CSS作用

页面美化 突出主题

CSS优势

1.页面与表现分离，网页的表现统一，容易修改

2.外部样式表可以极大的提高工作效率

3.丰富的样式使页面更加灵活

4.减少网页的代码量

5.增加网页的访问速度

6.应用独立页面的css有利于网页被搜索引擎收录

CSS语法

选择器{属性名:属性值;属性名2:属性2的值....}

标签选择器

1.与标签的写法一致

类选择器 class

.class名{}

ID选择器 id

#id名{}

属性选择器

优先级：ID选择器>类选择器>标签选择器

CSS样式写法

1.内部样式：给<head>里加<style>标签

作用：只针对当前页面生效

2.内联样式: 给标签添加一个属性（style）

作用：只针对当前标签生效

3.外部样式

<link>:在<head>标签里添加<link>标签，引入外部样式表

@import:导入样式表 在<head>标签里的<style>标签里添加@import "样式表的路径";

优先级：内联样式>内部样式>外部样式

<link>与@import的区别

1.<link>标签属于XHTML，@import属于CSS2.1;

2.使用<link>链接CSS文件先加载到网页中，再进行编译显示

3.使用@import导入的CSS，客户端显示HTML结构之后才加载CSS文件

4.@import只属于CSS2.1特有，对于不兼容CSS2.1的浏览器无效

字体：font:字体样式 字体粗细 字体大小/行高 字体;

font-family 字体

font-size:字体大小

font-style:字体样式

font-weight:字体粗细(bold bolder normal lighter)

文本：text

color:设置文字颜色

text-algin:left(左)/center（中）/right（右）/justify（两端对齐） 设置文字的水平对齐方式

text-indent:段落首行缩进

text-decoration:underline(下划线)/overline（上线）/line-through（删除线）/none（无） 设置文本修饰

text-shadow:x轴（正数偏右，负数遍左） y轴（正数偏下，负数偏上） 光晕大小 颜色 设置文字阴影

text-transform:uppercase(大写)/lowercase（小写）/capitalize（首字母大写） 控制字母

letter-spacing:字（字母）间距

word-spacing：词（单词）间距

white-space:nowrap 阻止换行，当遇到<br>换行

单行文字溢出显示省略号：overflow:hidden; text-overflow:ellipsis; white-space:nowrap

背景：background:颜色 url("图片地址") 是否重复 位置

background-color:背景颜色

background-image:背景图片

background-repeat:repeat(横向，纵向)/repeat-x（横向）/repeat-y（纵向）/no-repeat（不重复） 设置背景是否重复

background-position:left top/left center/left bottom/center top/center center/center bottomo/right top/right center/right bottom 设置背景的位置

背景的写法：

background:应文单词

background: #000000

background：rgba(255,0,0,0.7)

DIV的显示与隐藏

display:none(隐藏)/inline-block（行内块）/block（块）/inline（内联）

将内联元素转化为块级元素： display:block

将内联元素转化为行级块：display:inline-block

将块级元素转化为内联元素：display:inline

将块级元素转化为行内块：display:inline-block

列表样式：

清空默认样式：margin:0;padding:0; list-style:none;

