indicator')  }  indicators: {      el: '#indicator'  } |  |
| options.indicators.ignoreBoundaries | 是否忽略容器边界。设为true 可以设置滚动速度  [parallax demo](http://lab.cubiq.org/iscroll5/demos/parallax/) | false |
| options.indicators.listenX  options.indicators.listenY | 指示器监听那个方向的滚动，可以设置为一个方向或2个方向 | true |
| options.indicators.speedRatioX  options.indicators.speedRatioY | 指示器相对主滚动条的速度 | 0 |
| options.indicators.fade  options.indicators.interactive  options.indicators.resize  options.indicators.shrink | 如scrollbars的设置  [minimap demo](http://lab.cubiq.org/iscroll5/demos/minimap/) |  |
| options.probeType | 需要使用iscroll-probe.js才能生效  probeType：1  滚动不繁忙的时候触发  probeType：2  滚动时每隔一定时间触发  probeType：3  每滚动一像素触发一次 |  |
| 分割页面snap | options.snap | 自动分割容器，用于制作走马灯效果等。  Options.snap:true//根据容器尺寸自动分割  Options.snap:el//根据元素分割 | false |
| 缩放  zoom | options.zoom | 是否打开缩放  最好使用iscroll-zoom.js  如放大模糊，可将源容器定义为2倍大小，然后scale(0.5)  [zoom demo](http://lab.cubiq.org/iscroll5/demos/zoom/) | false |
| options.zoomMax | 最大缩放等级 | 4 |
| options.zoomMin | 最小缩放等级 | 1 |
| options.startZoom | 初始缩放等级 | 1 |
| options.wheelAction | 滚轮动作  设为’zoom’，可以用滚轮缩放 | undefined |
| 更多设置 | options.bindToWrapper | 光标、触摸超出容器时，是否停止滚动 | false |
| options.bounceEasing | 弹力动画效果  预置效果：'quadratic', 'circular', 'back', 'bounce', 'elastic'（最后两个不能通过css3表现）  还可以自定义效果  bounceEasing: {      style: 'cubic-bezier(0,0,1,1)',//css3时      fn: function (k) { return k; }//不使用css3，使用requestAnimationFrame时  } | 'circular' |
| options.bounceTime | 弹力动画持续的毫秒数 | 600 |
| options.deceleration | 滚动动量减速  越大越快，建议不大于0.01 | 0.0006 |
| options.mouseWheelSpeed | 鼠标滚轮速度 |  |
| options.preventDefaultException | 列出哪些元素不屏蔽默认事件； | {tagName: /^(INPUT|TEXTAREA|BUTTON|SELECT)$/ } |
| options.resizePolling | 重新调整窗口大小时，重新计算IScroll的时间间隔 | 60 |
| 键位绑定 | options.keyBindings | 监听按键事件控制IScroll  例如：  keyBindings: {      pageUp: 33,      pageDown: 34,      end: 35,      home: 36,      left: 37,      up: 38,      right: 39,      down: 40  } |  |
|  |  |  |  |

IScroll5的API：

|  |  |  |
| --- | --- | --- |
| 所属 | 方法名 | 说明 |
| 滚动 | scrollTo(x, y, time, easing) | 滚动到：x，y，事件，easing方式  x:int  y:int  time:int  Easing: quadratic | circular | back | bounce | elastic  见IScroll.utils.ease 对象  例：  myScroll.scrollTo(0, -100, 1000, IScroll.utils.ease.elastic); |
|  | scrollBy(x, y, time, easing) | 滚动到相对于当前位置的某处  其余同上 |
|  | scrollToElement(el, time, offsetX, offsetY, easing) | 滚动到某个元素。el为必须的参数  offsetX/offsetY：相对于el元素的位移。设为true即为屏幕中心  [scroll to element](http://lab.cubiq.org/iscroll5/demos/scroll-to-element/) |
| 分割页面snap | goToPage(x, y, time, easing) | 根据options.snap分割页面，跳转到横向、纵向某页。XY可以同时生效。  结合options.snap使用 |
| next()  prev() | 上一页，下一页  结合options.snap使用 |
| 缩放 | zoom(scale, x, y, time) | 缩放容器  Scale:缩放因子 |
| 刷新 | refresh() | 刷新IScroll |
| 销毁 | destroy() | 销毁IScroll，节省资源 |
|  |  |  |

IScroll的事件

事件使用实例：

1. myScroll = **new IScroll('#wrapper');**
2. myScroll.on('scrollEnd', doSomething);

|  |  |
| --- | --- |
| beforeScrollStart | 用户点击屏幕，但是还未初始化滚动前 |
| scrollCancel | 初始化滚动后又取消 |
| scrollStart | 开始滚动 |
| scroll | 滚动中 |
| scrollEnd | 滚动结束 |
| flick | 轻击屏幕左、右 |
| zoomStart | 开始缩放 |
| zoomEnd | 缩放结束 |

IScroll的属性

|  |  |
| --- | --- |
| myScroll.x/y | 当前位置 |
| myScroll.directionX/Y | 上一次的滚动方向(-1 下/右, 0 保持原状, 1 上/左) |
| myScroll.currentPage | 当前Snap信息 |
| myScroll.maxScrollXmyScroll.maxScrollY | 当滚动到底部时的 myScroll.x/y |