

# AlloyTouch-触摸一切

---

dntzhang(张磊)

# 简介

---



[触摸运动解决方案AlloyTouch](#)

[手势解决方案AlloyFinger](#)

[开放现代的Web组件框架Omi](#)

[Markdown转网站利器md2site](#)

[omi-touch: Omi+AlloyTouch](#)



Tencent AlloyTeam

# 移动开发三利器



transformjs

移动交互特效解决方案



AlloyTouch

加速到减速回弹运动解决方案



AlloyFinger

触屏设备手势解决方案


- transform.js
- AlloyTouch
- omi与omi-touch
- 案例

transformjs

## 简单直接地操作CSS3 Transform

# transform

<code>none</code>
<code>matrix(<i>n,n,n,n,n,n</i>)</code>
<code>matrix3d(<i>n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n</i>)</code>
<code>translate(<i>x,y</i>)</code>
<code>translate3d(<i>x,y,z</i>)</code>
<code>translateX(<i>x</i>)</code>
<code>translateY(<i>y</i>)</code>
<code>translateZ(<i>z</i>)</code>
<code>scale(<i>x,y</i>)</code>
<code>scale3d(<i>x,y,z</i>)</code>
<code>scaleX(<i>x</i>)</code>
<code>scaleY(<i>y</i>)</code>
<code>scaleZ(<i>z</i>)</code>
<code>rotate(<i>angle</i>)</code>
<code>rotate3d(<i>x,y,z,angle</i>)</code>
<code>rotateX(<i>angle</i>)</code>
<code>rotateY(<i>angle</i>)</code>
<code>rotateZ(<i>angle</i>)</code>
<code>skew(<i>x-angle,y-angle</i>)</code>
<code>skewX(<i>angle</i>)</code>
<code>skewY(<i>angle</i>)</code>
<code>perspective(<i>n</i>)</code>

```
element.style {  
    transition: ▶ transform 1s  ease-in;  
    transform: translateY(100px);  
}
```



# animation+transform

```
@keyframes bounceOut {  
  20% {  
    transform: scale3d(.9, .9, .9);  
  }  
  
  50%, 55% {  
    opacity: 1;  
    transform: scale3d(1.1, 1.1, 1.1);  
  }  
  
  to {  
    opacity: 0;  
    transform: scale3d(.3, .3, .3);  
  }  
}  
  
.bounceOut {  
  animation-name: bounceOut;  
}
```

```
$("#some_element").animate({  
  opacity: 0.25, left: '50px',  
  color: '#abcdef',  
  rotateZ: '45deg', translate3d: '0,10px,0'  
}, 500, 'ease-out')
```

```
var tween = new TWEEN.Tween( { x: 50, y: 0 } )
    .to( { x: 400 }, 2000 )
    .easing( TWEEN.Easing.Elastic.InOut )
    .onUpdate( function () {
        output.innerHTML = 'x == ' + Math.round( this.x );
        var transform = 'translateX(' + this.x + 'px)';
        output.style.webkitTransform = transform;
        output.style.transform = transform;
    } )
    .start();
```

```
transform: translateX(100px) scaleX(0.5);
```

```
transform: scaleX(0.5) translateX(100px);
```

# transform顺序

```
new WebKitCSSMatrix("translateX(100px) scaleX(0.5)")
```

```
new WebKitCSSMatrix("scaleX(0.5) translateX(100px)")
```

```
m11: 0.5  
m12: 0  
m13: 0  
m14: 0  
m21: 0  
m22: 1  
m23: 0  
m24: 0  
m31: 0  
m32: 0  
m33: 1  
m34: 0  
m41: 100  
m42: 0  
m43: 0  
m44: 1
```



```
m11: 0.5  
m12: 0  
m13: 0  
m14: 0  
m21: 0  
m22: 1  
m23: 0  
m24: 0  
m31: 0  
m32: 0  
m33: 1  
m34: 0  
m41: 50  
m42: 0  
m43: 0  
m44: 1
```

