HTML && CSS

## IE8

margin、padding负值可能会赘余

使用ES3语法

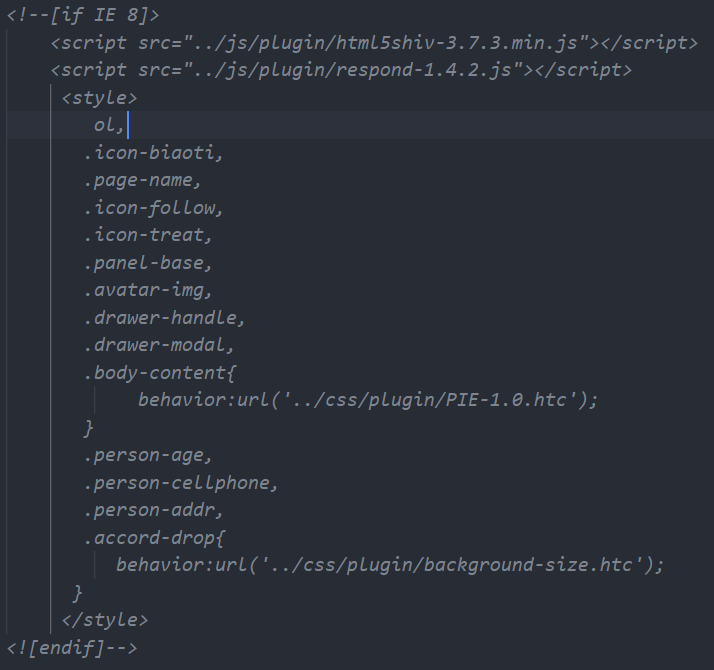
兼容border-radius – 使用PIE.htc文件

兼容栅格布局 – 服务器使用respond.js文件

box-shadow - ie滤镜:Shadow

background-size - 使用background-size.htc文件

检测浏览器选择样式：(注释括号中不能有空格）*<!--[if IE 8]> style <![endif]-->*



ie滤镜参考文件：<https://blog.csdn.net/c11073138/article/details/79646878>

## 动画(animation)

动画停留在最后一帧的样子

animation-fill-mode：forwards (默认none)

动画执行一次

animation-iteration-count:**1**;

图片旋转动画：

fatherDiv: position:relative; childDiv:position:absolute; // 定位，避免因旋转产生的宽高变化导致滚动

transform-origin: 50% 50%; // 设置变形中心  
 transform: perpective(300px) rotate(45deg) // perpective: 查看此图的距离，类似路灯效应 rotate: 旋转





## input

去除H5 input默认样式：

input[type=number] {

-moz-appearance: textfield;

}

input[type=number]::-webkit-inner-spin-button,

input[type=number]::-webkit-outer-spin-button {

-webkit-appearance: none;

margin: 0;

}

设置placeholder颜色：

input::-webkit-input-placeholder{

color: rgba(0,0,0,0.1)

}

## textarea

禁止拉伸：resize:none

## select

获取选择的option数值：查找此select元素获取value值

下拉被隐藏：【

1. 父元素overflow:hidden，因此导致超出部分被隐藏
2. z-index,层级不足，被高层级元素遮盖

】

多选：select中添加multiple,用数组接受select value值

设置下拉图标

select{

appearance:none;

-moz-appearance:none;

-webkit-appearance:none;

padding-right: 14px;

background: url("http://ourjs.github.io/static/2015/arrow.png") no-repeat scroll right center transparent;

}

//ie

select::-ms-expand { display: none; }

## float

作用：将元素设置为外部内联，内部块级的特性

外部内联：元素宽度为内部占据的最小空间宽度

内部块级：可设置宽高等

## Table

Tbody 超出大小滚动显示， 伪固定table头

thead{

display: block;

width: 100%;

background-color: orange;

}

tbody{

display: block;

height: 70px;

background-color: palegoldenrod;

overflow-y: auto;

