

# HarmonyOS 轻量 JS 开发框架与 W3C 标准差异分析

HarmonyOS 轻量 JS 应用开发框架(下文简称“框架”), 是 HarmonyOS 为开发者提供的一套开发 JS 应用的开发框架。开发框架采用类小程序的 web 开发方式, 其实现大部分遵循 W3C 标准(主流 web 开发标准), 但由于设备条件限制(例如, ROM 和 RAM 大小), “框架”中部分组件和属性与 W3C 标准存在差异, 需要开发者在开发过程中了解和掌握。

本文将会梳理当前“框架”已有的组件和属性与 W3C 标准相对应组件和属性的主要差异点, 并配以示例说明, 供开发者在开发过程中参考。主要差异点包括: width 和 height 属性、绝对定位和相对定位、文字显示、页面滑动和长按事件。

## width 和 height 属性

作为一个前端开发者, 在开发网页或者应用的过程中, 我们一般不会手动设置组件的高度和宽度, 而是希望元素的大小根据窗口或者子元素自动调整, 这就是自适应能力。

自适应能力使得布局更灵活, 可适应不同设备、不同窗口和不同分辨率下的显示。

但是, “框架”当前提供的组件除 text 外, 皆不支持内容高度和宽度的自适应能力, 必须由开发者明确指定高度和宽度, 否则组件不会显示。

以一段简单的代码为例:

```
<div style="background-color: red;">  
  
    <div style="width: 100px;height: 100px;background-color: yellow;"> </div>  
  
</div>
```

根组件 div 未设置高宽属性，子组件 div 设置了高宽属性，在“框架”和 W3C 标准中其运行效果图分别如图 1 和图 2 所示：



图 1 “框架”中不设置根组件的高度和宽度显示效果图



图 2 W3C 标准中不设置根组件的高度和宽度显示效果图从图 1 和图 2 可以看出，“框架”中根组件必须设置高度和宽度，否则无法显示，即使根组件的子组件设置了高宽度属性，界面也无法显示。另外由于 W3C 标准的 div 组件是块状元素，如果没有设置宽度，会独占一行，所以在图 2 中，没有设置宽度，其显示范围为一整行。

“框架”中设置根组件的高度和宽度，则正常显示，效果如图 3 所示：

```
<div style="width:200px;height:200px;background-color: red;">  
  
    <div style="width: 100px;height: 100px;background-color: yellow;"></div>
```

来源：HarmonyOS 开发者微信号 <https://mp.weixin.qq.com/s/cAKrVR4iuPh-tFrdx1jg0A>

</div>



图 3 “框架”下根组件设置高度和宽度效果图

针对 text 组件，不设置高宽属性的条件下，“框架”和 W3C 标准下 web 开发界面显示效果一致，如图 4 和图 5 所示：

<text>

Hello World!

</text>



图 4 “框架”下 text 组件的文本显示图



图 5 W3C 标准下 text 组件的文本显示图

## 绝对定位和相对定位

如何实现绝对定位和相对定位？如果你常用 html 语言开发，你可能会首先想到设置 position 属性为 absolute 和 relative，然后通过 left 和 top 设置距离即可实现绝对定位和相对定位的效果，其实现代码如下所示：

```
<html>

<head>

  <style type="text/css">

    h2.left {

      position: relative;

      left: 30px;

      top: 30px;

    }

    h2.right {
```

```

        position: absolute;

        left: 90px;

        top: 160px;

    }

</style>

</head>

<body>

    <h2>这是位于正常位置的标题</h2>

    <h2 class="left">这个标题相对于其正常位置向右向下移动</h2>

    <h2 class="right">这个标题相对于其页面左上角位置向右向下移动</h2>

</body>

</html>

