第一部分 开发前准备：

1，WWW,Internet,W3C,ip,域名

WWW：(world wide web)

万维网，是基于Internet的信息服务系统，官方定义为"WWW is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents",简而言之，WWW是一个以Internet为基础的计算机网络，它允许用户在一台计算机上通过Internet访问另一台计算机上的信息。从技术上讲，万维网是Internet上那些支持WWW协议和超文本传送协议的客户机与服务器的集合，通过它可以访问世界各地的超文本文件，包括文字，图形，声音，动画，资料库以及各种内容。

Internet:

是由计算机连接起来的物理网络，它们依靠标准和固定的规则进行通信。

W3C:(world wide web consortium)

万维网联盟，创建于1994年，是Web技术领域最具权威和影响力的国际中立性技术标准机构。到目前为止，W3C已发布了200多项影响深远的Web技术标准及实施指南，如广为业界采用的超文本标记语言（标准通用标记语言下的一个应用）、可扩展标记语言（标准通用标记语言下的一个子集）

ip地址：

IP地址是指互联网协议地址（英语：Internet Protocol Address，又译为网际协议地址），是IP Address的缩写。IP地址是IP协议提供的一种统一的地址格式，它为互联网上的每一个网络和每一台主机分配一个逻辑地址。例如：192.168.189.1

域名：（万网）

IP地址是Internet主机的作为路由寻址用的数字型标识，人不容易记忆。因而产生了域名（domain name）这一种字符型标识。

com:商业组织公司

edu:教研机构

gov:政府部门

net:网络服务商

org:非盈利组织

cn:中国

jp:日本

uk:英国

2，网站(B/S架构项目)

是构成web的基础，指在万维网上根据一定的规则，使用html,css,java等语言制作的用于展示特定内容的相关网页的集合。

3，网页

是网站中的一个页面，是构成网站的基础，网页是纯文本文件，但是具有一定的格式，也就是HTML语言定义的格式，由于HTML被翻译为超文本标记语言，因此网页也被称为超文本文件，用于展示文本，图片，影音等内容。

4，互联网产业分类

行业服务类：

新闻资讯（网易）

信息搜索（百度，Google）

邮箱 （163）

购票类 （12306）

商务应用类：

电子商务 （天猫，京东）

人才招聘 （赶集网，智联招聘）

网络教育 （极客学院）

第三方支付 （支付宝）

其他类

由于"互联网+"概念的提出，互联网将渗入到医疗，教育，农业等传统行业

5，网站制作流程

需求分析->原型图（Axure）

|

前端设计->效果图（psd） 美工

|

前端开发->静态页面（html）

|

后台开发->接口服务（java）

|

部署运行（将网站部署到服务器上）

注意：在介绍时，尽量从实际开发着手，可以为学生展示我们公司的项目

6，网站的访问方式

通过浏览器来读取网页的内容，浏览器可以将信息的格式进行处理，将内容以一定的方式呈现到屏幕上。我们可以通过URL地址进行网站的访问

每个网页在Internet中都有一个唯一的URL地址，URL是Internet上用来指定一个位置或一个网页的标准方式，URL的语法格式如下：

协议名称://主机名称[:端口号/存放目录/文件名称]

http://127.0.0.1:8888/sg/index.jsp

http://www.baidu.com:80/xin/a.html

协议名称：浏览器默认协议为http协议

主机名称：网站所在的主机地址

端口号： 主机上存放该网站的服务器软件

存放目录：要访问网页的存放目录

文件名称：要访问的网页的名称

7，WebUI课程体系的介绍

第二部分 了解HTML：

1，html概念

超文本标记语言，是一种解释执行的文本类标记语言，是Internet上用于编写网页的主要语言。“超文本”就是指页面内可以包含图片、链接，甚至音乐、程序等非文字元素。

html也是一种规范，一种标准，它通过标记符号来标记要显示的网页中的各个部分。网页文件本身是一种文本文件，通过在文本文件中添加标记符，可以告诉浏览器如何显示其中的内容（如：文字如何处理，画面如何安排，图片如何显示等）。浏览器按顺序阅读网页文件，然后根据标记符解释和显示其标记的内容，对书写出错的标记将不指出其错误，且不停止其解释执行过程，编制者只能通过显示效果来分析出错原因和出错部位。

分析简单的HTML结构

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

<html>

<head>

</head>

<body>

</body>

</html>

1) 符号介绍

2) 元素介绍

3) 结构介绍

2，html发展历程

版本 年份

HTML1.0~2.0 1989~1991

概念不明确

HTML3 1995

浏览器战争年代，Netscape和Microsoft都在试图争霸世界

HTML4 1998

浏览器大战结束，万维网协会(W3C)解救我们，他们的计划是创建一个唯一的HTML标准。计划的关键是将HTML的结构和表现分解为两种语言，一种语言用于实现结构（HTML）一种语言用于表现（CSS）.

HTML4.01 1999

这段日子过得很惬意，HTML4.01在1999年闪亮登场，称为接下来十年中HTML"必备"版本

XHTML1.0 2001

正当我们感觉安逸的时候一个新兴事物引起所有人注意，即XML。它让HTML开始心烦意乱，最终两者结合，XHTML1.0诞生。

XHTML承诺，由于它的严格，再加上它提供的一些新方法，只要遵循这个标准，WEB的所有争端将就此平息。但是，大多数人都很讨厌XHTML，Web开发人员对HTML的灵活性更感兴趣，而不是XHTML的严格性。

HTML5

由于没有得到大家的祝福，这场婚姻的结局并不好，很快被HTML的新版本取代，即HTML5.它支持HTML4.01标准的大部分特性，还提供一些新特性。基于一些新特性，如支持类似博客的元素，新的视频和图形功能，以及一大堆用来构建web应用的功能，HTML5注定成为大家公认的标准。

