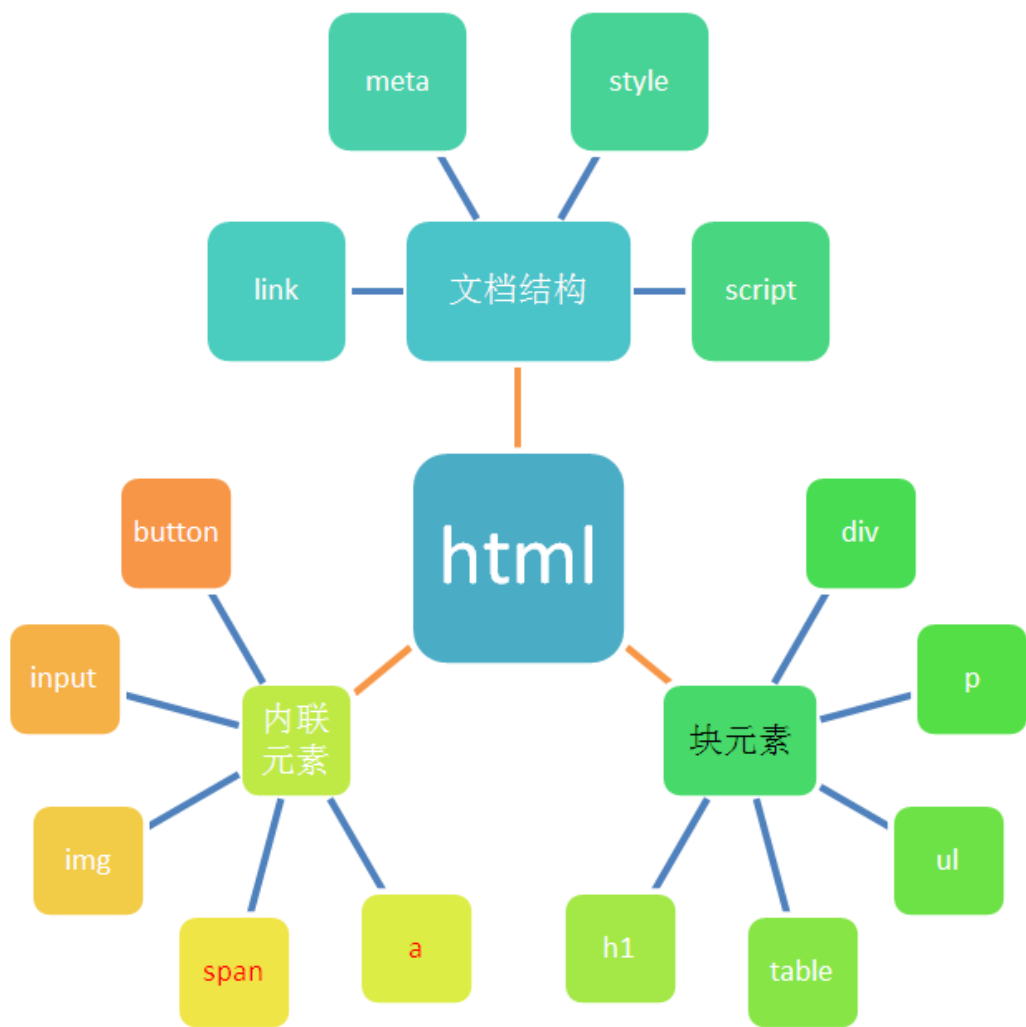
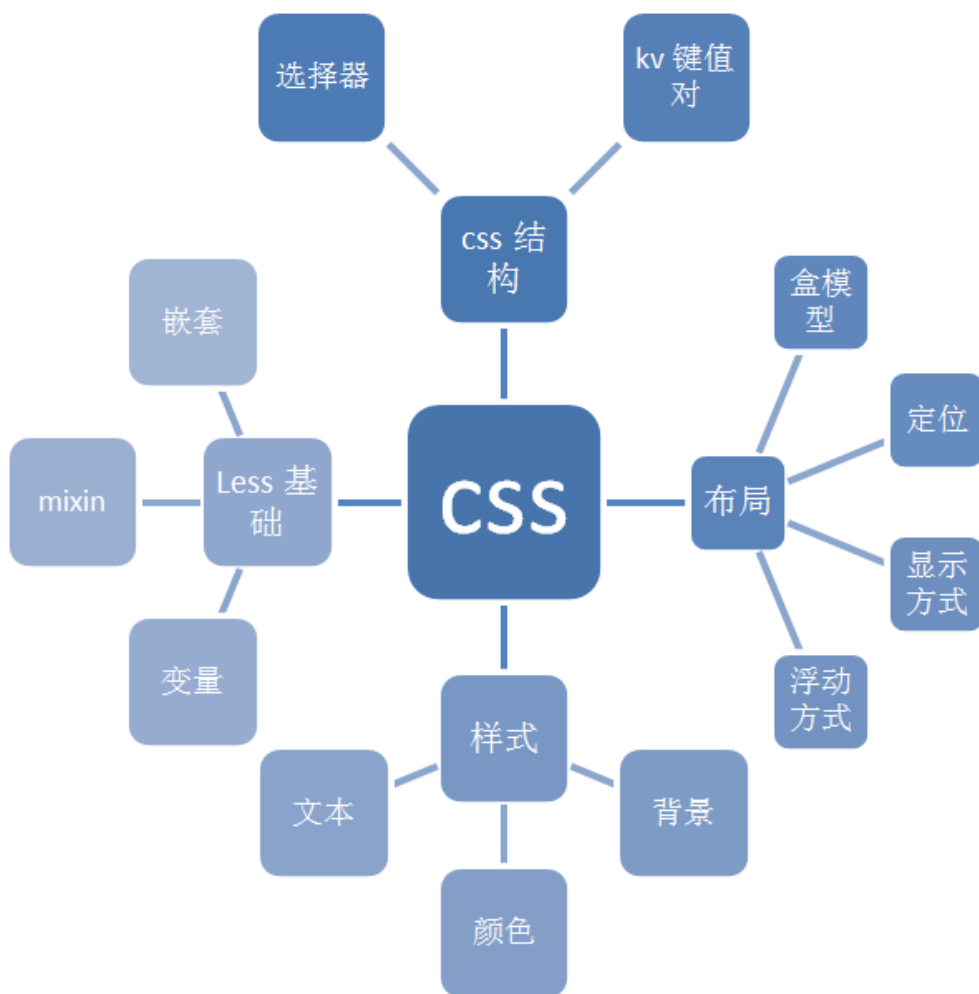