```

但是在“框架”中，我们查看 HarmonyOS 官网 JS API 文档后发现，position 属性是不支持设置 absolute 和 relative，那该如何实现绝对定位和相对定位这两种效果呢？如果我们仔细阅读 HarmonyOS 官网文档，我们会发现“框架”提供了一个新组件 stack，借助这个组件我们可以实现绝对定位的效果，示例代码如下所示，添加一个 stack 根组件（同时也作为父组件），其子组件为 div，然后设置其 left 和 top 的值为 50px 来实现绝对定位的效果，其效果如图 6 所示，子组件 div 相对于父组件 stack 向下、向右各偏移 50px。

```

<stack style="width: 200px; height: 200px; background-color: red;">

    <div

```

```
style="
    left: 50px;
    top: 50px;
    width: 100px;
    height: 100px;
    background-color: yellow;"
>
</div>
</stack>
```



图 6 “框架”中绝对定位实现效果

另外，绝对定位存在一种特殊场景：如果是根组件，直接设置 top 和 left 属性即可实现绝对定位功能；非根组件，其父组件必须是 stack，才能通过设置 top 和 left 属性来实现绝对定位功能。根组件绝对定位是相对于页面左上角做偏移，非根组件绝对定位是相对于父组件做偏移。如下代码所示，根组件直接设置 top 和 left 属性实现绝对定位，由于此时根组件作为父组件不是 stack，所以设置子组件的绝对定位无效，如图 7 中所示，子组件黄色 div

相对于父组件红色 div 未实现绝对定位的效果。

```
<div  
  style="  
    width: 200px;  
    height: 200px;  
    background-color: red;  
    top: 50px;  
    left: 50px;"  
>  
  
<div  
  style="  
    left: 50px;  
    top: 50px;  
    width: 100px;  
    height: 100px;  
    background-color: yellow;"  
>  
  
</div>  
</div>
```



图 7 “框架”中非根组件和根组件绝对定位实现效果图

相对定位：由于设备硬件 ROM 和 RAM 大小限制，当前“框架”不支持 W3C 标准中的相对定位实现（position:relative），只支持部分 flex 布局属性，与 W3C 标准相比差异较大，具体差异见表 1 所示。

样式属性	W3C	框架
display	flex   inline-flex	flex PS：仅允许在div组件上设置。
flex-direction	row   row-reverse   column   column-reverse	row   column
flex-wrap	nowrap   wrap   wrap-reverse	nowrap   warp
flex-flow	<flex-direction>    <flex-wrap>	不支持
justify-content	flex-start   flex-end   center   space-between   space-around	flex-start   flex-end   center   space-between   space-around
align-items	flex-start   flex-end   center   baseline   stretch	flex-start   flex-end   center   stretch
align-content	flex-start   flex-end   center   space-between   space-around   stretch	不支持
order	<number>	不支持
flex-grow	<number>	不支持
flex-shrink	<number>	不支持
flex-basis	<length>   auto	不支持
flex	[<flex-grow>    <flex-shrink>    <flex-basis>]   none	不支持
align-self	auto   flex-start   flex-end   center   baseline   stretch	不支持

表 1 “框架”下 flex 布局可支持属性对比

在“框架”中实现 flex 布局，将需要 flex 布局的子组件用 div 容器包裹一下，并将 div 的 display 属性设置为 flex 即可。如下图所示，我们要实现三个 div 一行排列且居中显示的 flex



