# HTML语法结构及规范

## 什么是HTML？

Html是一种超文本标记语言;是描述网页内容和结构的基本标记语言。

## HTML中如何注释？

<!--注释内容-->

注释的作用：代码说明;

## HTML语法结构？

1.符号

使用尖括号<>,所有的HTML语言都必须括在其中;

2.标签格式

双标签：成对出现<标签名></标签名> <titile>HTML语言</title>

单标签：<br/>只存在开始标签，不需要结束;

总结：双标签通常用于包含内容的元素,单标签表示无内容的元素;

3. 标签结构

标签由3个组成部分：标签 属性 值。

语法结构：<标签名 属性="值" 属性="值">...</标签名 >

a.标签可以用于属性，属性和值都写在开始标签的尖括号中；

b.标签和属性之间用空格隔开，多个属性之间也用空格隔开；

c.如果属性有对应的值，则属性和值之间事等式关系，且值需要在英文引号内;

## htmL使用规范

1.标签名和属性名都使用小写;

2.属性的值用双引号括起来;

3.标签使用符号嵌套规则;

## 编写网页的流程

1.找参考

2.准备素材/建站

3.添加内容元素

4.标签结构化

## 基本结构

<html>

<head></head>

<body></body>

<html>

<!DOCTYPE html><!--html文档的文档类型声明-->

<html><!--html文档的开始-->

<head><!--主要用来说明html文档的相关属性的或设值-->

<meta charset="UTF-8"><!--指定网页的编码-->

<title></title><!--设定网页的标题-->

</head>

<body><!--html文档的主体区域，对应浏览器的呈现内容-->

</body>

</html><!--html文档的结束-->

# CSS语法结构节规范

## 什么是CSS?

CSS（Cascading Style Sheets，层叠样式表），是一种用来表现HTML等文件样式的计算机语言。CSS不仅可以静态地修饰网页，还可以配合各种脚本语言动态地对网页各元素进行格式化。

## CSS中如何注释？

/\*注释内容\*/

## CSS基础语法结构？

**css基础语法**

组成部分：**对象{样式声明}**

基本格式：**选择符{属性:值;}**

**说明：1.选择符即是css样式要作用的html对象；**有多种类型:标签本身（标签名称）、标签的特定属性（id或class）、标签的某中状态（伪类）；

**2. 样式声明必须被括在花括号{}当中**

**3.样式的声明由样式的属性和对应的值两部分;**根据属性不同的视觉

表现方式进行分类，可分为：文本的设定、尺寸的设定、

色彩的设定；对应的值根据样式声明属性的不同类型，有对应

的值及单位，如色彩的值、尺寸的单位、方向值;

**4.样式的声明的属性和值之间用冒号：分隔;**

**5.样式声明用分号(;)结束；**

实例：

#first-p{color:red;}

.nav{font-size:18px;}

p{background:gray;}

## 常用选择符(对象)

**选择符(对象)**：以html标签作为css的对象（标签本身,标签的特定属性id或class,标签的某种状态（伪类））;

### 1.id选择符(#id名)

以html标签的唯一标识符id属性的值作为对象;

在html语言中：id属性是html中的通用属性，但是id的值的命名是唯一的,不允许重复；

css语言中：#代表后面的名称代表是id的值,id选择符主要用于针对具有唯一性的html对象;

### 2.class选择符(.class名)

将一组css样式归纳为一个类,做为css应用的对象；

html语言中：class属性是html标签的通用属性，可以定义多个,类名之间以空格分隔；

css语言中：class选择符主要用于需要调用相同的css样式的多个html对象;

### 选择符优先级

主要对网页文档中的基础标签进行统一的标准化设定；

优先级：

id选择符>class选择符>标签选择符

### id和class命名规则

1.id属性和class属性的值命名都为自定义；

2.不允许用数字开头，不允许使用特殊符号;

3.命名最好具有语义化，可使用-或\_来连接词组;

### 4.组合选择符

#### 1.通配选择符（\*）

适配文档中的所有的html对象；

用于定义html文档中所以对象的基础样式（消除浏览器默认设置的基础样式）

根据就近原则，要放在css代码的首行；（从上至下读，先加载默认的在做其他特殊的操作）

最常用的用法： \*{margin:0;padding: 0;} 消除所有html标签默认存在的内外边距；

#### 2.包含选择符（父级对象 子级对象）

根据html对象的嵌套关系，指定某个元素的子级元素的样式；

选择符对象可以是多重包含，即一层层的包含下去，样式仅作用在最后的内部对象；

元素对象之间用空格分割；

#### 3.群组选择符（对象1,对象2....）

将相同的样式用于多个选择符对象，选择符对象之间用逗号隔开;

h1,h2,h3,h4{color:blue;}

#### 4.指定选择符(标签名.class类名)

将class类选择符和标签选择符结合在一起使用；

主要适用于在使用了某个class类的所有html对象中特别指定某一标签;

p.h1{color:blue}

## css样式的分类

1.行内样式

<div id="box" style="font-size: 15px;"></div>

通过标签style属性来设定css样式，css代码位于HTML标签的标记中;