# transform-origin

```
transform-origin: x-axis y-axis z-axis;
```

值	描述
x-axis	定义视图被置于 X 轴的何处。可能的值： <ul style="list-style-type: none"><li>• left</li><li>• center</li><li>• right</li><li>• <i>length</i></li><li>• %</li></ul>
y-axis	定义视图被置于 Y 轴的何处。可能的值： <ul style="list-style-type: none"><li>• top</li><li>• center</li><li>• bottom</li><li>• <i>length</i></li><li>• %</li></ul>
z-axis	定义视图被置于 Z 轴的何处。可能的值： <ul style="list-style-type: none"><li>• <i>length</i></li></ul>

不直观:顺序影响结果

不直接:无法step递进递减控制

不方便:控制旋转点需要设置transform-origin

## API : Transform ( dom )

```
dom.translateX = 100 ;  
dom.scaleX= 0.5 ;  
dom.rotateZ= 30 ;  
dom.originX= 50 ;  
dom.originY= 50 ;
```



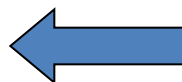
Object.defineProperty ( dom )



translateX, translateY, translateZ,  
scaleX, scaleY, scaleZ,  
rotateX, rotateY, rotateZ,  
originX, originY, originZ



element.style.transform



set的callback计算matrix3d

# AlloyTouch

## 触摸反馈到任意属性的运动

alloy\_touch.js      requestAnimationFrame + 缓动函数

alloy\_touch.css.js      CSS3 transition (用transition-timing-function设置速率变换)

## iScroll

**iscroll-probe.js**

2149行代码

**iscroll-probe.min.js**

6.54 kB gzipped

## AlloyTouch

**alloy\_touch.js**

270行代码

**alloy\_touch.min.js**

小于1k gzipped

# 功能对比

功能特性	AlloyTouch	IScroll
滚动过程callback	✓	✓ 默认版本不支持
Canvas/WebGL/SVG	✓	✗
惯性、摩擦、回弹	✓	✓
无限滚动能力	✓	✗
旋转、放大等任意属性运动	✓	✗
智能默认事件阻止	✓	✓
分区滚动校正	✓	✓
惯性回弹灵敏度摩擦可配置	✓	✗