list-style：

list-style-type:列表类型

无序列表：list-style-type:circle 列表样式空心圆

list-style-type:square 列表样式实心方块

默认：实心圆

有序列表： list-style-type:upper-roman 罗马文字

list-style-type:lower-alpha 小写英文字母

list-style-image:url("图片路径") 列表样式为图片

盒子模型：content（内容） padding（内边距） border（边框） margin（外边距）

1.盒子的实际宽度=内容的宽度+内边距的宽度（左右）+边框的宽度（左右）+外边距的宽度（左右）

盒子的实际高度=内容的高度+内边距的高度（上下）+边框的高度（上下）+外边距的高度（左右）

2. padding:上内边距 右内边距 下内边距 左内边距

padding:上内边距 左右内边距 下内边距

padding:上下内边距 左右内边距

padding:上下左右内边距

padding-left:左内边距

padding-right:右内边距

padding-top:上内边距

padding-bottom:下内边距

margin:上外边距 右外边距 下外边距 左外边距

margin:上外边距 左右外边距 下外边距

margin:上下外边距 左右外边距

margin:上下左右外边距

margin-left:左外边距

margin-right:右外边距

margin-top:上外边距

margin-bottom:下外边距

margin:0 auto 上下外边距为0； 左右自动。（作用：用于块级元素水平居中）

3. margin和padding的区别

1.margin是外边距，padding是内边距

2.盒子里有padding属性，会使盒子的内部大小发生变化（变大）

3.margin是外边距，不会使盒子的内部大小发生变化，但是它会使盒子和盒子之间的距离发生变化。

4. 要使padding（内边距）不让盒子自动发生变化，使用box-sizing解决

box-sizing:border-box/content-box

border-box:盒子实际大小和所定义的大小始终保持一致，不管盒子里是否有内边距（padding）

content-box:内容的大小始终和定义的大小保持一致，盒子的里的内边距（padding）会使盒子的实际大小发生变化。

边框：border:宽度 样式 颜色

border-width:边框的宽度

border-style:边框的样式

none：无边框

solid:实线边框

dashed:虚线边框

dotted:点线边框

double:双线边框

inset:3D嵌入边框

outset:3D突出边框

ridge:脊边界

groove:沟槽边界

border-color:边框的颜色

border-left:宽度 样式 颜色

border-right:宽度 样式 颜色

border-top:宽度 样式 颜色

border-bottom:宽度 样式 颜色

border-left-width:左边框的宽度

定位：position:fixed/absolute/relative（top,left,right,bottom属性在元素设置定位后生效。）

fixed:相对于浏览器窗口定位

absolute:绝对定位 相对于她的父级定位，如果它的父级没有定位，那么它就相对于HTML定位；

特点：不占据空间

relative:相对定位 相对于其正常位置定位

特点：占据原有空间

z-index：数字；元素的堆叠次序（数字最大，在最上层，数字越小，表示在最底层）

特点：在元素设定定位后生效。

浮动：float:left/right

特点：1.浮动元素不占据原有空间

2.如几个元素都设置浮动，那么他们会紧相邻

3.如果浮动元素过多，一行无法全部放下，那么他会自动换行（浮动元素当遇到父级框的时候会自动换行。）

4.当浮动元素中高度不同，那么在换行后会被高度大的元素卡住。

5.浮动元素向左浮动时，它后面的元素会按顺序排序，

6.浮动元素向右浮动时，它后面的元素会按倒序排序

7.浮动元素会影响它之后元素，不影响它之前的元素

清除浮动：clear:left/right/both

left:不允许左侧有浮动元素

right:不允许右侧有浮动元素

both:左右侧均不允许出现浮动元素

vertical-align:top（顶）/middle（中）/bottom（底） 垂直方向对齐

鼠标形状 cursor:pointer(小手)/help（帮助）/text（文本）/wait（等待，运行）/crosshair（十）

组合选择符

div p 后代选择器

1.以空格分开

2.匹配所有后代子元素

3.div是p的父级或者是祖父级

div>p 子元素选择器

1.用">"分开

2.匹配div的子元素p,(包含P元素内的所有元素)

3.div 是p的父级，不是祖父级

div+p 相邻兄弟选择器

1.用“+”分开

2.div和p有相同父级

3.div和p紧相邻，p在div之后

div~p 普通相邻兄弟选择器

1.用“~”分开

2.div和P有相同的父级

3.选择的是div之后的所有P元素

伪类：

语法：选择器:伪类名{声明1；声明2....}

a:link 未访问的链接样式

a:visited 访问过的链接样式

:hover 鼠标经过的样式

a:active 鼠标按下时的样式

选择器:before 在指定选择器前插入内容

选择器:after 在指定选择器之后插入内容

选择器:nth-child(n) 指定选择器的第n个元素 n还可以是表达式（2n+1）

选择器:first-child 指定选择器的第一个元素

选择器:last-child 指定选择器的最后一个元素

选择器:nth-last-child(n) 指定选择器的倒数第n个元素 n还可以是表达式（2n+1）

选择器:focus 多用于表单，指定选择器光标聚焦时的样式

选择器:first-letter 指定选择器的第一个字母（字）的样式

选择器:first-line 指定选择器的第一行的样式

隐藏一个区域块的方法有哪些：

display:none 和 visibility：hidden

他们的区别是：display:none 是在浏览器下不显示区域块，等同于没有这个区域块，不占用原有空间

visibility:hidden，是在浏览器下不显示区域块的内容，但是依然占用原有空间

CSS文字类型不生效

1.CSS文件是否引入

2.编码问题----1）HTML页面的编码 2）CSS文件的编码 @charset "utf-8";