布局。

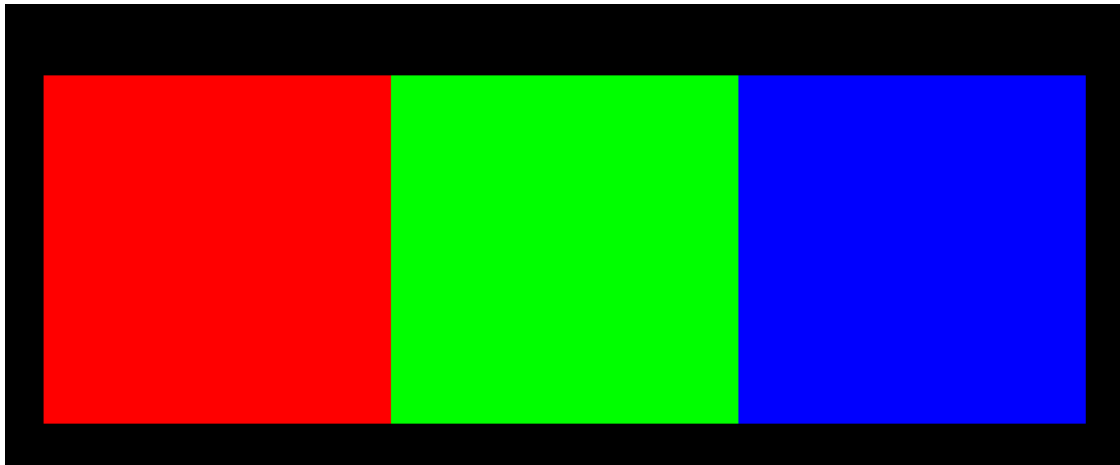


图 8 div flex 布局效果图

在“框架”中我们可以这样实现：添加一个 div 父组件，其子组件按行居中排列，分别设置 justify-content 和 align-items 两个属性值为 center(div 组件的主轴方向默认值是 row，如果想实现纵向排列需要设置 flex-direction: column)即可实现图 8 中的效果，详细代码如下所示：

```
//index.html
```

```
<div class="container">  
  
  <div class="box box--red"> </div>  
  
  <div class="box box--green"> </div>  
  
  <div class="box box--blue"> </div>
```

```
</div>
```

```
//index.css
```

```
.container {  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    width: 100%;  
    height: 100%;  
}  
  
.box {  
    width: 30%;  
    height: 60%;  
}  
  
.box--red {  
    background-color: #f00;  
}  
  
.box--green {  
    background-color: #0f0;  
}  
  
.box--blue {
```

```
background-color: #00f;
}
```

W3C 标准下的 web 开发代码如下所示，和“框架”下 web 开发代码实现基本一致。

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta      name="viewport"      content="width=device-width,
initial-scale=1.0">

  <title>Flex Layout</title>

  <style>

    .container {

      align-items: center;

      background-color: #000;

      display: flex;

      height: 454px;

      justify-content: center;

      width: 454px;
```

```
}

.title {

    text-align: center;

}

.box {

    height: 60%;

    width: 30%;

}

.box--red {

    background-color: #f00;

}

.box--green {

    background-color: #0f0;

}

.box--blue {

    background-color: #00f;

}

</style>

</head>

<body>

    <div class="container">

        <div class="box box--red"> </div>
```

```
<div class="box box--green"></div>

<div class="box box--blue"></div>

</div>

</body>

</html>
```

## 文字显示

在前端开发中，遵循 W3C 标准开发者可以使用多种标签来实现文字显示。如段落 “<p></p>”、标题 “<h1></h1>”，甚至部分标签中直接放入文字也可以显示，如 “<div>标题</div>”。但是基于“框架”开发的应用想要显示文字，必须将文字内容放在 text 组件中，或者设置 input 组件的 value 值，才可以正常显示。如下代码所示，在 div 组件中直接添加文字 “Hello world”，可以看到 W3C 标准下 web 开发界面显示出文字（如图 9 所示），“框架”下 web 开发界面则无法显示文字（如图 10 所示）。

```
<div style="width: 200px; height: 200px; background-color: red;">

    Hello World!