前端基础知识分享

2014 2月 19, 星期三, 12:26 下午

分享内容





html基础

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <link rel="stylesheet" type="text/css" href="src/css/main.css" />
  </head>
  <body>
    <div class="wrapper">
      <a href="http://www.baidu.com" title="百度搜索">百度</a>
    </div>
  </body></html>
```

HTML 块元素

大多数 **HTML** 元素被定义为块级元素或内联元素。

编者注：“块级元素”译为 **block level element**，“内联元素”译为 **inline element**。

块级元素在浏览器显示时，通常会以新行来开始（和结束）。

例子：

```
<h1>, <p>, <ul>, <table>
```

HTML 内联元素

内联元素在显示时通常不会以新行开始。

例子：

```
<b>, <td>, <a>, <img>
```

大部分html标签都有一定语义；但是部分元素只是用来包装元素和方便样式定义

HTML 分组标签

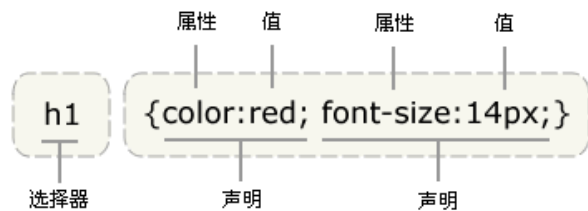
标签	描述
div	定义文档中的分区或节（ division/section ）。
span	定义 span ，用来组合文档中的行内元素。

html 标签元素类型

- 标题
- 列表
- 表格
- 图像
- 链接
- 输入元素

CSS 基础

CSS 语法结构



选择器

- 标签选择器