3，基本语法

1）特性

可以使用.html与.htm作为html文件的后缀名（扩展名）

可以使用任意文本编辑器创建HTML，推荐使用

windows操作系统

文件名.扩展名

linux操作系统

文件名

2）注释

<!-- 注释内容 -->

3）基本概念

标签：

标签用来标记内容，也是用标签表名文本的意义。标签使用"<",">"包围。

标签分为成对标签和单标签

元素：

一个元素通常是由一个开始标签，内容，其他元素以及一个结束标签组成的

属性：

与元素相关的特性称为属性，属性由键值对组成。

<元素名 属性1=值1 属性2=值2></元素名>

coreattrs属性

大多数元素都可以使用的属性。

id 唯一标识

class 标识一类元素

style 样式

title 描述信息

规范：

元素名和属性都不区分大小写。

4）文档结构

文档类型声明:

版本和文档类型声明，版本和文档类型声明是对应的，文档类型使用DTD来指定

XML

1）严格的文档类型

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

2）宽松的文档类型

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

3）frameset

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">

3）H5文档类型

<!DOCTYPE html>

html：

标识HTML文档

head

标识HTML文档的头部

可以包含该文档的标题，文档使用的脚本，样式定义，元数据等信息

body

标识HTML文档的体部

body中的内容可以显示到浏览器中。

例如：

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html lang="en">

<head>

<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">

<title>标题</title>

</head>

<body>

</body>

</html>

5）meta元素

定义文档元数据

1，定义元数据关键字

<meta name="Author" content="licy">

<meta name="Copyright" content="版权信息">

<meta name="Description" content="描述信息">

<meta name="keywords" lang="zh-cn" content="精品图书">

<meta name="keywords" lang="en-us" content="good book">

2，报头规范

<meta http-equiv="Content-Type" content="text/html;charset=utf-8">

将会创建如下的消息头

Content-Type:text/html;charset=utf-8

<meta http-equiv="Content-Languaga" content="zh-CN">

将会创建如下的消息头

Content-Language:zh-CN

<meta http-equiv="Refresh" content="n;url=http://yourlink">

定时让网页在指定的时间n内跳转到页面http://yourlink

<meta http-equiv="Pragma" content="no-cache">

设置网页禁用浏览器缓存

将会创建如下的消息头：

Pragma:no-cache

<meta http-equiv="Cache-Control" content="no-cache">

设置网页禁用浏览器缓存

4. 颜色与大小

1）颜色

颜色由红(R)、绿(G)、蓝(B)组成，每个颜色的最低值为0(十六进制为00)，最高值为255(十六进制为FF)。十六进制值的写法为#号后跟三个或六个十六进制字符。

3位十六进制 6位16进制 rgb 颜色

#000 #000000 rgb(0,0,0) 黑色 black

#F00 #FF0000 rgb(255,0,0) 红色 red

#FFF #FFFFFF rgb(255,255,255) 白色 white

2）大小

px 像素

HTML 页面结构

CSS 页面表现

JS 页面动作

5. 标签介绍

1)块级(block)标签

<div></div>

特点：

独占一行

h1~h6 标题

特点:

1)有字体大小的设置

2)文本有加粗强调设置

3)上下文之间有较大间距

p 段落

特点：

1)独占一行

2)上下文之间具有距离

ul li 列表

特点：

1) 配合使用

2) ul li都独占一行空间

3) ul 上下文之间有很大空间

4) li与列表的样式显示(默认点状)，并且由文本缩进

<ul>

<li>列表1</li>

<li>列表2</li>

</ul>

dl dt dd（definition list）

特点：

1)dl dt dd独占一行空间

2)dl上下文之间有很大欧诺关键

3)dd有文本缩进

<dl>

<dt>列表标题</dt>

<dd>列表内容</dd>

<dd>列表内容</dd>

<dd>列表内容</dd>

</dl>

2)行内（内联inline）标签

hao123

span

特点：

1)最干净的内联标签（没有任何修饰）

2)不独占一行

a 超链接

特点:

1)不独占一行

2)点击可以发生跳转（或进行对应操作），语法颜色为绿色

3)文本具有颜色，具有下划线，指向后光标为手型，有状态的色彩提示

decoration 文本修饰标签

<u>下划线修饰</u>

<i>斜体修饰</i>

<b>加粗</b>

<em>强调</em>

<strong>加粗</strong>

<s>删除线</s>

<sup>上标</sup>

<sub>下标</sub>

img

src="图片地址"

绝对路径 http://www.baidu.com/car.jpg

http://www.briup.com/demo/one.html

相对路径 one.html ../a.jpg

第一部分

1，超链接

从一个web资源到另外一个web资源的连接

绝对路径：

每个网页都有一个唯一的地址，称为URI 统一资源定位符，也称为该网页的绝对路径。

http://ip:port/目录/文件名

相对路径：

相对于当前文档所在的路径

../imgs/a.jpg

./imgs/a.jpg

imgs/a.jpg

本地连接

file:///d:/html/index.html

超链接中不允许写本地连接

服务器路径

http://localhost:8888/test/index.html

1）a 超链接

连接状态：

未访问

已选择

已访问

<a href="">内容显示</a>

属性：

href 定义连接的目标URI

绝对路径：http://www.baidu.com

相对路径：b.html

邮件地址：mailto:licy@briup.com

锚点 ：#mao

空连接 ：javascript:void(0);

target 定义连接的目标窗口

\_blank

\_parent

\_self

\_top

自定义目标名

title 定义连接的提示信息

type 连接到任何类型的文件以供下载

<a href="../docs/myWord.doc" type="application/msword">报名表</a>

锚点：

锚点可以让用户在文档中设置标记，这些标记通常放在文档的特定主题处或顶部，然后创建到这些锚点的连接，指向网页中的特点位置。

例如：

<h2 id="section1">1,什么是万维网</h2>

<p>万维网，是基于Internet的信息服务系统，官方定义为"WWW is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents",简而言之，WWW是一个以Internet为基础的计算机网络，它允许用户在一台计算机上通过Internet访问另一台计算机上的信...</p>