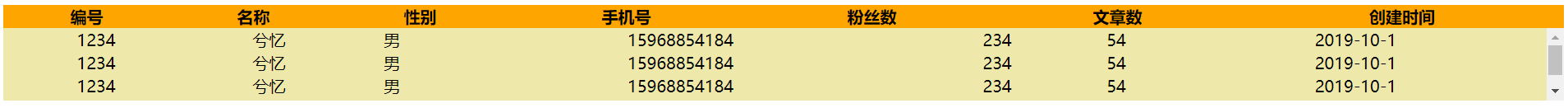
width: 100%;

}

此时table变形，失去display:table效果：

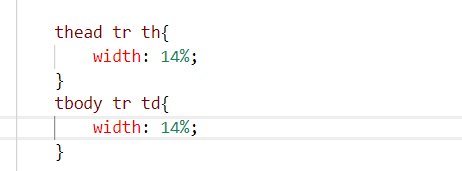


再将thead、tbody中的tr添加display:table可给其恢复各自的table特性：



但由于表头和body中使用了各自的table 自动分配空间，导致分布错位：解决方案：

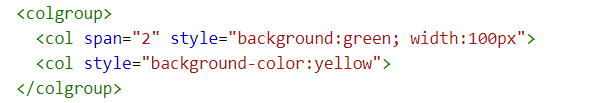
给thead与tbody中每个表格设置宽度，使其上下一致对齐

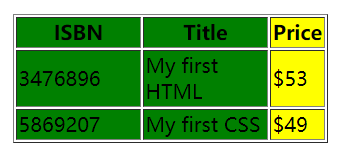




Colgroup：设置整个组（表）样式，目前仅支持背景颜色和宽度

Col：设置列样式 （背景色和宽度） 通常配合使用



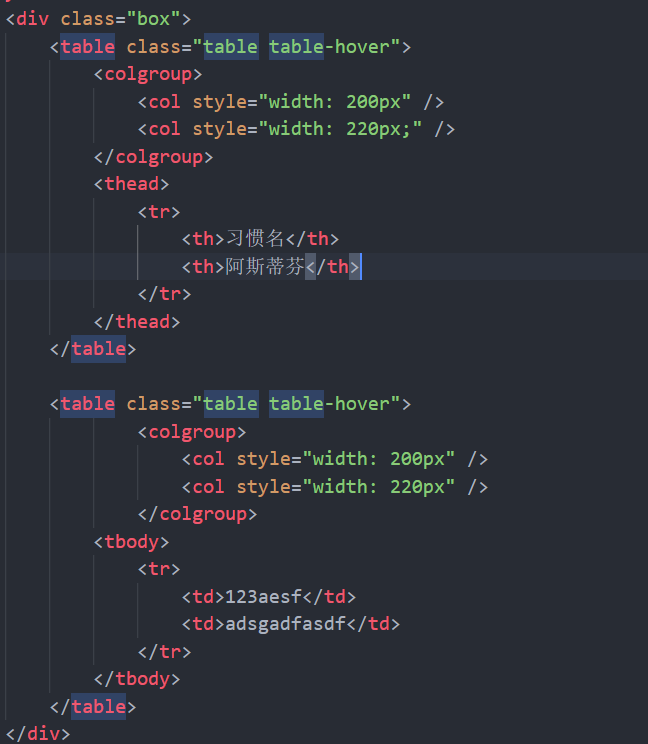


table固定表头：

1. 设置两个table，一个table中设置表头，另一个为列表
2. 设置table属性，table-layout:fixed;默认为table-layout:auto;需设置table宽度，否则会自动缩减
3. 使用colgroup与col配合使用，用来设置table中td宽度；
4. 设置盒子宽度超出滚动、table高度超出滚动
5. (当table中td总宽度小于table时，自动扩增内部td宽度)
6. 【td内容超出换行：word-wrap:break-word;(在长单词或url地址内部进行换行)