</div>
```

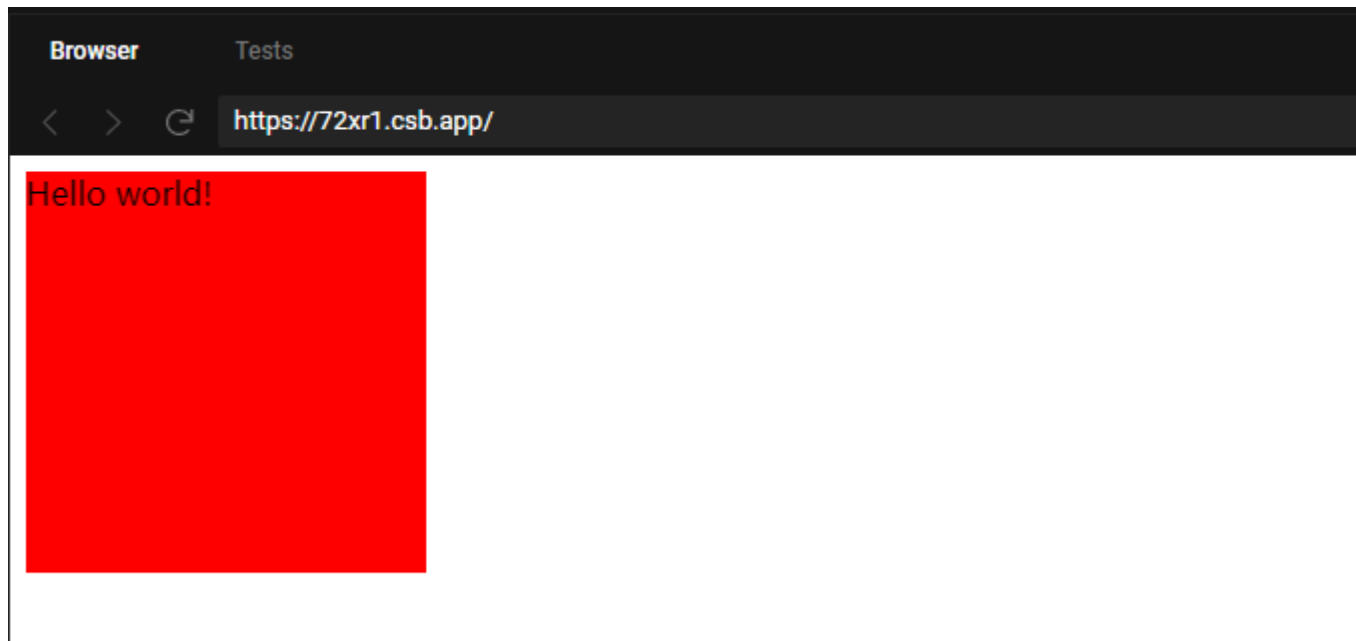


图 9 W3C 标准下 div 中直接添加文字效果图



图 10 “框架”下 div 中直接添加文字效果图

“框架”下正确使用方式是将要显示的文字放在 text 组件中，或者设置 input 组件的 value 值才能正常显示文字。另外，对于智能穿戴设备，只支持 30 和 38 两种字号的字体，字体只支持 HYQiHei-65S；如果对其他大小的字体有需求，暂时只能通过贴图的方式去实现开发者所需的效果。text 组件显示文字的代码如下：

```
<text style="width: 200px; height: 200px; text-align: center;">
```

```
  hello world
```

`</text>`



图 11 “框架”下 text 显示文字效果图

input 组件显示文字的代码如下，通过设置 value 属性显示文字：

```
<input style="width: 200px; height: 50px; text-align: center;" value="hello world">
```

`</input>`

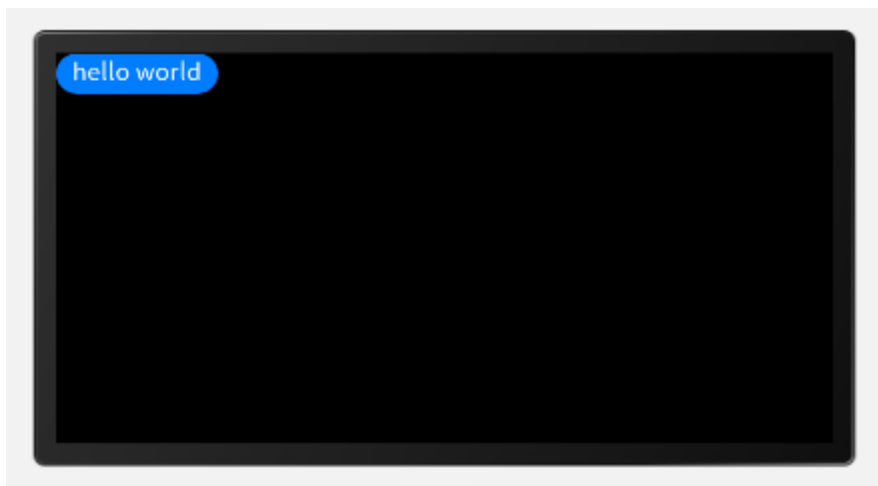


图 12 “框架”下 input 显示文字效果图

## 页面滑动和长按事件

“框架”原生支持页面滑动(swipe)和长按(longpress)事件,其中 swipe 支持“上下左右”四个方向的滑动,开发者通过事件返回值即可获取滑动方向。

如下代码所示,我们添加一个 div 组件和一个 text 组件, text 组件用来显示屏幕上面的滑动方向,事件回调函数 swipeAction 返回相应的属性值,根据回调函数返回值中的 direction 属性,我们可以获取滑动方向,并在 text 组件中显示。