# AlloyTouch属性无关设计



WgwbD<>

28岁 抚州

关注 7 | 粉丝 24



我的部落





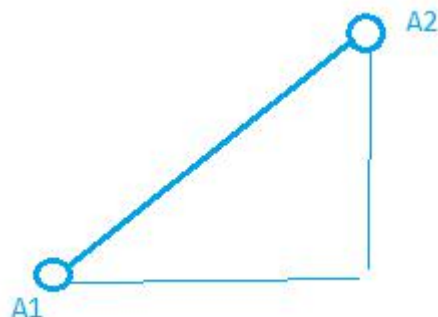


高级运动：力、冲量

中级运动：速度、加速度

低级运动：直接改位置

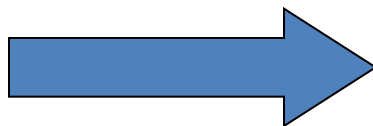
低级运动：直接改位置



touchend坐标与最后一次touchmove的位移和时间间隔计算**终点**和**时间**传给tween

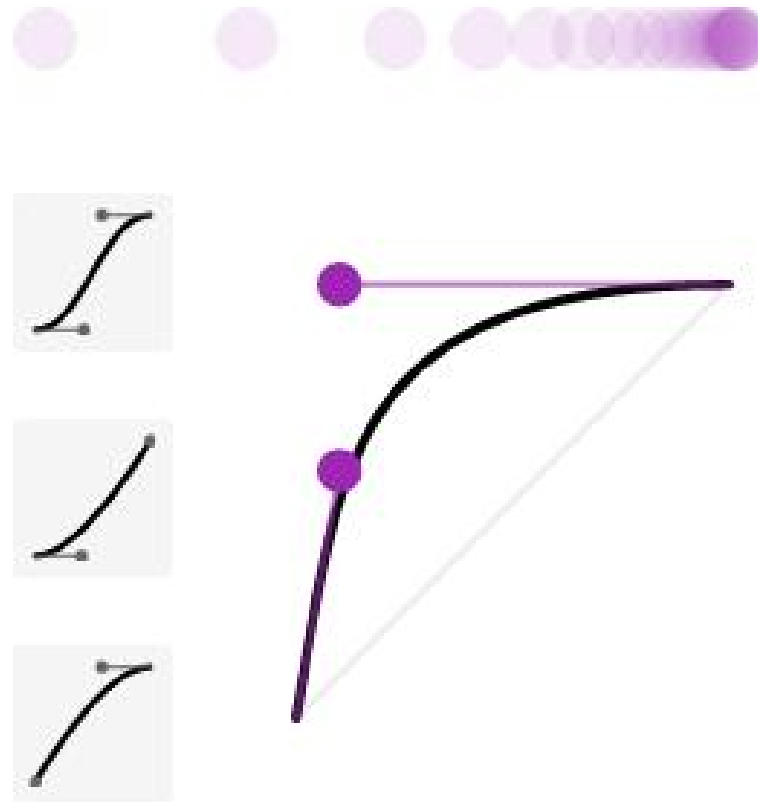
$$S = vt + \frac{1}{2}at^2$$

$$t = \frac{|v|}{|a|}$$



$$S = \frac{v^2}{2 * a}$$

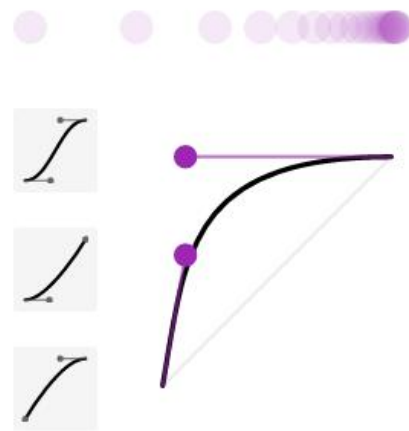
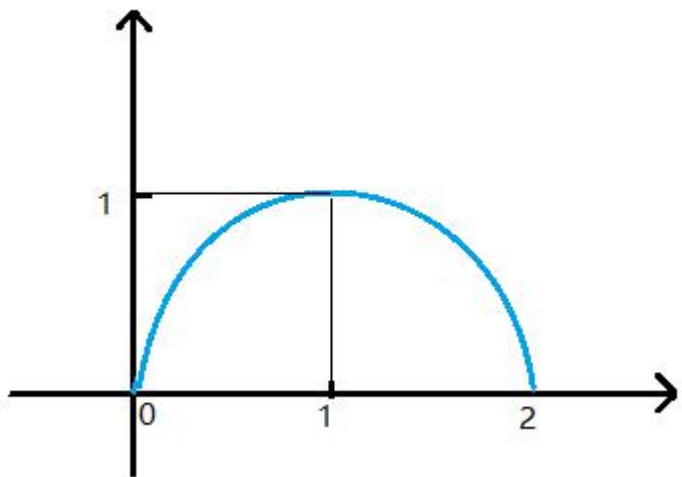
# AlloyTouch 摩擦、惯性、回弹



```
cubic-bezier(0.1, 0.57, 0.1, 1)
```

# AlloyTouch 摩擦、惯性、回弹

$$y = \sqrt{1 - (x - 1)^2}$$



cubic-bezier(0.1, 0.57, 0.1, 1)

```
function animationEase(x) {  
    return Math.sqrt(1 - Math.pow(x - 1, 2));  
}
```

```
new AlloyTouch({
    touch: "#wrapper", //反馈触摸的dom
    vertical: true, //不必需, 默认是true代表监听垂直方向touch
    target: target, //运动的对象
    property: "translateY", //被运动的属性
    sensitivity: 1, //不必需, 触摸区域的灵敏度, 默认值为1, 可以为负数
    factor: 1, //不必需, 表示触摸位移与被运动属性映射关系, 默认值是1
    min: 100, //不必需, 运动属性的最小值
    max: 2000, //不必需, 滚动属性的最大值
    spring: true, //不必需, 是否有回弹效果。默认是true
    change: function() { }, //不必需, 属性改变的回调
    touchStart: function(value) { },
    touchMove: function(value) { },
    touchEnd: function(value) { },
    reboundEnd: function(value) { } //回弹结束
})
```