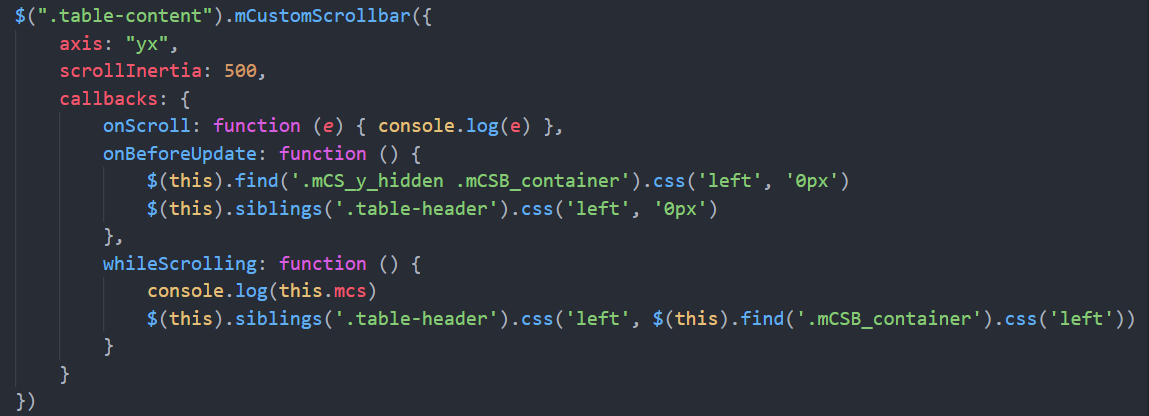
word-brak:break-all;(允许在单词内换行)

】



固定表头：

1. 盒子设置position:relative;table-header,table-body{position:absolute}
2. 设置table-body垂直水平滚动
3. 当table-body水平滚动时，同步设置table-header左边距



## div布局

核心思想：一个盒子难以达成效果，可以套用盒子组合来完成效果

如：单独一个盒子无法使用flex居中时，可以在此盒子外添加有大小的盒子，对外部盒子添加flex属性达到目标盒子居中的效果，**总想着节约标签往往适得其反，事倍功半**

**手机端页面：使用100%高度控制页面满屏显示**

## background

background-color：背景色

background-image：背景图片

background-repeat：背景图是否填充（no-repeat：不重复 repeat-y：垂直方向重复显示 repeat-x：水平方向重复显示 repeat：水平垂直方向均重复显示(默认)）

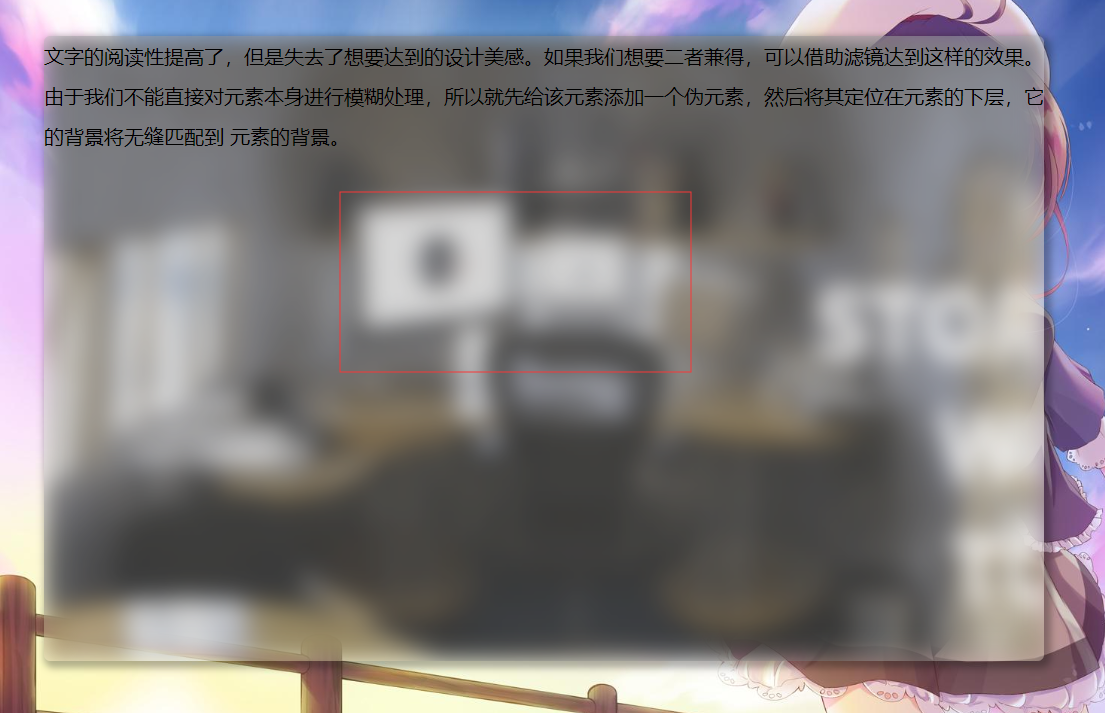
background-attachment：背景图滚动情况（fixed：不随页面滚动而滚动 local：随元素内容滚动而滚动 scroll：随页面滚动而滚动(滚动)）

