# 一. CSS

## 1.页面组织方式 (了解)

table组织页面

div+css组织页面

## 2.div span p标签 (!掌握)

容器,可以用来包裹其他的html内容,形成一个组的内容

可以对这一组内容统一进行样式的改变和结构的调整

块级元素: 默认情况下独占一行, <div> <p> <h1>~<h6>...

行内元素: 默认情况下多个行内元素可以处在同一行当中, <span> <a> <font>...

## 3.CSS概述 (理解)

层叠样式表,可以对html页面进行美化,实现了显示内容的html代码和展示样式的css代码的分离,增强了页面的展示能力

## 4.在html页面中引入css样式的方式(四种方式) (掌握)

(1)通过标签上的style属性引入

可以使用标签上的style属性为当前标签指定具体的css样式

<div style="color:red">aaabbbccc</div>

适用于个别标签添加样式,不建议在页面中大量使用,不利于代码后期的维护

(2)通过style标签的方式引入

在页面的<head>标签中 通过在<style type="text/css"></style>标签内写css样式的方式引入css,

可以将所有的样式代码集中统一管理,初步实现了样式和html代码的分离

(3)通过链接方式引入CSS样式

在HTML页面的head标签中,通过一个<link rel="stylesheet" href="1.css" type="text/css"/>标签引入外部的css文件的内容

可以将所有的样式代码集中统一管理,真正实现了样式和html代码的分离

(4)通过import引入css

通过@import url("xxx.css") 在css文件的内部引入另一个css文件的内容 实现代码的复用

## 5.基本选择器 (掌握)

所谓选择器就是可以帮助我们在HTML中选择出想要修饰的标签

### (1)标签名选择器

格式: 标签名{...}

通过标签名选择器可以选中指定名称的标签进行样式的修饰

### (2)类选择器

class -- 通过HTML中通用的属性class, 可以为标签指定所属的类, class值相同的标签则为一类

通过 .类名{} 的方式来为这一类的标签设置样式

另外,可以为class指定多个值, 多个值之间用空格隔开, 表示一个标签属于多个类,多个类的样式会同时作用到这个标签上

### (3)id选择器

id -- 通过HTML中通用的属性id, 特点是id的值在整个HTML中是独一无二的, 可以作为标签的唯一标识

通过 #id值{} 的方式来为指定id值的标签设置样式

## 6.扩展选择器 (掌握)

基于基本选择器进行更加复杂选择的选择器

(1)后代选择器

在父选择器选择的元素内部 选中指定的子孙元素 进行样式的设置

格式: 父选择器 子孙选择器{...}

### (2)子元素选择器

在父选择器选择的元素内部 选中指定的子元素 进行样式的设置

格式: 父选择器>子选择器{...}

(3)分组选择器

将多个选择器选中的元素统一进行样式的设定

格式: 选择器1,选择器2,.....,选择器n{...}

### (4)属性选择器

选择具有指定属性的元素

格式:　选择器[属性...]...{}

(5)相邻兄弟选择器

如果两个元素具有相同的父亲，且紧邻在一起，则为相邻兄弟，可以通过相邻兄弟选择器选择紧邻A元素后的B元素

格式: 大哥+小弟{}

(6)伪元素选择器

伪元素选择器选择的不是某个元素，而是元素的某个指定状态。

可以选中元素的如下状态

:link 表示元素未被点击的状态

:hover 表示光标移入的状态

:active 表示元素被点击的状态

:visited 表示元素被访问过后的状态

格式: 选择器:状态{}

## 7.盒子模型 (!!掌握)

在css中, 所有的元素都可以看成是一个盒子(框),这些框都可以具有外边距/边框/内边距,这些值的设定,会直接影响框与框之间的位置关系和框内容与框之间关系,以及边框的样式

盒子模型

外边距

margin 统一的设置外边距

margin-top 设置顶部外边距

margin-right 设置右部外边距

margin-bottom 设置底部外边距

margin-left 设置顶左外边距

\*\*外边距垂直方向上的合并。

内边距

padding

padding-top

padding-right

padding-bottom

padding-left

边框

border

border-xxx-color

border-xxx-style

border-xxx-width

(其中xxx代表的是top left right bottom)

以上属性

可以使用属性单独指定上下左右的样式

也可以使用复合属性指定样式, 如

margin:10px 20px 30px 40px //上 右 下 左 顺时针方向

margin:10px 20px 30px // 上 左右 下

margin:10px 20px //上下 左右

margin:10px //上下左右

框(盒子)的类型补充

块级元素:

默认情况下独占一行

可为其指定宽高,如果不指定,其中高取决于内容, 由内容撑开, 宽自动填满父元素宽度;

如果指定宽高,就是指定的宽高, 不会被撑开

其中设置的内边距和边框及外边距都会起作用

行内元素:

默认情况下多个行内元素可以处在同一行

不能指定宽高,宽高只能由内容撑开

设置内边距和边框及左右外边距会起作用, 设置上下外边距无效

## 8.页面的布局方式 (!!掌握)

### (1)文档流布局

这种布局方式是按照HTML中元素声明的顺序, 由上到下由左到右按序排列, 在排列的过程中会计算外边距保持元素之间的位置

HTML文档默认就是这种布局方式, 也是我们最常用的布局方式

### (2)浮动布局

这种布局方式元素会脱离文档流, 元素将会按照指定的方向移动, 直到遇到父边框或者相邻浮动的元素的边框为止

浮动后的元素, 不管之前是块级元素还是行内元素, 都会变为inline-block类型, 这种类型的元素既具备行内元素的特性(如可以同行显示),又具备块级元素的特性(如可以指定宽高)

清除浮动

clear: left right both 指定当前元素的 左边 右边 两边 不能存在浮动元素

### (3)定位布局

position: relative absolute

元素设置为定位后, 不管是绝对定位还是相对定位, 都会先保持在原位置

### 相对定位(relative):

若一个元素为相对定位, 可以将元素相对于原来的位置进行移动, 可以通过 top left right bottom四个方向上设置相对于原位置移动的距离

相对定位的元素不会脱离文档流, 只是显示的位置发生了变化, 在原位置还占用空间, 原位置不会被其他元素占用

由于位置发生了变化, 可能会覆盖其他元素, 可以通过z-index属性来设置元素的显示顺序, z-index值大得元素会排在值小的元素上面

### 绝对定位(absolute):

如果一个元素为绝对定位, 可以将元素基于最近的已定位的祖先元素进行定位, 如果祖先元素中没有已定位的元素, 最终会基于页面的边框来定位

绝对定位的元素会脱离文档流, 原位置不再占用空间, 会被别的元素占用

由于位置发生了变化,可能会覆盖其他元素, 可以通过z-index属性来设置元素的显示顺序

总结:

定位布局适用于摆放没有太大规律的元素, 一般情况下, 会把这些元素的父元素设置为相对定位, 再把这些元素设置为绝对定位, 可以利用类似坐标的方式, 随意摆放元素的位置

9.其他的css属性 (!掌握)

display: 设置元素的显示方式

取值:

block: 块级元素的默认值,

inline: 行内元素的默认值

inline-block: 元素显示为行内元素, 元素的内容显示为块级元素

none: 隐藏元素,display设置元素隐藏后,元素就真的不存在了,也不占用空间了

visibility:

inherit visible hidden 这种方式设置元素隐藏后,元素虽然隐藏了,但是仍然占用空间

text-align: 设置元素中文本的水平对其方式

取值:left(默认) center right justify

vertical-align: 设置或检索对象内容的垂直对其方式。

top middle baseline bottom

10.easymall首页练习