# CSS

## CSS关系选择符

### 1.包含选择符(E F):

**div p{rules}**

选择所有被div元素包含的p元素(包含了儿子,孙子,孙子的孙子...);

### 2.子选择符(E>F)

**div>p{rules}**

选择所有作被div元素包含的的子元素p(不含孙辈)。

### 3.相邻选择符(E+F)

**div+p{rules}**

选择紧贴在div元素之后第一个p元素，元素div与p必须同属一个父级;

### 4.兄弟选择符(E~F)

**div~p{rules}**

选择div元素后面的所有兄弟元素p，元素didv与p必须同属一个父级;

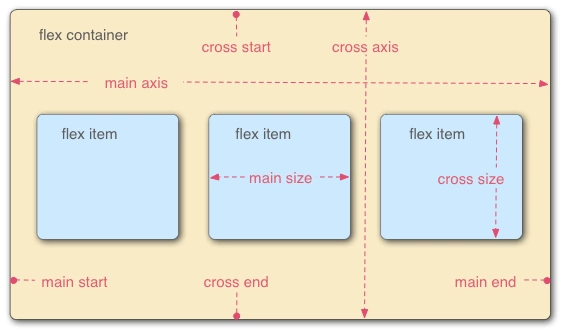
## 弹性布局(display:flex)

### 1.什么是Flexbox

Flexbox是 CSS3 引入的新的布局模式, 它能够扩展和收缩 flex 容器内的元素，以最大限度地填充可用空间。

### 2.基本概念：

Flexbox有三个核心概念：

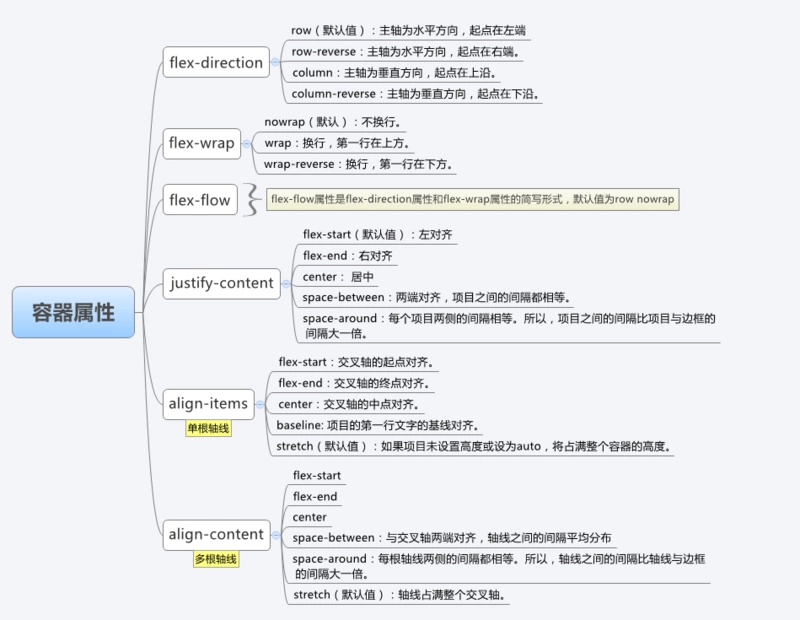


– flex 项（注：也称 flex 子元素），需要布局的元素

– flex 容器，其包含 flex 项

– 排列方向（direction），这决定了 flex 项的布局方向

### 3. 容器属性：



### 4.项目属性:



### 5.项目属性中的flex属性

flex: flex-grow flex-shrink flex-basis;

flex-grow：定义项目的放大比例，默认为0，即如果存在剩余空间，也不放大；

flex-shrink：定义了项目的缩小比例，默认为1，即如果空间不足，该项目将缩小；

flex-basis:

默认值:flex: 0 1 auto;

