

块布局Box

首先创建容器 `div class="box"` ，内部的块元素支持10%-100%区间的大小配置，每5%为一个间隔，10%、15%、20%依此类推至100%（不含95%），其中支持特殊宽度 `.grid33`、`.grid66`

类名	权重	小屏幕 (0-750px)	中等屏幕 (0-1200px)	大屏幕 (>1200px)
.grid	默认	保持设定值	保持设定值	保持设定值
.grid-m	高于.grid	触发设定值	触发设定值	不触发
.grid-s	高于.grid及.grid-m	触发设定值	不触发	不触发

.grid50		.grid50	
.grid33		.grid66	
.grid20	.grid80		
.grid100			

```
<div class="box">
  <div class="grid50">.grid50</div>
  <div class="grid50">.grid50</div>
  <div class="grid33">.grid33</div>
  <div class="grid66">.grid66</div>
  <div class="grid20">.grid20</div>
  <div class="grid80">.grid80</div>
  <div class="grid100">.grid100</div>
</div>
```

没有设定大小值的块元素会自动均分主体空间

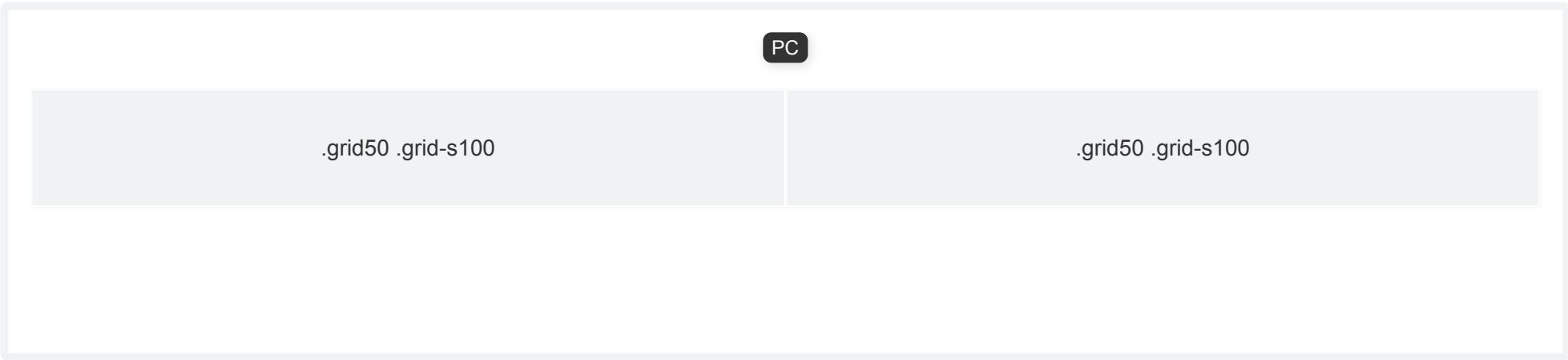
.grid		.grid		
.grid	.grid	.grid40	.grid	.grid

```
<div class="box">
  <div class="grid">.grid</div>
  <div class="grid">.grid</div>
</div>
<div class="box">
  <div class="grid">.grid</div>
  <div class="grid">.grid</div>
  <div class="grid40">.grid40</div>
  <div class="grid">.grid</div>
  <div class="grid">.grid</div>
</div>
```

响应式布局

通过预定义的 `class="grid-m"`、`class="grid-s"` 来对中、小屏幕作布局调整

例：添加 `class="grid50 grid-s100"` ，即块元素默认为50%宽度，视窗像素小于750px则转换为100%宽度



```
<div class="box">
  <div class="grid50 grid-s100">.grid50 .grid-s100</div>
  <div class="grid50 grid-s100">.grid50 .grid-s100</div>
</div>
```

`class="grid-m"`、`class="grid-s"` 可以同时叠加，在不同视窗大小时按权重执行设定值

```
<div class="box">
  <div class="grid50 grid-m75 grid-s100">grid</div>
  <div class="grid50 grid-m75 grid-s100">grid</div>
</div>
```