```
// index.html
```

```
<div style="
  width: 200px;
  height: 200px;
  align-items: center;
  justify-content: center;
  border-width: 2px;
  border-color: red;"
  onswipe="swipeAction"
>
  <text style="width: 100px; height: 50px; text-align: center;">{{title}}</text>
</div>
```

```
// index.js
```

```
export default {
  data: {
    title: ""
```



```
},  
  
swipeAction(e){  
    switch(e.direction) {  
        case 'left':  
            this.title = '左滑';  
            break;  
        case 'right':  
            this.title = '右滑';  
            break;  
        case 'up':  
            this.title = '上滑';  
            break;  
        case 'down':  
            this.title = '下滑';  
            break;  
        default:  
            break;  
    }  
}  
}
```

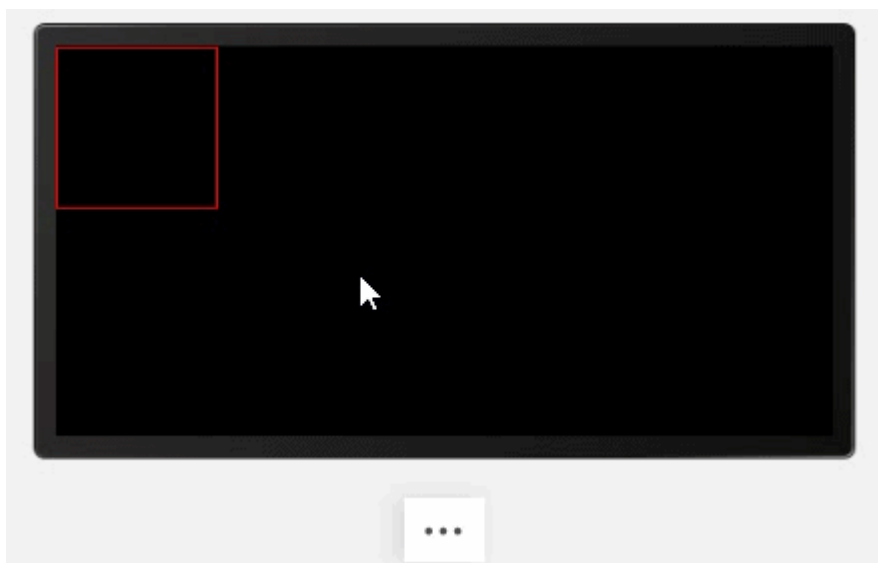


图 13 “框架”下页面滑动效果

除了上面关键差异点外，“框架”和 W3C 标准间还会存在一些组件名称(<image>和<img>等)、属性名称等细节信息差异，建议各位开发者在开发过程中多多参考 HarmonyOS 官网开发文档：

<https://device.harmonyos.com/cn/docs/develop/apiref/js-framework-file-00000000000611396>

HarmonyOS 轻量 JS 应用开发框架还在不断演进和完善的过程中，本文主要描述的是当前 HarmonyOS 轻量 JS 应用开发框架和主流 web 开发的 W3C 标准关键差异点，希望可以帮助到各位开发者上手 HarmonyOS 开发。