```
div {line-height: 30px}
```

- ID 选择器

```
#main-wrapper {width: 500px; margin: 10px auto;}
```

- class 选取器

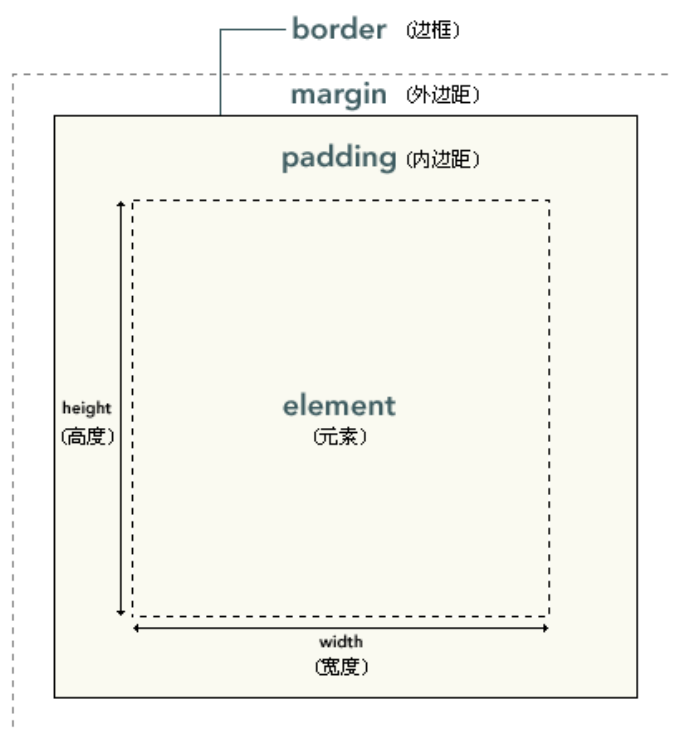
```
.clear {clear: both;}
```

- 组合选择器

```
#wrapper li {list-style: none; width: 150px;}
```

更多子类选择器；相邻选择器；属性选择器；伪类选择器参考w3c文档

CSS 盒模型



大小: width, height

边框: border

内边距: padding

外边距: margin

- 边框和内边距及外边距都会占用元素布局空间; 因此计算时要考虑这些因素
- 外边距可以使用负值; 但是其它属性不可以;
- 各属性都可以单独定义四边大小;
- 单行定义 ((上下-左右; 顺时针方向 分别为top – right-bottom-left))
- 只有块元素设置 `width` 、 `height` 属性有效

上下外边距自动合并, 参考w3c文档

定位: position

通过使用 `position` 属性, 我们可以选择 4 种不同类型的定位, 这会影响元素框生成的方式。

- static

元素框正常生成。块级元素生成一个矩形框, 作为文档流的一部分, 行内元素则会创建一个或多个行框, 置于其父元素中。

- relative

元素框偏移某个距离。元素仍保持其未定位前的形状, 它原本所占的空间仍保留。

- absolute

元素框从文档流完全删除, 并相对于其包含块定位。包含块可能是文档中的另一个元素或者是初始包含块。元素原先在正常文档流中所占的空间会关闭, 就好像元素原来不存在一样。元素定位后生成一个块级框, 而不论原来它在正常流中生成何种类型的框。

- fixed

元素框的表现类似于将 `position` 设置为 `absolute`, 不过其包含块是视窗本身。

`position` 属性说明定位方式; 位置确定可以通过属性 `top` `right` `bottom` `left` 实现