仅对当前使用的标签有效;

2.内嵌样式

<style>\*{margin: 0;padding: 0;}<style>

将css放在页面的head区域中，使用 <style></style>标签进行

调用; 仅对当前页面有效,不能跨页面使用;

3.外部样式

<link rel="stylesheet" href="css/yuedu.css" />

单独的css样式文件 （.css）,在网页的head区域通过 <link>标签关联css文件; <link>标签的rel属性指定为stylesheet,href属性指定css文件的路径;实现了html结构和css样式完全分离,多个网页可以调用1个样式文件;

4.就近原则

a.相同权值得情况下，离被设置元素越近优先级越高;

行内样式>内嵌样式>外部样式

## css属性和值

1.css参考手册;

http://t.mb5u.com/css3/

2.常见的取值：

数值单位 ：px(像素)/ew(倍数（相对父元素）)/%（百分比）

方向：top/right/left/bottom/center

色彩：16进制代码（#RRGGBB或#RGB）0-9/a-z/基本色彩单词；

引用路径：url(文件路径)

3.值的一些写法规则

1.表示长宽高的数值必须带上单位；

2.表示色彩的十六进制代码前面必须用#号；

3.一个属性如果有多个值，值与值之间用空格分隔；

4.如果指定多个方向按上右下左的顺序排列；

# 1.HTML元素塌陷问题解决方案

### 问题描述

当父元素只包含了浮动元素且父元素未设置高度的时候，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

3代码实现

a.HTML代码

<div id="mod\_container" class="clearfix">

      <div id="sliderBannerWrapper">BannerWrapper</div>

      <div id="sliderRecommendWrapper">RecommendWrapper</div>

</div>

b.CSS代码

<style>

#mod\_container{

    border: 10px solid #AA00BB;

 }

#sliderBannerWrapper{

    width: 400px;

    height: 470px;

    float: left;

    background-color: #ADFF2F;

}

#sliderRecommendWrapper{

    width: 194px;

    height: 470px;

    float: left;

    background-color:#A6C6FF;

 }

 clearfix{zoom:1}

.clearfix:after{

    content:".";

    display:block;

    visibility:hidden;

    height:0;

    clear:both;

 }

</style>

4.相关属性说明

<style>

    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; \*/*

         }

</style>

# 2.HTML对Checkbox操作

### 1.Jqueyr对checkbox的操作

1.如何选择checkbox:使用selector.prop("checked","checked")

2.判断checkbox是否选中:selector.is(':checked')

3.获取checkbox选项的个数:selector.length;

4.获取勾选的checkbox选项个数:$("selector:checked").length;

5.获取未勾选的checkbox选项个数:$("selector:not(:checked)").length;

6.默认勾选：添加属性：checked="checked"

7.正常能勾选的checkbox:$("selector:not(:disabled)");

### 2.默认样式

<script>

   $(function(){

*//全选/取消全选*

    $(".default-checkbox .checkbox.all").click(function(){

     var checked = $(this).is(':checked');

     $(".default-checkbox .checkbox").length;

     if(checked){

      $(".default-checkbox .checkbox:not(:disabled)").prop("checked","checked");

     }else{

      $(".default-checkbox .checkbox").prop("checked","");

     }

    });

    $(".default-checkbox .btn-chcked").click(function(){

     var arr = new Array();

     $(".default-checkbox .checkbox:checked").each(function(){

      arr.push($(this).val());

     })

          console.log(arr);

       })

    $(".default-checkbox .btn-unchecked").click(function(){

     var arr = new Array();

     $(".default-checkbox .checkbox:not(:checked)").each(function(){

      arr.push($(this).val());

     })

          console.log(arr);

       })

   });

