目录

Part I

超文本标记语言

HyperText Markup Language, HTML

Chapter 1

HTML

Part II

层叠样式表
Cascading Style Sheets, CSS

Chapter 2

CSS

2.1 CSS颜色

2.2 元素显示模式

元素显示模式就是元素（标签）以什么方式进行显示，比如说div自己占一行，而span却可以一行内写多个。

HTML元素一般分为块元素和行内元素两种类型。

2.2.1 块元素

常见的块元素有h1-h6，p，div，ul，ol，li，其中标签是最典型的块元素。（似乎暗示自带换行的功能？）

块级元素的特点：

- 自己独占一行；
- 高度、宽度、外边距预计内边距都可以控制；
- 宽度默认是容器（父级宽度）的100%；
- 是一个容器及盒子，里面可以放行内或者块级元素；

注意：

- 文字类的元素不能使用块级元素；
- p标签主要用于存放文字，因此里面不能放块级元素；
- h1-h6都是文字类块级标签，里面也不能放其他块级元素；

2.2.2 行内元素（内联元素）

常见的行内元素有a、strong、b、em、i、del、s、ins、u、span等，其中span标签是最典型的行内元素。

行内元素的特点：

- 相邻行内元素在一行上可以显示多个；
- 高、宽直接设置是无效的；
- 默认宽度就是元素本身内容的宽度；
- 行内元素只能容纳文本或其他行内元素；

注意：

- 链接里面不能再放链接；
- 特殊情况链接a里面可以放跨级元素，但是给a转换一下块级模式最安全；

2.3 CSS背景

2.3.1 背景图像固定（背景附着）

background-attachment属性设置背景图像是否是固定或者随着页面

2.3.2 背景复合写法

当使用简写属性时，没有特定的书写顺序，一般习惯按照约定顺序为：

`backgroud: color url repeat attachment position`

2.3.3 背景色半透明

CSS3为我们提供了背景颜色半透明的效果。

```
backgroud: rgba(0, 0, 0, .3);
```

- 最后一个参数时alpha透明度，取值范围在0 – 1之间；
- 背景半透明是指盒子背景半透明，盒子里面的内容不受影响；

表 2.1: 选择器权重

选择器	选择器权重
继承或者*	0,0,0,0
元素选择器	0,0,0,1
类选择器、伪类选择器	0,0,1,0
ID选择器	0,1,0,0
行内样式style=""	1,0,0,0
!important	无穷大

2.3.4 背景总结

属性	作用	值
background-color	背景颜色	预定义的颜色值或十六进制或RGB值
background-image	背景图片	url(path)
background-repeat	是否平铺	repeat/no-repeat/repeat-x/repeat-y
background-position	背景位置	length/position 分别是x和y的坐标
background-attachment	背景附着	scroll（背景滚动） fixed（背景固定）
背景简写	书写更简单	背景颜色背景图片背景平铺背景滚动背景位置
背景色半透明	背景颜色半透明	background: rgba(red, green, blue, alpha)

2.4 CSS三大样式

CSS有三个非常重要的三个特性：层叠性、继承性、优先级。

2.4.1 层叠性

相同选择器设置相同的样式时，此时新的样式就会覆盖另一个冲突的样式，层叠性主要解决样式冲突的问题。

层叠性原则：

- 样式冲突，遵顼的原则是就近原则，哪个样式离结构近，就执行哪个样式；
- 样式不冲突，不会层叠；

2.4.2 继承性

2.4.3 优先级

Table 2.1显示了选择器的权重

1. 继承的权重是0，如果该元素没有直接选中，不管父元素权重多高，子元素得到的权重都是0。

2. a链接浏览器默认指定列一个样式：蓝色的 有下划线

权重叠加

2.5 盒子模型

2.5.1 盒子模型的组成

所谓盒子模型，就是把HTML页面中的布局元素看作是一个矩形的盒子，也就是一个盛装内容的容器。

CSS盒子模型本质是一个盒子，封装周围的HTML元素，它包括：边框（border），外边距（margin），内边距（padding）和实际内容（content）。

边框（**border**）

表格的细线边框 border-collapse属性控制连六七绘制表格边框的方式，它控制相邻单元格的边框。

语法格式为：

```
border-collapse: collapse;
```

边框会影响盒子的实际大小

内边距（**padding**）

padding属性用于设置内边距，即边框与内容之间的距离。

如果盒子本身没有指定width/height属性，则此时padding不会撑开盒子大小。

外边距（**margin**）

margin属性用于设置外边距，即控制盒子与盒子之间的距离。

属性	作用
margin-left	左外边距
margin-right	右外边距
margin-top	上外边距
margin-bottom	下外边距

清除外边距

网页元素很多都带有默认的内外边距，而且不同浏览器默认的值也不一样，因此我们在布局前，首先要清除网页元素的内外边距。

```
* {
    padding: 0;
    margin: 0;
}
```

注意：行内元素尽量只设置左右的内外边距，不要设置上下内外边距。但是转换为块级或者块元素就可以了。

2.5.2 圆角边框

border-radius属性可以设置元素的外边框圆角。
语法格式为：

```
border-radius: length|pct;
```

- 参数值可以为数值或者百分比的形式；
- 如果是正方形，想要一个圆，课数值修改为高度或宽度的一般即可，或者直接写50%；
- 如果是一个矩形，设置为高度的一般即可；
- 该属性是一个简写属性，可以写四个值，分别表示左上角、右上角、右下角、左下角；
- 分开写：border-top-left-radius、border-top-right-radius、border-bottom-right-radius、border-bottom-left-radius；