```
{
  position: absolute;
  left: 100px;
  top: 150px
}
```

显示方式 display

隐藏元素: `display: none`

- 区别 `visibility: hidden`

显示为块元素: `display: block`

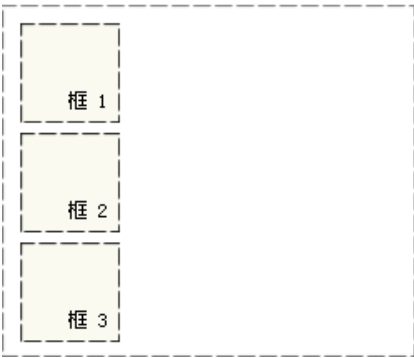
显示为内联元素: `display: inline`

值	描述
none	此元素不会被显示。
block	此元素将显示为块级元素，此元素前后会带有换行符。
inline	默认。此元素会被显示为内联元素，元素前后没有换行符。
inline-block	行内块元素。（CSS2.1 新增的值）
list-item	此元素会作为列表显示。
run-in	此元素会根据上下文作为块级元素或内联元素显示。
table	此元素会作为块级表格来显示（类似 <code><table></code> ），表格前后带有换行符。
inline-table	此元素会作为内联表格来显示（类似 <code><table></code> ），表格前后没有换行符。
inherit	规定应该从父元素继承 <code>display</code> 属性的值。

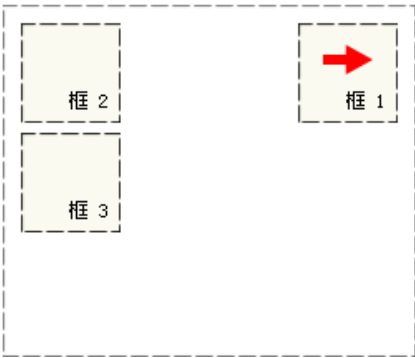
浮动: float

浮动的框可以向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止。

不浮动的框



框 1 向右浮动



浮动元素脱离原默认的布局空间；如果要占用一定空间，不遮住其它元素，需要用到 `clear` 属性

样式

- 背景

-- 背景颜色 `background-color`

-- 背景图片 `background-image`

-- 背景重复 `background-repeat`

-- 背景位置 `background-position`

```
p {background-color: gray;}
body {
  background-image: url("/i/eg_bg_03.gif");
  background-repeat: no-repeat;
  background-position: -60px -30px;
}
```

- 文本

属性	描述
color	设置文本颜色
line-height	设置行高。
letter-spacing	设置字符间距。
text-align	对齐元素中的文本。
text-decoration	向文本添加修饰。
text-indent	缩进元素中文本的首行。
text-shadow	设置文本阴影。CSS2 包含该属性，但是 CSS2.1 没有保留该属性。
word-spacing	设置字间距。

- 链接

```
a:link {text-decoration: none;}
a:visited {text-decoration: none;}
a:hover {text-decoration: underline;}
a:active {text-decoration: underline;}
```

- 列表

属性	描述
list-style	简写属性。用于把所有用于列表的属性设置于一个声明中。
list-style-image	将图象设置为列表项标志。
list-style-position	设置列表中列表项标志的位置。
list-style-type	设置列表项标志的类型。

嵌入样式方法

外部样式表

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css" />
</head>
```

内部样式表

```
<head>
<style type="text/css">
  hr {color: sienna;}
```

```
p {margin-left: 20px;}
body {background-image: url("images/back40.gif");}
</style>
</head>
```

内联样式

```
<p style="color: sienna; margin-left: 20px"></p>
```

less基础

变量定义

```
@nice-blue: #5B83AD;
@light-blue: @nice-blue + #111;
#header { color: @light-blue; }
```

输出结果为

```
#header { color: #6c94be; }
```

mixin

```
.bordered {
  border-top: dotted 1px black;
  border-bottom: solid 2px black;
}

#menu a {
  color: #111;
  .bordered;
}
```