<a href="#section1">查看第一部分内容</a>

2）link 文档关系连接

只能出现在head标签中，定义了当前文档和另一个资源之间的联系。

通常用于链接到外部样式表

<link rel="stylesheet" href="style.css" type="text/css">

属性：

href： 定义关系链接地址

rel: 定义当前文档与所连接文档之间的关系。

type: 文档类型

3）base 基准链接地址

设置页面中所有文档相对路径相对的基准URI。如果设定了基准链接地址，当前页面中的所有相对路径都基于该路径

<base href="http://localhost:8888/sg">

4）图片

1，图片类型

适合在网站上进行快速查看的图片格式

GIF （graphics interchange format,图形交换格式）

可以将背景设置为透明的，图片最多使用256中颜色，最适合显示色调不连续或具有大面积单一颜色的图片，如导航条，按钮，图标等。由于GIF图片中存储的颜色信息较少，因此占用空间极小，使用起来更方便。

JPEG（joint photographic experts group,联合图像专家组）

最适合摄影或连续色调图像的彩色照片，jpeg文件可以包含数百万中颜色，保证图片不失真。JPEG图片无法拥有透明的背景

PNG（portable network graphics,可移植网络图形）

PNG可以包含256种以上的颜色，并可以具有透明的背景。PNG文件可保留所有原始层，矢量，灰度，颜色和效果信息，并且在任何时候所有元素都是可以编辑的。

2，链入图片文字

<img src="" alt="">

属性：

src : 图片的源地址

title : 对图片的文字说明，当用户把鼠标放在图片上稍作停留，alt属性的值就会以浮动的形式显示出来。

width : 图片的宽度

height : 图片的高度

alt : 当图片文件找不到的时候显示的文本信息

border ：图片的边框

align ：图片和文字的对齐

当align值为left时，图片靠在最左方，周围的文字显示在右侧上方

当align值为right时，图片靠在最右方，周围的文字显示在左侧上方

当align值为top时，图片靠在最上方，周围的文字显示在上方

当align值为bottom时，图片靠在最上方，周围的文字显示在下方

hspace ：图片的水平间距

vspace ：图片的垂直间距

为图片添加链接

<a href=""><img src="" alt=""></a>

热点 area

<img src="" alt="" usemap="#myMap">

<map name="myMap">

<area shape="" coords="" href="beijing.html" alt="北京">

<area shape="" coords="" href="shanghai.html" alt="上海">

</map>

可以在一副图片的某几个部位建立链接

<area shape="" coords="" href="" alt="">

属性：

shape：default默认为整个区域,rect矩形,circle圆形,poly多边形

coords:

rect: left-x ,top-y ,right-x ,bootom-y

circle: center-x,center-y,radius

poly: x1,y1,x2,y2...xn,yn

使用DW创建热点：

2，表格

1）table

为表格添加边框的方法：

1.<**table width="400px" border="1px" cellspacing="0"**>

2.table{width:400px;border-collapse:collapse;}

th,td{border:1px solid #ccc; text-align:center;}

定义表格

属性：

border: 设置表格边框线条宽度

width: 设置表格宽度

align: 表格在页面中对其

bgcolor: 表格背景色

cellspacing: 单元格之间的间距

cellpadding: 单元格之间的衬距，单元格内容与其边框之间的空白

frame: 表格中边框线的显示

void 不显示边框

above 上边框

below 下边框

hsides 上下边框

vsides 左右边框

lhs 左边框

rhs 右边框

box 四个边框

border：四个边框

rules: 表格中分割线的显示

none 无分割线显示

groups 仅在列分组间和行分组间显示分割线

rows 仅在行间显示分割线

cols 仅在列间显示分割线

all 在所有行列间显示分割线

2）tr

定义表格行

3）th/td

th 定义单元格

td 定义内容单元格

属性：

colspan 跨列

rowspan 跨行

align 单元格水平对其

left,center,right,justify 两端对其

valign 单元格垂直对其

top,middle,bottom,baseline基准

对齐的继承

1）内容自身的设置具有最高优先级

2）th,td元素的对齐设置

3）tr,thead,tfoot,tbody元素的对齐设置

4）table元素的对齐设置具有全局性

5）默认的设置

4）caption

表格的标题

5）表格的分组显示

thead 表格头

tfoot 表格尾

tbody 表格主体

3，框架文档

一个框架文档由四部分组成，文档声明，html元素，head元素，frameset元素

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"