# omi与omi-touch

# 面向对象的Web组件化框架



# Omi vs React

	React	Omi
组件通信	★★★★☆	★★★★★★
稳定性	★★★★★★	★★★★★☆☆
灵活性	★★★★☆	★★★★★★
扩展性	★★★★☆	★★★★★★
测试性	★★★★★★	★★★★★☆☆
文件大小	★★★☆☆☆	★★★★★★
功能特性	★★★☆☆☆	★★★★★☆☆
DOM性能	★★★★★★	★★★★★☆☆
动画性能	★★★★☆	★★★★★★
抽象复杂度	★★★★☆	★★★★★★
异步编程	★★★★★★	★★★★★☆☆

```
1 class Hello extends Omi.Component {
2   constructor(data) {
3     super(data);
4   }
5   style () {
6     return
7     <style>
8       h1{
9         cursor:pointer;
10      }
11     </style>
12   };
13 }
14 handleClick(target, evt){
15   alert(target.innerHTML);
16 }
17 render() {
18   return
19   <div>
20     <h1 onclick="handleClick(this, event)">Hello , {{name}}!</h1>
21   </div>
22   `;
23 }
24 }
25 }
26
27 Omi.makeHTML('Hello', Hello);
28
29 class App extends Omi.Component {
30   constructor(data) {
31     super(data);
32   }
33
34   render() {
35     return
36     <div>
37       <Hello data-name="Omi" />
38     </div>
39     `;
40   }
41 }
42 }
43
44 Omi.render(new App(), "#container");
```



Omi插件体系可以赋予dom元素一些能力  
并且可以和组件的实例产生关联

```
1 import Omi from 'omi';
2 import OmiDrag from '../..//omi-drag.js';
3
4 OmiDrag.init();
5
6 class App extends Omi.Component {
7   constructor(data) {
8     super(data);
9   }
10
11   moveHandlerA() {
12     console.log('a is moving');
13   }
14
15   moveHandlerB() {
16     console.log('b is moving');
17   }
18
19   render() {
20     return `
21 <div>
22   <div omi-drag class="test" dragMove="moveHandlerA" >Drag Me A</div>
23   <div omi-drag class="test" dragMove="moveHandlerB" >Drag Me B</div>
24 </div>
25 `;
26   }
27
28   style() {
29     return `
30   .test {
31     width:100px;
32     height:100px;
33     color:white;
34     line-height:90px;
35     text-align:center;
36     background-color:#00BFF3;
37   }
38 `;
39   }
40 }
41
42 Omi.render(new App(), "#container");
```



```
1 import Omi from 'omi';
2 import OmiTouch from '../..//omi-touch.js';
3
4 OmiTouch.init();
5
6 class App extends Omi.Component {
7   constructor(data) {
8     super(data);
9   }
10
11   render() {
12     return
13     <div class="main">
14       <div omi-touch class="touchArea" motionRef="scroller" min="-1750" max="0" >
15         <ul ref="scroller">
16           <li>Hello, Omi-Touch!</li>
17           <li>AlloyTouch</li>
18           <li>Transformjs</li>
19           <li>AlloyFinger</li>
20           <li>Omi</li>
21           ...
22           ...
23           <li> row 24</li>
24           <li style="border-bottom: none;"> row 25</li>
25         </ul>
26       </div>
27     </div>
28   ;
29 }
30 }
31
32 Omi.render(new App(), "#container");
```



## 案例

< 返回

QQ天气



IT大咖说  
不止于技术

2015/12/17 星期四

深圳 [切换]