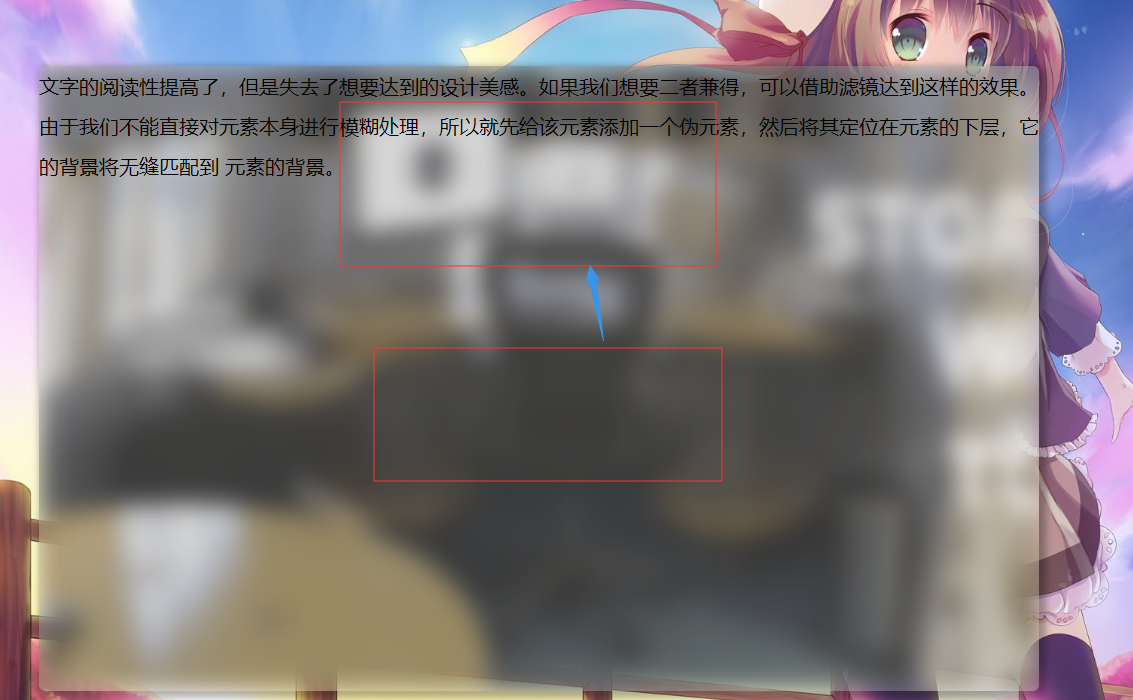
background-position / background-size：注意：背景图定位与大小通用时需要用”/”分隔

background-position：背景图片所在的位置（x, y：水平方向|垂直方向，如：xx-position: 200px 100px）;

background-size：背景图片高度和宽度（length：x y：宽度|高度 cover：保持图像纵横比并将图像缩放成将完全覆盖背景定位区域的最小大小

contain：保持图像纵横比并将图像缩放成将适合背景定位区域的最大大小）





## BFC

### 特性

1. 内部的Box会在垂直方向，从顶部开始一个接一个地放置
2. Box垂直方向的距离由margin决定。属于同一个BFC的两个相邻Box的margin会发生叠加
3. 每个元素的margin box的左边，与包含块border box的左边相接触（对于从左往右的格式化，否则相反）。即使存在浮动也是如此
4. BFC的区域不会与float box叠加
5. BFC就是页面上的一个隔离的独立容器，容器里面的子元素不会影响到外面的元素，反之亦然。
6. 计算BFC的高度时，浮动元素也参与计算