"http://www.w3.org/TR/html4/frameset.dtd">

1）frameset 框架集

rows

用来定义将框架水平分隔为子框架的数量和这些子框架的宽度

cols

用来定义将框架垂直分隔为子框架的数量和这些子框架的高度。

<frameset rows="10%,90%">

<frame src="">

<frameset cols="15%,85%">

<frame src="">

<frame src="">

</frameset>

</frameset>

2）frame 框架窗口

属性

src: 设置框架的初始内容

frameborder:框架窗口边框线

marginwidth: 在框架的边缘和边框的内容之间可以出现的空白（左右边）

marginheight: 在框架的边缘和边框的内容之间可以出现的空白（上下边）

srolling: 框架视图的滚动条设置

auto 必要时提供，默认值

yes 始终提供滚动条

no 不提供滚动条

noresize: 改变框架窗口大小

name: 框架名称，作为该框架的标识

target: 框架目标

\_blank

\_parent

\_self

\_top

框架名

3）iframe 内联框架

iframe允许用户在一个文本中插入一个框架，iframe元素可以使用frame元素的所有属性，实现功能也相同。

<body>

<iframe src="" frameborder="0"></iframe>

</body>

4，表单

主要用于收集来自用户的信息，并将收集的信息发送给服务器端处理程序处理。表单时客户端和服务器端传递数据的桥梁，是实现用于与服务器互动的最主要方式。

1）form

表单控件的容器

<form action="">

</form>

属性

action 设定处理表单数据URI的地址

method 设定数据传送到服务器的方式

get 将输入的数据追加到action地址后面

post将输入的数据保存到HTTP协议的报文中

name 设定表单的名称

enctype 设定表单数据的内容类型

一.application/x-www-form-urlencoded 在发送前编码所有字符（默认）

编码方式：

1）控件的名称和值都被转义，空白字符使用【+】替换，保留的字符一般都是用来实现特定的目的，例如（: / ? ; @ = & 等）。非数字和字母的字符使用%HH（这里HH表示两个十六进制数字，代表该字符的ASCII码）进行转换，

2）控件的"名称/值"对按照它们在文档数据流中出现的顺序列出来。"名称""值"使用"="分割，两个"名称/值"之间使用&隔开。

二.multipart/form-data 不对字符编码。在使用包含文件上传控件的表单时，必须使用该值。

数据分成多个部分，每个部分代表一个结构良好的控件，作为文档数据流的一部分，每一个部分都按照它们在文档数据流中出现的顺序依次发送到服务器端，并且，每一部分的边界不会出现在数据中。每一部分有一个content-desposition标题头，它的值的格式是：

Content-Disposition:form-data;name="myControl"

三.text/plain 空格转换为 "+" 加号，但不对特殊字符编码。

accept-charset 设置服务器端可以处理的字符编码

2）input 基本表单控件

<input type="text">

属性：

type 控件类型

text 单行文本框

textarea 多行文本框

Password 密码框

checkbox 复选框

radio 单选按钮

submit 提交按钮

reset 重置按钮

file 文件

hidden 隐藏域

image 图像按钮

button 普通按钮

name

控件名称，这个名称将与控件的当前值形参"名称/值"对 一同随表单提交

value

用于设定初始化，可选。

checked

单选框，复选框默认选中属性

size

当前控件的初始宽度，这个宽度以像素为单位。除非控件类型是text,password，这时宽度是整数值，表示字符的数目

maxlength

指定可以输入的字符的最大值。适用于控件类型为text,password。

3）button 按钮控件

<button></button>

属性

name 控件名称

value 控件初始值

type 控件类型

button 普通按钮

submit 提交按钮

reset 重置按钮

图片按钮

<button><img src="" alt=""></button>

4）select 下拉列表

<select name="" id="">

<option value=""></option>

<option value=""></option>

</select>

属性

name: 控件名称

size: 列表框中行的显示数量

multiple: 是否允许多选

如果select元素不包含属性size和属性multiple时，表单类型显示为菜单（组合框）

如果使用了属性size和属性multiple中的任意一个，则表单类型显示为列表框

selected: 默认选中

option 下拉列表选项

属性：

value: 定义控件的初始值。提交表单时，初始值会被提交给服务器。

optgroup 分组选项

<select name="" id="">

<optgroup label="大洋">

<option value="">太平洋</option>

<option value="">大西洋</option>

<option value="">印度洋</option>

</optgroup>

<optgroup label="大海">

<option value="">东海</option>

<option value="">南海</option>

<option value="">渤海</option>

</optgroup>

</select>

4）textarea 多行文本框

属性

name: 控件名称

rows: 定义文本框行数

cols: 定义文本框列数

warp: 是否自动换行。

off 不自动换行

hard自动硬回车换行，换行元素一同被传送到服务器中

soft自动软回车换行，换行元素不会传到服务器中

5）label 为表单控件定义标签

一些表单控件内建有标签，当内建有标签时，一般使用value属性的值作为标签，而另外一些表单控件没有标签，则直接使用文本作为标签来说明控件的意义。每个label元素都要和表单控件关联到一起

<table>

<tr>

<td><label for="username">用户名：</label></td>

<td><input type="text" id="username" name="username"></td>

</tr>

<tr>

<td><label for="passwold">密码：</label></td>

<td><input type="password" id="password" name="password"></td>

</tr>

</table>

6）fieldset 为表单添加结构

一般与legend元素配合使用，fieldset元素可以包含其他的表单控件，在这些表单控件周围画一个方框，而legend元素可以显示一个标签说明被包含的这些表单控件。

7）其他控件特性

disabled 禁用

支持该属性的控件：button,input,optgroup,option,select,textarea

1）禁止的元素不接受节点

2）禁止的控件的值不与表单一起被提交

readonly 只读

支持该属性的控件：input,textarea

1）可以接受焦点，但是不能被用户修改

2）只读元素的值可以与表单一起被提交。

2，CSS简介

CSS(cascading style sheet，层叠样式表)是描述文档怎么样被呈现的语言，使用CSS可以对HTML文档进行描述。

语法：

1)CSS属性和值之间用冒号分隔

2)CSS属性之间用分号分隔（建议每个属性后都书写分号）

3)CSS的值有多个的时候使用空格分隔

3，CSS在网页中的使用

1）内嵌式 style="css"

每个HTML元素都包含有一个style属性，使用该属性可以直接指定样式

<p style=""></p>

存在问题：

1.不方便修改

2.结构与显示不能很好的分离

3.建议在测试或个别情况下使用

2）嵌入式

CSS样式定义内容位于style元素之间，其type属性必须被定义为text/css

<head>

<style type="text/css">

</style>

</head>

3）外部引用式

可以在多个文档间共享样式表，提高效率；可以改变样式表而无须改变HTML文档。

<head>

<link rel="stylesheet" href="">

<style type="text/css">

@import "style.css";