  </script>

<h2>默认样式</h2>

   <input type="button" value="获取勾选的值" class="btn-chcked button--plain"/>

   <input type="button" value="获取未勾选的值" class="btn-unchecked button--plain" />

  <div class="default">

    <div class="default-checkbox" >

   <input type="checkbox" value="" class="checkbox all"/>全选/取消全选

   <input type="checkbox" value="travel" class="checkbox" checked="checked"/>旅行

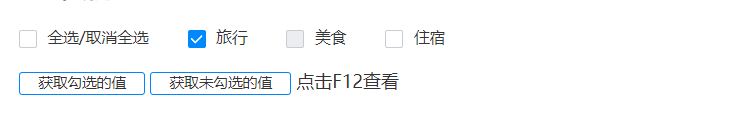
   <input type="checkbox" value="food" class="checkbox"/ disabled="disabled">美食

   <input type="checkbox" value="hotel" class="checkbox"/>住宿

    </div>

  </div>

### 3.自定义样式



<style>

   .desktop-form-check-label{

    display: inline-block;

       vertical-align: middle;

       cursor: pointer;

       margin-right: 30px;

       margin-bottom: 16px;

       font: inherit;

       font-size: 14px;

       color: #353537;

       font-weight: 500;

   }

   .desktop-form-checkbox{

    position: absolute;

    left: -9999em;

   }

   .desktop-form-checkbox:checked+.desktop-icon-checkbox:before {

       content: " ";

       width: 8px;

       height: 4px;

       border-left: 2px solid #fff;

       border-bottom: 2px solid #fff;

       border-radius: 1px;

       display: inline-block;

       vertical-align: 8px;

       margin-top: 3px;

       transform: matrix(0.71,-0.71,0.71,0.71,0,0);

       -ms-transform: matrix(0.71,-0.71,0.71,0.71,0,0);

       -webkit-transform: matrix(0.71,-0.71,0.71,0.71,0,0);

   }

   .desktop-form-checkbox:checked+.desktop-icon-checkbox{

    border: 1px solid #0088FF;

    background-color: #0088FF;

   }

   .desktop-form-checkbox:disabled+.desktop-icon-checkbox {

                background-color: #EBEDEF;

            }

   .desktop-icon-checkbox{

    display: inline-block;

    vertical-align: middle;

    width: 16px;

    height: 16px;

    margin-right: 5px;

    border: 1px solid #C9CDD3;

    box-sizing: border-box;

    overflow: hidden;

    background-color: #fff;

    font-size: 0;

    text-align: center;

    border-radius: 2px;

   }

   .desktop-form-check-content{

   }

  </style>

  <script>

   $(function(){

    $(".user-defined-css .desktop-form-check-label .checked-all").click(function(){

     var check = $(this).is(":checked");

     if(check){

     $(".user-defined-css .desktop-form-check-label .desktop-form-checkbox:not(:disabled)").prop("checked","checked");

     }else{

     $(".user-defined-css .desktop-form-check-label .desktop-form-checkbox").prop("checked","");

     }

       });

    $(".user-defined-css .btn-chcked").click(function(){

     var arr = new Array();

     $(".user-defined-css .desktop-form-checkbox:checked:not(:first)").each(function(){

      arr.push($(this).val());

     })

          console.log(arr);

       })

    $(".user-defined-css .btn-unchecked").click(function(){

     var arr = new Array();

     $(".user-defined-css .desktop-form-checkbox:not(:checked):not(:first)").each(function(){

      arr.push($(this).val());

     })

          console.log(arr);

      })

   });

  </script>

<div class="user-defined-css">

   <input type="button" value="获取勾选的值" class="btn-chcked"/>

   <input type="button" value="获取未勾选的值" class="btn-unchecked" />

   <div class="article-tags">

    <label class="desktop-form-check-label">

      <input type="checkbox" class="desktop-form-checkbox checked-all" value="">

      <i class="desktop-icon-checkbox"></i>

      <span class="desktop-form-check-content">全选/取消全选</span>

    </label>

    <label class="desktop-form-check-label">

      <input type="checkbox" class="desktop-form-checkbox" value="travel" checked="checked">

      <i class="desktop-icon-checkbox"></i>

      <span class="desktop-form-check-content">旅行</span>

    </label>

    <label class="desktop-form-check-label">

      <input type="checkbox" class="desktop-form-checkbox" value="food" disabled="disabled">

      <i class="desktop-icon-checkbox"></i>

      <span class="desktop-form-check-content">美食</span>

    </label>

    <label class="desktop-form-check-label">

      <input type="checkbox" class="desktop-form-checkbox" value="hotel">

      <i class="desktop-icon-checkbox"></i>

      <span class="desktop-form-check-content">住宿</span>

    </label>

   </div>

  </div>