CSS3的好处

1.以前用css2不能完成，却用javascript的动画，样式等，现在用CSS3可以简单快捷的实现

2.浏览器加载比较快

3.这些CSS3属性能被大部分的现代浏览器支持

border-image 边框图片：border-image:url("图片路径") 大小 方式

注意：1.在使用border-image之前，需要对元素设置边框（border）;

2.大小是不带单位

3.方式：stretch:拉伸，要多宽有多宽

round：圆满平铺，中间的图片重复，从两端向中间挤

repeat：平铺，从中间向两端重复排列

box-shadow 盒子阴影（text-shadow 文字阴影）：box-shadow:X轴偏移 Y轴偏移 光晕大小 扩展半径 颜色 阴影方式

X轴偏移：可以有正值和负值，正值是向右偏移，负值是向左偏移

Y轴偏移：可以有正值和负值，正值是向下偏移，负值是向上偏移

光晕大小：可选值，只有正值，正值越大，模糊效果越强

扩展半径：可选值，可以有正值和负值，正值越大，阴影的扩展越大，负值越小，阴影缩小

阴影方式：外阴影，不写的情况下，默认是外阴影，内阴影，用inset;只能放在第一个或者最后一个才生效

多个阴影的写法：用逗号分开

box-shadow:X轴偏移 Y轴偏移 光晕大小 颜色 阴影方式，X轴偏移 Y轴偏移 光晕大小 颜色 阴影方式

background 背景

线性渐变：

background:linear-gradient(方向,起始颜色 百分比,...最后颜色百分比)

方向：to right,to right top,

颜色：英文单词，十六进制，rgba

百分比：后面颜色百分比一定要大于前面颜色的百分比，否则将没有过度效果

径向渐变：

background:rodial-gradient(方向，形状 大小，起始颜色 百分比，...，最后颜色 百分比)

方向：1.at right top 2.用百分比，第一个值是水平方向的，第二个值是垂直方向，在使用大小的时候，一定要用百分比定位

形状：默认是圆形，ellipse定义椭圆

大小：1.closest-side指定径向渐变的半径长度为从圆心到离圆心最近的边；

2.farthest-side指定径向渐变的半径长度为从圆心到离圆心最远的边；

3.closest-corner指定径向渐变的半径长度为从圆心到离圆心最近的角；

4.farthest-corner指定径向渐变的半径长度为从圆心到离圆心最远的角；

颜色：英文单词，十六进制，rgba

百分比：后面颜色百分比一定要大于前面颜色的百分比，否则将没有过度效果

背景大小：

background-size:百分比/数值/cover/contain

百分比：是元素大小的百分比，比不是图片大小百分比，如果写一个值，以宽为准的等比例缩放，如果是两个值，那个第一值是宽，第二值是高

数值：如果是一个值呢，以宽为准的等比例缩放，如果是两个值，那个第一值是宽，第二值是高

cover:让背景充满整体区域，图片不会变形，溢出的图片的隐藏

contain:图片不变形，贴紧一边，直到遇到另一边，就停止放大

背景固定：background-attachent：fixed;

背景相对于浏览器窗口固定，不会随着滚动条的滚动而滚动

背景起始位置：

background-origin:border-box/padding-box/content-box

border-box:从边框开始（左上角）

padding-box:从内边距开始

content-box：从内容区域开始

背景的裁剪：

background-cilp:border-box/padding-box/content-box

border-box:边框之外的背景被裁减

padding-box:内边距之外的背景被裁减

content-box：内容区域外的背景被裁减

注意：这个的裁剪与背景起始位置有关系

多背景：用逗号分开，只能用简写的方法，background：url() no-repeat 定位，url() no-repeat 定位

变形：transform:变形();