@import url("style.css");

</style>

</head>

其他样式的导入

@import url(common.css);

4. CSS基本语法规则

选择器{css属性1:css值1; css属性2:css值2;}

1）选择器

用于选择html中的元素

html选择器（标签选择器）

即html标签，任何一个HTML元素的标签名都可以是css的选择器

p{ text-indent:10px} /\*段落第一行缩进10像素\*/

h1{ color:red}

优先级：

默认情况下，子级通常先继承父级标签属性

如果子级和父级拥有相同的属性，子级优先，就近原则

类选择器的优先级高于标签选择器

类选择器

class属性

.类名(类名不能使用数字开头，不能使用关键字来命名)

.rr{ color :red}

使用class属性来调用类名称

<p class="rr">one</p>

<p class="rr">two</p>

<p class="rr">three</p>

ID选择器

id属性

#id名

#two{background-color:green}

<p class="rr">one</p>

<p id="two" class="rr">two</p>

<p class="rr">three</p>

关联选择器

用一个空格隔开的两个或者多个单一选择器组成的字符串

它们的优先级比单一的选择器大。

table a{color:red}

定义了table中的a样式，而多table外的a样式不做改变

组合选择器

使用逗号,隔开选择器，可以减少样式表的重复声明

h1,h2,h3,h4{color:red}

伪元素选择器

是指对同一个HTML元素的不同状态的一种定义方式

HTML标签:伪元素{}

a:link{} 超链接没有任何动作前的状态

a:hover{} 光标移动到超链接上的状态

a:active{} 选中超链接时的状态

a:visited{}访问过超链接的状态

text-decoration:none 去掉下划线

2）选择器优先级

多个CSS选择器同时作用于同一个html时，声明不同的属性具有继承的关系，如果声明的是相同的属性，则以优先级高的为主。

内嵌式

> 关联选择器(后代，子代，组合(且))

> ID选择器

> 类选择器

> HTML选择器

同级别的后者覆盖前者

就近原则

3）样式

样式是零个或多个以分号分割的【属性名：属性值】列表

4）规则集

选择器 样式

选择器{属性名:属性值;属性名:属性值}

5）注释

/\*

注释内容

\*/

5. 基本布局

布局顺序

由上至下，由左至右，由内至外

块居中

要居中的块必须小于父级元素

利用margin-left margin-right的自动值来居中

布局级别：

1）行布局级（横排）

2）列布局级（浮动布局）

1.设置浮动 float

任何元素都可以浮动，可以改变普通文档流的排列方式，可以使得块元素在同一行中排列，使我们的布局更加方便，浮动是脱离文档流的，也就是其他元素视而不见。

浮动何时停止？

1. 当遇到一个浮动元素后

2. 当遇到父级元素后

多个盒子都浮动后，就产生了块级元素水平排列的效果

多个浮动元素不会相互覆盖

若包含的容器太窄，无法容纳水平排列的多个浮动元素，那么最后的浮动盒子会向下移动，但如果浮动元素的高度不同，那么它们向下移动时可能会被卡住。

float:

left 元素向左浮动。

right 元素向右浮动。

none 默认值。元素不浮动，并会显示在其在文本中出现的位置。

2. 清理浮动

clear

left 在左侧不允许浮动元素

right 在右侧不允许浮动元素

both 在左右两侧均不允许浮动元素

margin上下值在行布局时共用，浮动后不再共用

clear 清除浮动对象对当前对象的影响

3. 最佳实践：

1) 假如某个div元素A是浮动的，如果A元素上一个元素也是浮动的，那么A元素会跟随在上一个元素的后边(如果一行放不下这两个元素，那么A元素会被挤到下一行)；如果A元素上一个元素是标准流中的元素，那么A的相对垂直位置不会改变，也就是说A的顶部总是和上一个元素的底部对齐。div的顺序是HTML代码中div的顺序决定的

2) 高度尽量给子级，父级高度自动

3) 子级浮动，父级未浮动，父级高度无法依赖子级，这时父级需要自己添加高度（overflow:hidden）。

4) margin上下值在行布局时共用，浮动后不再共用

5) clear 清除浮动对象对当前对象的影响

1. 合理化使用标签

1) 尽量用块来包含内联

2) 无法确定使用哪个标签的时候就使用div

3) 背景与图片的区别

背景：为了修饰使用背景图像;不占据空间

background-image

图片：通常由宣传意义的图;占据空间

<img src="">

为了避免图像周围有空隙，建议将图像定义为块元素 img{display:block}

4) 导航（横向）通常使用无序列表ul li布局（层级较多，效果好加）

5) dd/dt/dd 菜单列表（纵向）

6) 用语义的方式安排标签，建议在div内包含h1,p标签来布局，将div作为布局的父元素

7) div浮动后，宽度随着内容的增加而增加

2，CSS常见的样式属性和值 (dreamweaver)

1）CSS 尺寸属性

height 设置元素高度。

width 设置元素的宽度。

max-height 设置元素的最大高度。

max-width 设置元素的最大宽度。

min-height 设置元素的最小高度。

min-width 设置元素的最小宽度。

2） 字体属性

font-family字体族科

宋体 SimSun 黑体 SimHei

微软雅黑 Microsoft YaHei

微软正黑体 Microsoft JhengHei

新宋体 NSimSun

新细明体 PMingLiU

细明体 MingLiU

标楷体 DFKai-SB

仿宋 FangSong

仿宋\_GB2312 FangSong\_GB2312

楷体\_GB2312 KaiTi\_GB2312

font-size 字体大小

font-style 字体风格

normal 正常;italic 斜体;oblique 倾斜

font-weight字体加粗

normal 正常;bold 粗体;bolder 更粗;lighter 更细

text-decoration 规定添加到文本的修饰 ：

none 默认。定义标准的文本。

underline 定义文本下的一条线。

overline 定义文本上的一条线。

line-through定义穿过文本下的一条线。

blink 定义闪烁的文本。

3） 颜色

color 设定文本的颜色

opacity 设置透明度

所有浏览器都支持 opacity 属性。

注释：IE8 以及更早的版本支持替代的 filter 属性。

例如：filter:Alpha(opacity=50)。

一般两个属性一起写，保证兼容性

opacity：0.5;

filter:Alpha(opacity=50)