输出结果为

```
#menu a {
  color: #111;
  border-top: dotted 1px black;
  border-bottom: solid 2px black;
}
```

带参混合

```
.border-radius (@radius) {
  border-radius: @radius;
  -moz-border-radius: @radius;
  -webkit-border-radius: @radius;
}

#header {
  .border-radius(4px);
}
```



```
}
```

嵌套规则

less 可以基于dom元素的层次结构实现css的嵌套定义

```
#header {  
  color: black;  
  
  .navigation {  
    font-size: 12px;  
  }  
  
  .logo {  
    width: 300px;  
    &:hover { text-decoration: none }  
  }  
}
```

输出结果为

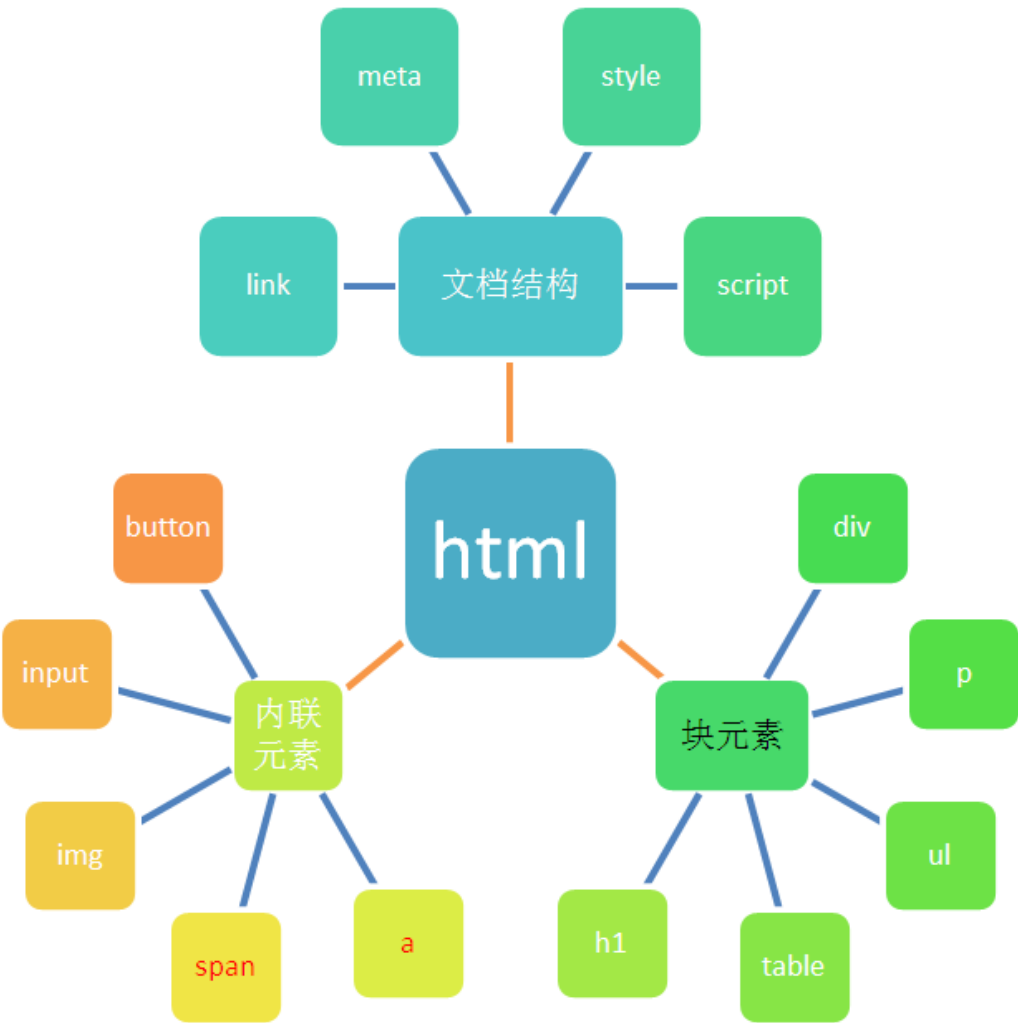
```
#header {  
  color: black;  
}  
#header .navigation {  
  font-size: 12px;  
}  
#header .logo {  
  width: 300px;  
}  
#header .logo:hover {  
  text-decoration: none;  
}
```