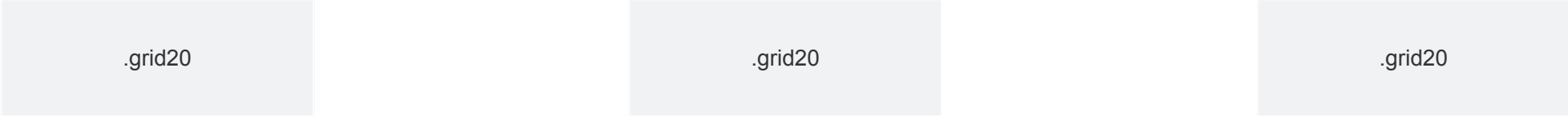
水平对齐

为容器添加 `class="x-start"`、`class="x-center"`、`class="x-end"` 定义块元素的水平对齐方式



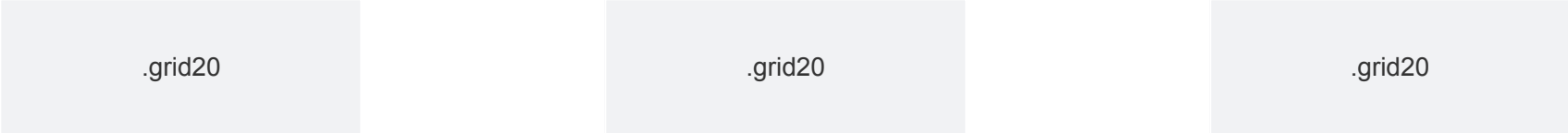
```
<div class="box x-start">
  <div class="grid20">.grid20</div>
  <div class="grid20">.grid20</div>
</div>
<div class="box x-center">
  <div class="grid20">.grid20</div>
  <div class="grid20">.grid20</div>
</div>
<div class="box x-end">
  <div class="grid20">.grid20</div>
  <div class="grid20">.grid20</div>
</div>
```

为容器添加 `class="x-between"` 使块元素两端对齐，块元素之间的间隔都相等



```
<div class="box x-between">
  <div class="grid20">.grid20</div>
  <div class="grid20">.grid20</div>
  <div class="grid20">.grid20</div>
</div>
```

为容器添加 `class="x-around` 使块元素两侧的间隔都相等



```
<div class="box x-around">
  <div class="grid20">.grid20</div>
  <div class="grid20">.grid20</div>
  <div class="grid20">.grid20</div>
</div>
```

垂直对齐

为容器添加 `class="y-start"`、`class="y-center"`、`class="y-end"` 定义块元素的垂直对齐方式



```
<div class="box y-start">
  <div class="grid">.grid</div>
  <div class="grid">.grid</div>
</div>
<div class="box y-center">
  <div class="grid">.grid</div>
  <div class="grid">.grid</div>
</div>
<div class="box y-end">
  <div class="grid">.grid</div>
  <div class="grid">.grid</div>
</div>
```

反序列

为容器添加 `class="reverse"` 使块元素反序列



```
<div class="box reverse">
  <div class="grid">.grid 1</div>
  <div class="grid">.grid 2</div>
  <div class="grid">.grid 3</div>
</div>
```

块序列

添加 `class="order1~12"` 定义块元素的序列

.grid .order1	.grid .order2	.grid .order3
---------------	---------------	---------------

```
<div class="box">
  <div class="grid order3">.grid .order3</div>
  <div class="grid order1">.grid .order1</div>
  <div class="grid order2">.grid .order2</div>
</div>
```

.grid .order1	.grid	.grid .order12
---------------	-------	----------------

```
<div class="box">
  <div class="grid order12">.grid .order12</div>
  <div class="grid">.grid</div>
  <div class="grid order1">.grid .order1</div>
</div>
```

嵌套

可以不断向块元素内嵌套更深层容器

.grid35		.grid65	
.grid50	.grid50	.grid50	.grid50

```
<div class="box">
  <div class="grid35">
    <div class="box">
      <div class="grid50">.grid50</div>
      <div class="grid50">.grid50</div>
    </div>
  </div>
  <div class="grid65">
    <div class="box">
      <div class="grid50">.grid50</div>
      <div class="grid50">.grid50</div>
    </div>
  </div>
</div>
```

间距

添加 `class="plr"` 使块元素填充左右内间距 `padding-left:0.75rem;padding-right:0.75rem`

.grid33 .plr	.grid33 .plr	.grid33 .plr
.grid50 .plr		.grid50 .plr

```
<div class="box">
  <div class="grid33 plr">.grid33 .plr</div>
</div>
```