2.5.3 盒子阴影（重点）

box-shadow属性可以为盒子添加阴影。
语法格式为：

```
box-shadow: h-shadow v-shadow blur spread color inset;
```

属性	描述
h-shadow	必需，水平阴影的位置，允许负值
v-shadow	必需，垂直阴影的位置，允许负值
blur	可选，模糊的清晰度
spread	可选，阴影的尺寸
color	可选，阴影的颜色，参见CSS颜色
inset	可选，将外部阴影（默认）改为内部阴影

- 默认为外部阴影（outset），这个值不写，如果写了将导致阴影无效；
- 盒子阴影不占用控件，不会印象其他盒子排列；

2.6 文字阴影（应用不多）

text-shadow属性将阴影应用于文本。语法格式为：

```
box-shadow: h-shadow v-shadow blur color;
```

属性	描述
h-shadow	必需，水平阴影的位置，允许负值
v-shadow	必需，垂直阴影的位置，允许负值
blur	可选，模糊的清晰度
color	可选，阴影的颜色，参见CSS颜色

2.7 CSS浮动

2.7.1 标准流（普通流/文档流）

标准流就是标签按照规定好默认的方式排序。

2.7.2 浮动

网页布局第一准则：多个块级元素纵向排列找标准流，多个块级元素横向排列找浮动。

float属性用于创建浮动框，将其移动到一边，直到左边缘或右边缘触及包含块或者另一个浮动框的边缘。

语法格式为：

```
selector {
    float: value;
}
```

属性值	描述
none	元素不浮动（默认值）
left	元素向左浮动
right	元素向右浮动

2.7.3 浮动特性

1. 浮动元素会脱离标准流（拖标），浮动的盒子不再保留原先的位置；
2. 浮动元素会一行内显示并且元素顶部对齐；
3. 浮动元素会具有行内块元素的特性

- 如果行内元素有了浮动，则不需要转换为行内块元素就可以直接指定高度和宽度；
- 如果块级盒子没有设置宽度，默认宽度和父级宽度一样宽，但是添加浮动后，它的大小根据内容来决定；
- 浮动的盒子中间是没有缝隙的，是紧挨在一起的；
- 行内元素同理；

为了约束浮动元素的位置，网页布局一般采取的策略是：先用标准流的父元素排列上下位置，之后内部子元素采取浮动排列左右位置，符合网页布局第一准则。

2.8 常见网页布局

2.9 清除浮动

清除浮动的本质就是清除浮动元素造成的影响。如果父盒子本身有高度，则不需要清除浮动。清除浮动之后，父级就会根据浮动的子盒子自动检测高度，父级指定高度之后，就不会影响下面的标准流了。

语法格式为：

```
selector{
    clear: value;
}
```

属性值	描述
left	元素向左浮动
right	元素向右浮动
both	同时清除左右两侧浮动的影响

清除浮动的策略是：闭合浮动。

2.9.1 清除浮动的方法

1. 额外标签法也称为隔墙法，是W3C推荐的做法；
2. 父级添加overflow属性；
3. 父级添加after伪元素；
4. 父级添加双伪元素；

父级添加overflow属性 可以给父级添加overflow属性，将其属性值设置为hidden、auto和scroll。代码简洁，但是无法显示溢出内容。

:after伪元素法

添加双伪元素

2.10 CSS属性书写顺序（重点）

1. 布局定位属性：display/position/float/clear/visibility/overflow；
2. 自身属性：width/height/margin/padding/border/background；
3. 文本属性：color/font/text-decoration/text-align/vertical-align/white-space/break-word；
4. 其他属性(CSS3)：content/cursor/border-radius/box-shaow/text-shadow/background: linear-gradient；

2.11 页面布局整体思路

为了提高网页制作的效率，布局时通常有以下的整体思路：

1. 必须确定页面的版心（可视区）；
2. 分析页面中的行模块以及每个行模块中的列模块；
3. 一行中的列模块经常浮动布局，先确定每个列的大小，之后确定列的位置；
4. 制作HTML结构，遵循先有结构，后有样式的原则，结构最重要；

Part III

JavaScript教程

Chapter 3

ECMAscript

3.1 变量

3.2 数据类型

JavaScript

3.2.1 数据类型分类

JS把数据类型分为两类：

- 简单数据类型（Number、String、Boolean、Undefined、Null）
- 复杂数据类型（object）

数字型Number

数字型进制

数字型返回

`isNaN()` 这个方法用来判断是否是数字，如果是返回false，反之返回true。（纯数字加引号也会被认定为数字）

表 3.1: 声明变量的特殊情况

情况	说明	结果
`var age; console.log(age);`	只声明不赋值	undefined
`console.log(age);`	不声明不赋值	报错
`age = 10; console.log(age);`	不声明只赋值	10

字符串型String

字符串型可以是引号中的任意文本，可以是单引号或者双引号。
因为HTML标签中里面的属性使用的是双引号，JS这里更推荐使用单引号。

字符串转义字符

字符串长度　length

字符串拼接　字符串+任何类型=字符串

```
var age = 19;
console.log('a' + (age + 1) + 'b');
```

布尔型Boolean

Undefined与Null

3.3 运算符

3.4 流程控制

Chapter 4

DOM

Chapter 5

BOM

Part IV

Vue2与Vue3

Part V

其他

Chapter 6

散落知识点

Event 对象代表事件的状态，比如事件在其中发生的元素、键盘按键的状态、鼠标的位置、鼠标按钮的状态等等。说的通俗一点就是，event是JS的一个系统内置对象。平时无法使用，当DOM元素发生按键、鼠标等等各种事件时，系统会自动根据DOM元素触发的事件生成一个event对象。