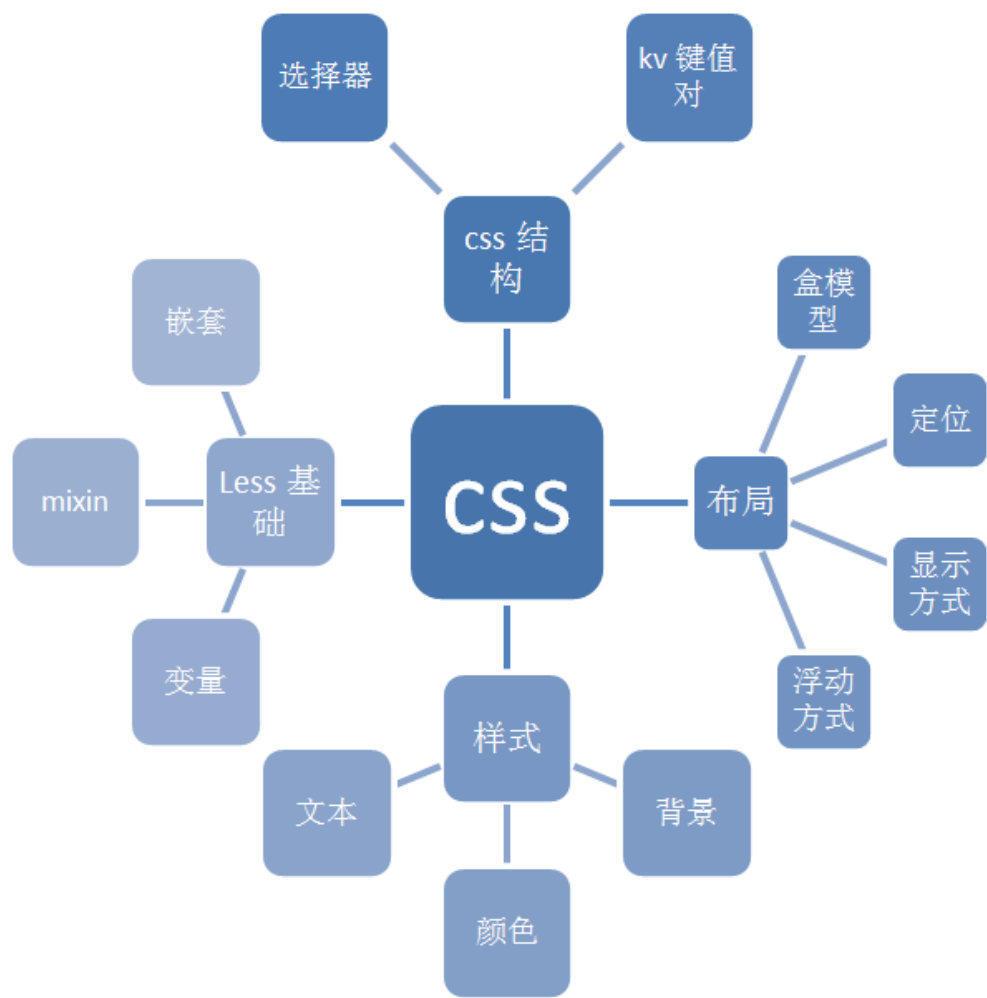
以下内容自行参考文档学习

- 条件混合
- 变量作用域
- 变量操作符
- 内嵌函数
- 忽略编译
- js语句执行

掌握less的核心语法；目标是为了写出可读性更高，执行效率更高，管理更方便的前端样式代码；

总结





less 在线练习

练习模拟