14°

16:05更新



多云

空气质量: 优

湿度: 22%

在 17时 18时 19时 20时



4°

16°

16°

16°

13°

13°



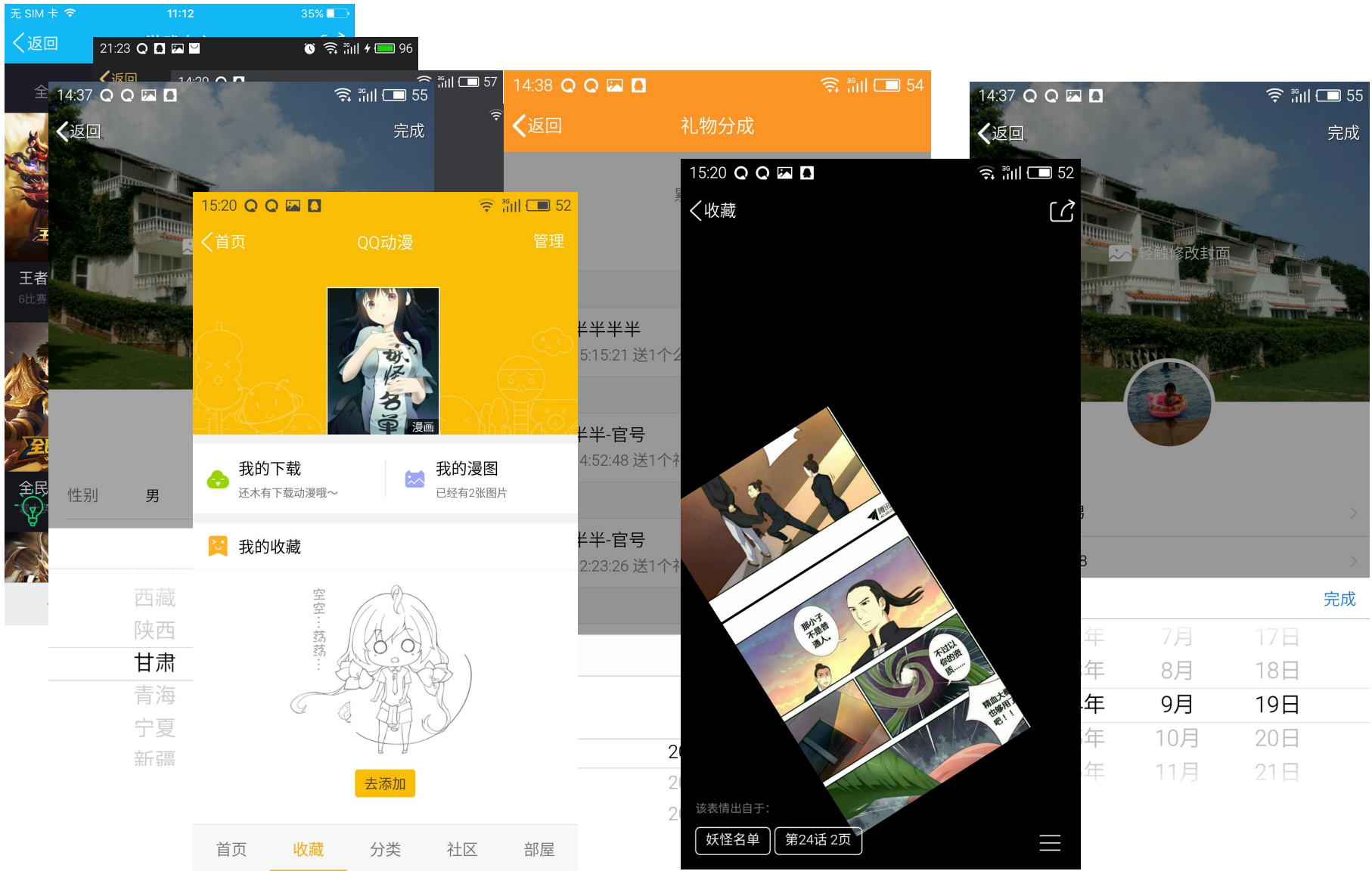
碧玉小耳钉

好看且不贵  
你可以更美





# AlloyTouch、AlloyFinger、transform.js 部分案例



# 完

更多精彩: [AlloyTeam.github.io](http://AlloyTeam.github.io)