4） 背景

设置元素的背景颜色。background-color:#CCC;

background-image

设置元素的背景图像。

url(bgimage.gif);

background-repeat

设置是否及如何重复背景图像。

repeat 默认。背景图像将在垂直方向和水平方向重复。

repeat-x 背景图像将在水平方向重复。

repeat-y 背景图像将在垂直方向重复。

no-repeat 背景图像将仅显示一次。

inherit 规定应该从父元素继承 background-repeat 属性的设置。

background-attachment

设置背景图像是否固定或者随着页面的其余部分滚动。

fixed 固定

scroll 滚动

background-position

设置背景图像的开始位置。这个属性设置背景原图像（由 background-image 定义）的位置， 背景图像如果要重复，将从这一点开始。您需要把 background-attachment属性设置为 "fixed"，才能保证该属性在 Firefox 和 Opera 中正常工作。

横向关键字： left center right

纵向关键字： top center bottom

百分比：

左上角是 0% 0%。右下角是 100% 100%。

background

简写属性在一个声明中设置所有的背景属性。如果不设置其中的某个值，也不会出问题，比如 background:#ff0000 url('smiley.gif'); 也是允许的。

图片精灵技术：

也叫CSS背景图片精灵技术，是将多张背景图片放到一个图片中，通过定位的方式来获取响应位置的图片，使得一个图片一次载入，多次使用，使得页面下载速度加快

5） 边框属性

border-style 设置4个边框的样式

dotted 定义点状边框。在大多数浏览器中呈现为实线。

solid 定义实线。

double 定义双线。

none 定义无边框

...

如果4个值都给定了，分别应用于上，右，下;

左如果给定1个值，应用于各边;

如果给定2个值，第一个值应用于上下边，第二个值应用于左右边

border-width 设置4个边框的宽度

值为关键字或者长度 medium,thin,thick

border-color 设置边框颜色

border 在一个声明设置所有的边框属性。

border:1px solid #ff0000

6） 鼠标光标属性

cursor 属性规定要显示的光标的类型（形状）。

none 无

auto 默认。浏览器设置的光标。

pointer 光标呈现为指示链接的指针（一只手）

wait 此光标指示程序正忙（通常是一只表或沙漏）。

help 此光标指示可用的帮助（通常是一个问号或一个气球）。

7） 列表属性

list-style 在一个声明中设置所有的列表属性。

list-style: square inside url('/i/eg\_arrow.gif')

list-style-image 将图象设置为列表项标记。

list-style-image:url("/i/arrow.gif");

list-style-position 设置列表项标记的放置位置。

inside 列表项目标记放置在文本以内，且环绕文本根据标记对齐。

outside 默认值。保持标记位于文本的左侧。列表项目标

记放置在文本以外，且环绕文本不根据标记对齐。

list-style-type 设置列表项标记的类型。

none 无标记。

disc 默认。标记是实心圆。

circle 标记是空心圆。

square 标记是实心方块。

decimal 标记是数字。

ower-roman 小写罗马数字(i, ii, iii, iv, v, 等。)

upper-roman 大写罗马数字(I, II, III, IV, V, 等。)

lower-alpha 小写英文字母The marker is lower-alpha (a, b, c, d, e, 等。)

upper-alpha 大写英文字母The marker is upper-alpha (A, B, C, D, E, 等。)

lower-latin 小写拉丁字母(a, b, c, d, e, 等。)

upper-latin 大写拉丁字母(A, B, C, D, E, 等。)

8）表格

优先级：

td,th-->tr-->tbody,thead,tfoot-->table

color,font-size

text-align 文字对齐

background

border 边框，只能用于table,th,td

margin 间距，只能用于table,caption

padding 内间距，只能用于th,td

width 宽，只能用于table,td,th

height 高，只能用于table,td,th、可以用于tr并且优先级高于td

border-collapse:

separate 默认值，分开

collapse 边框合并

inherit 从父元素中继承该属性

table-layout 宽度类型：

fixed 固定宽度

auto 自动宽度

caption-side 标题位置：

top/left/right/botton

3. div+css

div+css是一种网页的布局方法。

标签div span无特殊含义，配合css才具有特殊的样式

网页就是由许多个盒子通过不同的排列方式堆积而成，网页上每个元素都被浏览器看成是一个矩形的盒子，这个盒子由元素的内容，填充，边框，边界组成。默认盒子边框无，背景色透明，默认看不到盒子

1） 盒子模型

margin 外边距，定义区块外边界与上级元素距离的属性，值为长度

padding 内边距（填充），是设置区块的内边距的属性，是边框和元素内容之间的间隔距离

border 边框，在一个声明设置所有的边框属性。

width 盒子的宽度

height 盒子的高度

内容 盒子里面所包含的元素和内容

属性值：

1个：上下左右都是该值

2个：前者表示上下的值，后者表示左右的值

3个：前者表示上边的值，中间表示左右的值，后者表示下边的值

4个：上右下左，顺时针排序

2） 不同浏览器解析盒子模型的差异：

IE5盒子 width = 内容 + border + padding

盒子占据的宽度 = margin\*2+width

盒子占据的高度 = margin\*2+height

盒子实际的宽度 = width

盒子实际的高度 = height

W3C盒子 width = 内容

盒子占据的宽度 = margin\*2+border\*2+padding\*2+width

盒子占据的高度 = margin\*2+border\*2+padding\*2+height

盒子实际的宽度 = border\*2+padding\*2+width

盒子实际的高度 = border\*2+padding\*2+height

3） 设置浏览器去遵循w3c标准

只需要在网页的顶部加上DOCTYPE声明即可

!important的使用

p{

color:red !important;

color:blue;