Flex:1;===》 flex:1 1 0%；

Felx:1 1===> flex:1 1 0%

相关文档：

Felx布局相关文档：<https://www.cnblogs.com/hellocd/p/10443237.html>

flex属性相关文档：<https://blog.csdn.net/qq_31324879/article/details/102528797>

## 3.[函数类型取值 Functional Notations Type](https://www.xyhtml5.com/manual-css3/values/functional/index.htm)

### 1.calc()用于动态计算长度值

**用于动态计算长度值。**

* 任何长度值都可以使用calc()函数进行计算。
* calc()函数支持+, -, \*, /, mod运算

相关资料：<https://www.xyhtml5.com/manual-css3/>

## 4.font-size:0

代码自动格式化的时候，往往会设置一些适当的缩进、换行，但当元素的display为inline或者inline-block的时候，这些缩进、换行就会产生空白，导致前端页面展示变形。

为父元素设置font-size：0；然后单独设置子元素的字体大（font-size）小;

相关文档：<https://blog.csdn.net/weixin_43743148/article/details/107838014>

## 5.font-size:100%

设置字体属性为默认大小，是**相对于浏览器默认字体大小**或继承body设定的字体大小来说的

相关文档：https://www.cnblogs.com/myprogramer/p/10265582.html

## 6.图像类型取值 Image Type

### 1.linear-gradient()实现渐变的效果

**可以****实现渐变的效果;**

a{

display: block;

min-width: 96px;

height: 32px;

line-height: 32px;

text-align: center;

color: #fff;

background: linear-gradient(92deg,#ffba40 0,#ff503e 37%,#ff2f50 81%,#ff1b40 100%);

border-radius: 16px;

margin-top: calc((44px - 32px)/ 2);

}

## 7.appearance属性

使 div 元素看上去像一个按钮：

div

{

appearance:button;

-moz-appearance:button; /\* Firefox \*/

-webkit-appearance:button; /\* Safari 和 Chrome \*/

}

相关文档：<https://www.w3school.com.cn/cssref/pr_appearance.asp>

## 8.Flex兼容问题

.container{

display: -webkit-box; /\* Chrome 4+, Safari 3.1, iOS Safari 3.2+ \*/

display: -moz-box; /\* Firefox 17- \*/

display: -webkit-flex; /\* Chrome 21+, Safari 6.1+, iOS Safari 7+, Opera 15/16 \*/

display: -moz-flex; /\* Firefox 18+ \*/

display: -ms-flexbox; /\* IE 10 \*/

display: flex; /\* Chrome 29+, Firefox 22+, IE 11+, Opera 12.1/17/18, Android 4.4+ \*/

}

相关文档：<https://blog.csdn.net/qq_37029814/article/details/89521240>

# 页面设计

## 浏览器的内核

浏览器的核心部分是渲染引擎,一般我们称为浏览器内核。

内核的分类：

**1.Trident(IE内核)**

**2.Webkit（**Safari内核，Chrome内核原型**）**

**3.Blink**

**4.Gecko（**Firefox内核**）**

**IE浏览器：**Trident内核

**Chrom浏览器**：以前是webkit内核,现在是blink内核

FireFox浏览器：**Gecko内核.**

**Opera浏览器：**最初是presto内核,后来是webkit,现在采用blink内核

Safar浏览器：采用webkit内核.

**360浏览器、QQ浏览器:** IE（兼容模式）+chrome（极速模式）

相关文档：

<https://blog.csdn.net/qq_31214097/article/details/84708917>

<https://blog.csdn.net/qq_36135335/article/details/95166490>

## 布局

### 一列布局

特点:

1. 页面内容区域宽度固定;
2. 页面内容区域在浏览器自适应居中显示；

实现方式：

固定宽度和margin的方式实现;

<div class="w">

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

</div>

.w{

width: 固定宽度;

margin: 0 auto;

}

固定宽度设置:

以主流的分辨率来判断；

公众号677px;

应用：

公众号

### 两列布局

特点：一列固定宽度,一列自动适应宽;

1.固定宽度的列：通常称为边栏 主要放置一些固定的内容如导航、菜单之类;

2.自适应宽的列：根据浏览器窗口的大小自动判断宽度，主要放置主体内容

实现方式:

