自适应响应式

1.通过自适应+响应式的方式让页面无论是在pc pad或者手机端都能实现全屏填充展现

2.重要属性\*{ box-sizing:border-box}

## 3.核心代码：

css部分

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*页面布局\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*布局最后一层存放内容\*/

.grid-layout{ width:100%; height:100%;}

.grid-cell{ width:100%; height:100%; padding:1%;}

.cell-content{width:100%; height:100%; background:#3f99ff;}

/\*横向布局\*/

.row-grid1{height:100%; width:8.33%; float:left; }

.row-grid2{height:100%; width:16.66%; float:left; }

.row-grid3{height:100%; width:25%; float:left; }

.row-grid4{height:100%; width:33.33%; float:left; }

.row-grid5{height:100%; width:41.66%; float:left; }

.row-grid6{height:100%; width:50%; float:left; }

.row-grid7{height:100%; width:58.33%; float:left; }

.row-grid8{height:100%; width:66.66%; float:left; }

.row-grid9{height:100%; width:75%; float:left; }

.row-grid10{height:100%; width:83.33%; float:left; }

.row-grid11{height:100%; width:91.66%; float:left; }

.row-grid12{height:100%; width:100%; float:left; }

/\*纵向布局\*/

.col-grid1{width:100%; height:8.33%; float:left; }

.col-grid2{width:100%; height:16.66%; float:left; }

.col-grid3{width:100%; height:25%; float:left; }

.col-grid4{width:100%; height:33.33%; float:left; }

.col-grid5{width:100%; height:41.66%; float:left; }

.col-grid6{width:100%; height:50%; float:left; }

.col-grid7{width:100%; height:58.33%; float:left; }

.col-grid8{width:100%; height:66.66%; float:left; }

.col-grid9{width:100%; height:75%; float:left; }

.col-grid10{width:100%; height:83.33%; float:left; }

.col-grid11{width:100%; height:91.66%; float:left; }

.col-grid12{width:100%; height:100%; float:left; }

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*页面布局\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*以上布局不够时可自定义，格式参考上面\*/

/\*响应代码---根据页面实际需要做响应\*/

@media only screen and (max-width: 1599px) {}

@media only screen and (max-width: 1365px) {}

@media only screen and (max-width: 1199px) {}

@media only screen and (max-width: 799px) {

.row-grid6{ width:100%;}

}

@media only screen and (max-width: 479px) {}

html结构：（选取不同的布局时，通过调整高亮部分代码即可）

一层布局

<div class="col-grid6 grid-wrap">

<div class="grid-cell">

<div class="cell-content">具体内容，padding已经预留，实际页面开发只需将内容填入</div>

</div>

</div>

布局可嵌套：

<div class="row-grid6 grid-wrap">

<!--纵向-->

<div class="col-grid6 grid-wrap">

<!—此段代码写在最后一层布局里-->

<div class="grid-cell">

<div class="cell-content"></div>

</div>

<!—此段代码写在最后一层布局里-->

</div>

<div class="col-grid6 grid-wrap">

<div class="grid-cell">

<div class="cell-content"></div>

</div>

</div>

<!--纵向-->

</div>

## 4.优点

a.css+html代码量少，操作简单，通俗易懂

b.采用横向+纵向的组合方式，使布局多样化

c.边距padding采用百分比的方式，使块面的间距随不同尺寸屏幕而变化（缺点是块面不能过小）

d.后期代码维护方便，改动量小