### 触发BFC（判断BFC）

1. float除了none以外的值
2. overflow除了visible以外的值（hidden、auto、scroll）
3. display（table-cell、table-caption、inline-block、flex、inline-flex）
4. position值为（absolute、fixed）
5. fieldset元素（body根元素）

### BFC应用

解决margin重叠：给其中一个元素设置BFC（触发BFC的方式，如overflow:hidden）{ 利用BFC特性：2}

高度塌陷（子元素float）：为其设置BFC（如：overflow:hidden、display:flex等）,{ 利用BFC特性：6}

float元素重叠：为另外的这元素设置BFC（如：overflow:hidden、display:inner-block等）， { 利用BFC特性：4}

…

## 弹性布局

### Flex

Flex : flex-grow | flex-shrink | flex-basis （父元素设置display:flex，并且只对直接子元素起伸缩效果，其中若设置margin、padding，此padding间隔距离无法被压缩）

{放大比例 默认0 | 缩小比例 默认1 | 计算主轴是否有多余空间 默认auto,原本大小}

【flex-grow: 当父元素宽度大于子元素宽度之和，其可让子元素分享父元素多余空间】

【flex-shrink: 当父元素宽度小于子元素宽度之和，其让子元素按比例收缩】

【flex-basis: 设置元素宽度，当元素同时设置width和flex-basis，width值被其覆盖】

【设置align-self可让此元素宽度为原本大小，取消auto中继承stretch 100%】

当无flex-direction: column; 时，此flex无效，设置align-self后同过min-width设置width

例：flex: 0 0 50%;

**注意点：flex-wrap: wrap ：空间不足时折行**

**坑点：折行后align-content默认值：stretch（各行伸屏以占用剩余的空间，若剩**

**余空间不足，等效于flex-start，其他情况下，剩余空间被所有行平分）**

## 盒模型

box-size: border-box; //以自身空间自动计算实际大小

如：father{width:200px} son{width:200px,padding:20px}

此时子元素实际大小为width:240px height:240px;

设置box-size:border-box后, 子元素实际大小：width:200px height:200px;

内部空间：160px,即从自身去除了padding值

总结：未设置时：以自身向外添加 | 减少大小

设置后： 以自身向内添加 | 减少大小

## 栅格布局-bootstrp

使用：类名添加栅格标签： class=”col-md-4”

class：row --让此标签占据整行

class：form-inline --表单水平排列

class：form-group --设置元素为inline-block

class：form-control --设置元素width:100%，阴影效果，颜色:近灰色,border,padding

class：btn btn-default --类似control，添加悬浮变色

## 选择器

#### nth-child与nth-of-type

nth-child: 从父元素内的全部子元素选择子元素，且选择对象要存在

例：p:nth-child(5) ,选择父元素中的第五个元素，若第五个元素不是p，则不起作用

nth-of-type: 从父元素内的选择元素中排列顺序进行选择，且选择对象要存在

例：a:nth-of-type(2): 选择第二个a元素

## 注意项

1. 行内元素无法包含块级元素，块级元素可包含行内元素，同级元素可互相包含：

高等级包含低等级，不可逆

* 1. Inline:1
  2. Inline-block:2
  3. block:3

### 居中

1. flex居中【

display:flex; //设置为flex（

justify-content:center; //水平方向居中

align-items:center //垂直方向居中

】

2、文字居中【

Text-align: center; //继承 {父元素设置，子元素中的行内元素在

其之内水平居中 ，，只对行内元素有效}

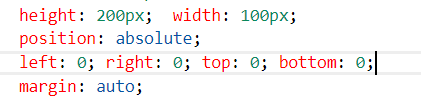
】

1. margin居中【

目标居中元素设置:margin: 0 auto; //水平居中

上下居中使用margin: auto;

//只能绝对定位后left,right,top,bottom=0(相对0位置并且本身与父容器有宽高的情况下再margin:auto时才有效)



】

1. transform:translate居中【

position: absolute;

top:50%;

left:50%;

transform:translate(-50%,-50%)