}

当不加!important;的时候，后者覆盖前者，当加上之后说明第一个,样式优先级更高，采用前者，但是Ie6不支持!important;

4）其他属性

border-width 边框高度

border-color 边框颜色

border-style 边框样式

none 无样式

dotted 点线

dashed 虚线

solid 实线

double 双线

groove 槽线

ridge 脊线

inset 内凹

outset 外凸

5）关于填充和边框的常见问题

1. 大部分的html元素的盒子属性（margin,padding）默认值为0，有少数html元素的（marigin，padding不为0）例如：body,p,ul,li,form等标签，有时需要将其先设置为0

2. 相邻两个兄弟元素的外边框会发生合并，一般布局会设定他们的外边距

3. 如果没有设置父级元素的内边距或边框，那么它的子元素的边界会和其合并。

4. 设置一个块元素居中： marigin:0 auto;

5. margin可以设置负值，padding不可以设置

6. 行内元素的margin值，只有左右值，没有上下值

6）行内元素与块级元素

行内元素

行内元素不可以设置宽（width）和高（height【但是可以通过line-height设置】），但可以与其他行内元素位于同一行，行内元素内一般不可以包含块级元素。行内元素的高度一般由元素内部的字体大小决定，宽度由内容的长度控制。常见的行内元素有:em,font,b,span,a,strong

块级元素

块状元素排斥其他元素与其位于同一行，可以设定元素的宽（width）和高（height），块级元素一般是其他元素的容器，可容纳块级元素和行内元素。常见的块级元素有div, p ,h1~h6等

IE6/7及IE8混杂模式中，text- align:center可以使块级元素也居中对齐。其他浏览器中，text-align:center仅作用于行内内容上。

改变元素类型：

display

可以将一个行级元素转换为块级元素，但是这种转换并不能改变元素本质，转换的只是CSS的盒子的外观

需要转换盒子类型的情况：

水平的列表菜单，不断行的标题，隐藏元素

none 隐藏元素。不会被显示，不占空间

block 块级元素。独占一行空间

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 兼有块级和行级元素特性，在行内显示但是可以设定宽高

list-item 此元素会作为列表显示。

table 此元素会作为块级表格来显示（类似 <table>），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 <table>），表格前后没有换行符。

table-row-group 此元素会作为一个或多个行的分组来显示（类似 <tbody>）。

table-header-group 此元素会作为一个或多个行的分组来显示（类似 <thead>）。

table-footer-group 此元素会作为一个或多个行的分组来显示（类似 <tfoot>）。

table-row 此元素会作为一个表格行显示（类似 <tr>）。

table-column 此元素会作为一个单元格列显示（类似 <col>）

table-cell 此元素会作为一个表格单元格显示（类似 <td> 和 <th>）

table-caption 此元素会作为一个表格标题显示（类似 <caption>）

inherit 规定应该从父元素继承 display 属性的值。

float,position

应用了浮动和绝对定位的元素，变成了块级元素，因此display属性一般被忽略

如果元素应用了display:none，该元素（以及子元素）被隐藏起来，对其再使用float,position将不再有意义

4.和页面布局相关的CSS属性

1）定位属性：

绝对定位 absolute:

是元素完全脱离文档流，页面中的其他元素视它不存在，绝对定位元素不会影响到其他元素。绝对定位是参照距离他最近的父级有定位属性的元素，如果父级元素没有定位属性，则会参照文档

相对定位 relative:

是参照元素原来的位置进行移动，元素原来的空间位不变，元素在移动时会覆盖其他元素。

固定定位 fixed:

将元素固定在窗口中的某个位置，绝对定位是相对于元素的，固定定位是相对于窗口的，会随着窗口的一定而移动。在低版本IE中不支持该属性

默认定位 static:

默认属性，指定了元素按照常规的文档流进行定位，静态定位元素不允许使用top,left和类似其他属性定位。position的值为static为非定位元素，为其他值时候为定位元素，因为static元素不能自定义元素的位置，而其他可以

top: 层距离页面顶点纵坐标的距离

left: 层距离页面定点横坐标的距离

text-align: 横向排列 left right center

line-height:行高，内容都在行的中间，可以使用这个属性设置内容垂直居中

z-index ： 第三个维度 ，值高的元素会覆盖值低的元素。该属性可以使得元素脱离文档流，可能会覆盖其他元素。

float: 设置区块浮动属性

clear: 指定一个元素是否允许有元素漂浮在它的旁边

4，盒子模型的浮动

5，css规范

1）overflow: 设置层内的内容超出层所能容纳的范围处理方式。

visible :默认值，如果需要，内容可以溢出并绘制在元素的边框的外面

hidden :裁剪掉和隐藏溢出的内容。

scroll :元素一直显示水平和垂直滚动条，如果内容超出元素尺寸，允许用户通过滚动来查看额外的内容。

auto :滚动条只在内容超出元素尺寸时显示，而非一直显示。

2）display : 属性设置元素如何显示。

none 此元素不会被显示【浏览器会认为这个元素不存在】

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。此元素会被显示为行内元素，元素前后没有换行符。

inline-block 行内块级元素

table 此元素会作为块级表格来显示（类似 <table>），表格前后带有换行符。

table-row 此元素会作为一个表格行显示（类似 <tr>）。

table-cell此元素会作为一个表格单元格显示（类似 <td> 和 <th>）

3）visiblity: 设定元素是否可见

visible 默认。元素框是可见的。

hidden 元素框不可见，但仍然影响布局【浏览器认为该元素存在，但是不显示】

4）具有padding,margin值的元素

1. h1~h6

2. dl,dd

3. ol,ul

4. form

在Chrome,Firefox,Safari,Opera,IE8中没有默认的margin,padding。但是在IE7以及以下版本有默认margin:19px 0px