1. 使用flaot方式实现

<div class="w">

<div class="main-column"></div>

<div class="right-sideBar"></div>

</div>

.固定容器{

width: 固定宽度;

float: left|right;

}

.自适应宽的容器{//不能设置宽度

margin-方向: 固定宽度;

}

.main-column{

width: 300px;

float: left;

}

.right-sideBar{

margin-left: 300px;

}

.w{

width: 1000px;

margin: 0 auto;

}

### 三列布局

特点：

1.中间自适应宽度,两边定宽;

2. 中间列放置主体内容，在浏览器中优先展示渲染；

原理：

当子元素处于浮动状态时，设置负margin>=子元素宽度时，子元素会叠盖到兄弟元素之上；

<div class="w">

<div class="mod-container">

<div class="main">main</div>

<div class="nav">nav</div>

<div class="slider">slider</div>

</div>

</div>

.w{/\*\*设置盒子的整体宽度使其居中显示\*\*/

width: 1200px;

margin: 0 auto;

}

.main{/\*\*中间自适应的盒子\*\*/

float: left;

width: 100%;

padding-left: 200px;

box-sizing: border-box;

padding-right: 300px;

/\*为了能看到效果添加了border和height\*/

border: 1px solid #ebebeb;

height: 500px;

}

.nav{/\*\*左边的盒子\*\*/

width: 200px; /\*固定宽度\*/

float: left;

margin-left: -100%;

/\*为了能看到效果添加了border和height\*/

border: 1px solid #ebebeb;

height: 500px;

}

.slider{/\*\*左边的盒子\*\*/

width: 300px; /\*固定宽度\*/

float: left;

margin-left: -302px;

/\*为了能看到效果添加了border和height\*/

border: 1px solid #ebebeb;

height: 500px;

}

### 弹性布局

传统的布局基于盒子模型，依赖于display属性+position属性+flaot属性,垂直居中很难实现;

设为Flex布局以后，子元素的float、clear和vertical-align属性将失效；

相关文档：

<https://www.runoob.com/w3cnote/flex-grammar.html>

<https://www.cnblogs.com/hellocd/p/10443237.html>

<http://www.ruanyifeng.com/blog/2015/07/flex-grammar.html?utm_source=tuicool>

<https://blog.csdn.net/lwr3379966/article/details/105003865/>

## 居中

1.元素水平居中

.container{

width: 固定宽度;

margin: 0 auto;

}

限制:

必须设置元素为固定宽度;

文字垂直居中

设置容器的height和line-height一致;

.container{

height: 32px;

line-height: 32px;

}

## 字体

1.正文字体大小:

一般设置为14px、15px、16px;

2.字体颜色一般设置为黑色:

#20232c #121212

3.border的颜色

#f6f6f6

# 遇到的问题及解决方案

## 1.height=100%无效；

%是一个相对父元素计算得来的高度，要想使他有效，我们需要设置父元素的height;

2.导航

https://www.zhihu.com/



https://www.csdn.net/



## 2.HTML元素塌陷问题及解决方案

如果父元素只包含了浮动元素 ，那么父元素在未设置高度的同时，则父元素高度塌缩为零;

解决方案：

1.创建一个用来清除浮动的css样式类（.clearfix）  
  2.针对包裹的全是浮动元素的父级容器使用

.clearfix{zoom:1}/\*IE浏览器的专用属性（针对IE\*/）

.clearfix:after{/\*伪对象选择符 （在这个对象被浏览器渲染后添加一定的内容）\*/

content:".";/\*添加内容写在这里的属性值 这个属性是专门配合伪类对象,必须写;\*/

display:block;/\*将添加的内容转换为块级元素\*/

visibility:hidden;/\*可视化属性；控制元素是否可见;无论是否可见，保留物理空间；和display属性不同;\*/

height:0;/\*将添加进去的内容的高度设置为0；消除其占位；\*/

clear:both;/\*将添加去的内容作为清除浮动的对象，实现外围对象中有内容存在，因此可以自动判断高度;（告诉浮动元素说不要出现在我的两边）\*/

}