//优势： 不需知晓本身宽高即可自动居中

//缺点： 兼容性较差，ie10+才支持使用

】

1. table布局居中【

father: { width: 200px; height: 200px; display: table; text-aligin:center;

vertical-align:middel}

son: {width:10px; height:10px; display:inline-block;}

**display:table-cell（需在其父元素设置display:table）：使子元素适应vertical-aligin:middle**

**子元素：display:inline-block; 让此元素适应text-aligin:center**

**以此达到：上下居中，左右居中**

】

1. vertical-align:middle居中【

father{ width: 200px; height: 200px; display:inline-block;}

father:after{ display:inline-block; content: ‘’; height:100%; vertical-align:middle;}

son{ display:inline-block; vertical-align:middle;}

原理：变化元素基准线

作用方式：让父元素的伪元素与其高度相等，并将基准线变为居中，子元素再设置vertical-align:middle达到让子元素基准对其父元素基准线，使上下居中

参考：<https://segmentfault.com/q/1010000012107350>

<https://developer.mozilla.org/zh-CN/docs/Web/CSS/vertical-align>

】

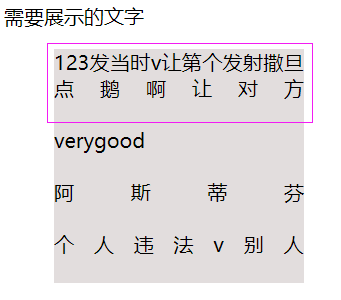
### 文字两端对齐

父容器设置text-align: justify; --两端对齐

兼容ie8：text-align-last:justify; 并且文字都有间隔

参考：<https://blog.csdn.net/qq_29608143/article/details/83625318>

单独设置时只对多行文字的情况生效



单行文字需添加占位标签，目的：添加索引去设置文字宽度100%

【对数字字母无效，空格分开的数字字母按文字效果对齐】

使用标签占位后会导致多出垂直方向的间隔，取消间隔可以设置容器高度方式，如

li{height:30px; line-height:30px}





## 未解疑惑

### 1、flex

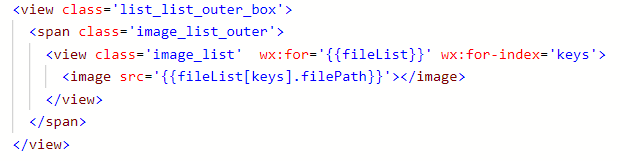
父元素设置display:flex 后，子元素中的元素都变为块级元素，重新设置display:inline无效。

## 已解疑惑

1. flex最后一行依照先前左对齐排列：

给整体元素添加一个盒子，对此盒子再添加对齐方式，盒子中的内容根据想要的方式排序（不成熟方案）





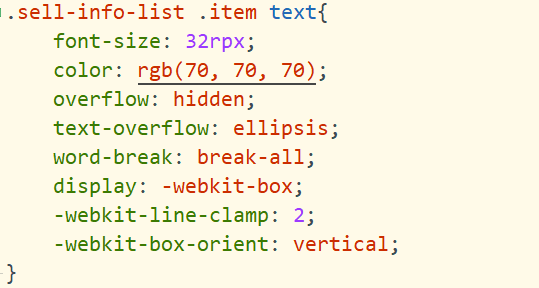


页面加载局部图片图标：

单行、多行文本溢出省略号显示：

overflow:hidden; text-overflow:ellipsis; white-space:nowrap;

display:-webkit-box; -webkit-box-orient:vertical; -webkit-line-clamp:3; overflow:hidden;



## 结论

1. 避免因padding、margin导致内容超出而出现滚动
   1. 设置父元素overflow:hidden (超出隐藏)
   2. 设置父元素flex，自动分配内容（但是margin与padding依旧占有空间，可能无法均分）
   3. 设置父元素box-sizing:border-box | content-box (默认) | inherit

border-box：怪异模式，border、padding均包含在盒子内

content-box：内容包含在盒子内 （默认值, border、margin、padding均向外撑）

inherit：继承父元素的box-sizing值

兼容：box-sizing:border-box;

-moz-box-sizing:border-box; /\* Firefox \*/

-webkit-box-sizing:border-box; /\* Safari \*/