旋转：transform:rotate(度数【数值+deg】)

transform:rotateX(度数【数值+deg】) 沿着X轴旋转

transform:rotateY(度数【数值+deg】) 沿着Y轴旋转

transform-origin设置旋转的中心位置：

语法：transform-origin：X轴 Y轴 Z轴

X轴的值：left center right length %

Y轴的值：top center bottom length %

Z轴的值：length

扭曲：transform:skew(度数【数值+deg】)

transform:skewX(度数【数值+deg】) 沿着X轴扭曲

transform:skewY(度数【数值+deg】) 沿着Y轴扭曲

位移：transform:translate(x,y) x,y都是数值，像素计

transform:translateX(值) X轴位移量

transform:translateX(值) Y轴的位移量

缩放：transform:scale(数值) 数值是指放大和缩小的倍数，1代表正常大小，小于1代表缩小，大于1代表放大

transform:scaleX(数值)

transform:scaleY(数值)

transform-style:flat/preserve-3d

flat:默认值，代表呈现2D平面效果

preserve-3d：呈现3D空间效果

简写：transform:perspective(像素) rotateX（度数) 透视效果

透视：perspective：数值 定义透视位置

数值为0；代表无透视效果

数值大于0：数值越大，代表眼睛离面的距离越远（物体的前面和后面的距离越近）

数值大于0：数值越小，代表眼睛离面的距离越近（物体的前面和后面的距离越远）

数值不能为负数，负数试为0

过渡：transition:属性 执行时间 过渡方式 延迟时间

transition-property:all all代表最终的所有属性

transition-duration：2s 过渡时间，以s（秒）为单位

transition-delay：2s 延迟时间（过渡开始之前的等待时间）

transition-timing-funciton:ease/ease-in/ease-in-out/ease-out/linear 过渡方式

动画：animation：动画名 动画的执行时间 动画的执行函数 延迟时间 动画的播放次数 定义动画的方向

animation-name:动画名

animation-duration 动画执行的时间

animation-delay 动画执行前的延迟时间

animation-timing-function：ease/ease-in/ease-in-out/ease-out/linear 定义动画的播放方式

animation-direction:nomorl/alternate 定义动画方向 alternate代表反向也有动画

animation-iteration-count:numble/infinite 定义动画播放的次数 infinite无限次循

环播放

animation-play-state:running/paused running默认值代表运行 paused代表暂停，paused之后再次开始播放，是从暂停的地方开始，并非重新开始

animation-fill-mode:forwards 定义动画停在最后一个动作上

定义动画：@keyframes 动画名{定义节点}

定义节点：1.from{} to{}

2. 百分比（从0%到100%）

CSS3属性选择器：选择器[属性 通配符 属性值]{}

选择器[属性^=val]{} 给选择器中特定属性的属性值以val开头选择器添加样式

选择器[属性$=val]{} 给选择器中特定属性的属性值以val结尾选择器添加样式

选择器[属性\*=val]{} 给选择器中特定属性的属性值包含val的选择器添加样式

CSS3伪类：

:root：代表根选择器（html）设置

:not：有否定意思，用于处not(选择器)之外的所有元素添加样式，多用于表单

:empty 空元素，只给没有内容（连一个空格都不能出现）的元素添加样式，多用于表格

:first-of-type 父元素下的某个类型的第一个子元素

:last-of-type 父元素下的某个类型的倒数第一个子元素

:nth-of-type(n) 父元素下的某个类型的第n个子元素,n还可以是表达式

:nth-last-of-type(n) 父元素下的某个类型的倒数第n个子元素,n还可以是表达式

:only-child 父元素下只有一个子元素，而且还是唯一的子元素 例:.nav a:only-child{}

:only-of-type 父元素下有多个子元素，但某个类型只有一个子元素

:target 目标选择器，链接跳转到的选择器设置样式

自定义字体：@font-face{font-family:自定义字体名字}

在使用自定义字体时步骤

1.下载iconfont字体

2.自定义字体@font-face{font-family:自定义字体名字;src:url(字体路径),url(字体路径)...}

3.使用字体，选择器{font-family:自定义字体名字;}

1.如何使图片与屏幕的高度保持一致

答：html,body{height:100%} 或者 :root,body{height:100%}

2.CSS3伪类选择器a:first-of-type与a:first-child有什么区别

答：a:first-child 父级元素的第一个子元素 如果第一个子元素为a标签的话，可以添加样式

a:first-of-type 父级元素下某个类型的第一个子元素，实例中是指父级元素下第一个a标签元素