表单控件一般都有默认的padding和marigin

5. p

6. body

5）IE浏览器需要注意的兼容性问题

1. IE6双倍边距（？）

在一排浮动的元素中，如果一个元素设置了margin-left在ie浏览器中就会显示第一个块有双倍的边距。

解决方案：

1）给第一个子元素设置display:inline

2）利用CSS hack单独设置在ie6下的左边距

2. IE浏览器中会自动给添加了连接的图片加边框

6）页面布局中规范

页面布局中所有的块级元素必须要设置宽高

文字和图片在一行布局时可以统一采用块元素的方式进行布局

标签不能随意嵌套

控制页面样式时，尽量少用标签选择器

图片必须设置宽高属性

7）CSS命名规范

通用基类：

base.css

或者

base.atom.css

base.reset.css

base.layout.css

公共模型

model.css

或者

model.box.css

model.list.css

model.form.css

model.content.css

页面样式

page.ccss

或者

page.index.css

整体结构

头 head/header

内容 content/container

导航 nav

主导航 mainnav

边导航 sidebar

左右导航 leftsidebar.rightsidebar

顶导航 topnav

子菜单 submenu

栏目 column

左右中 left,right,center

页面主体main

尾 footer/foot

侧栏 sidebar

8）属性书写顺序

定位属性（比如：display， position， float， clear， visibility， table-layout等）

自身属性（比如：width， height， margin， padding， border等）

文本属性（比如：font， line-height， text-align， text-indent， vertical-align等）

其他属性（比如：color， background， opacity， cursor，content， list-style， quotes等）

---行级布局----

1. div是块级元素，独占一行空间，宽度默认沾满父级元素

2. div的高度由子级元素来决定

---列级布局---

1. 浮动元素与兄弟元素之间的关系

<div class="a"></div>

<div class="b"></div>

1）a浮动，b不浮动，b钻到a下面

2）a不浮动，b浮动，b的上边界和a的下边界重叠

3）a浮动，b浮动，a,b一行显示，当行内放不下的时候b另起一行

2. 浮动元素的边距

1) a,b不浮动，a设定外边距-下，b设定外边距-上，共用（谁大用谁的）

2) a,b浮动，a设定外边距-右，b设定外边距-左，不共用（两者和是他们的边距）

margin上下值在行布局时共用，浮动后不再共用

3. 浮动元素与父级元素之间的关系

1) 层级关系

<div class="outer">

<div class="inner"></div> 60px

<div class="inner"></div> 60px

</div>

<div class="outer2">

</div>

2) 高度宽度的问题？

1.高度应该由子级元素来指定

当子级浮动的时候，无法撑开父级的高度。只能由父级自己指定（overflow:hidden）

2.宽度应该由父级元素来指定

div没有浮动，宽度默认占满父级元素

div浮动了，宽度默认由子级元素决定

4. 背景

background-color:

background-img:

background-repeat:

background-position:

背景

不占据屏幕空间

不能直接下载

图片

img

占据屏幕空间

可以在网页上右击“图片另存为”

图片精灵：

1. 设定显示图片盒子的宽和高

2. 加背景

3. 定坐标（盒子不变，边背景位置）

background-position: x y;

默认情况下 x = 0; y=0;

当x为负值背景向左移

当x为正值背景向右移

当y为负值背景向上移

当y为正值背景向下移

5. 盒子模型

1)IE盒子模型（IE5）

width = 内容+padding+border

height= 内容+padding+border

实际高度=2\*height+2\*margin

2)w3c盒子模型

width = 内容

height= 内容

实际宽度= width + 2\*border+2\*padding+2\*margin

实际高度= height+ 2\*border+2\*padding+2\*margin

6. display 显示方式（）

none 隐藏（不占据屏幕空间visibility:hidden;占据屏幕空间）

block 块级

inline 行级

inline-block 行级兼块级

行级(span)

不独占一行空间，分享一行空间

不能指定宽高

块级(div)

独占一行空间

可以指定宽高

div布局 块级

span

7. 定位元素

position

fixed 固定定位

脱离当前文档流，相对当前浏览器窗口进行布局，默认显示在坐标（0,0）位置上，不会随着网页的滚动而滚动

absolute绝对定位

脱离当前文档流，相对

1）如果父级有定位元素，相对于该父级元素进行定位

2）如果父级没有定位元素，相对于当前网页

默认显示在坐标（0,0）位置上，会随着网页的滚动而滚动

relative相对定位

相对当前元素所在位置，默认位置就在它原本应该在的位置

如果一个元素被positon修饰，并且position的值为fixed，absolute，relative其中一个，那么这个元素为定位元素。

定位元素的特权

可以使用定位属性

left

top

right

bottom

8. 属性与样式

当描述元素的表现形式的时候使用 样式

img的宽度，高度，设定表格的宽度，表格边框，表格的背景（css）

当描述元素的功能时，使用属性

<input name="" value="">

coreattr

id

class

style

title

9.overflow

1) 清除子级的浮动

overflow:hidden

2）子级溢出父级部分的内容处理方式

overflow:

hidden 隐藏

sroll 始终显示滚动条

auto 当超过时显示滚动条

11. css优先级

内嵌式 style=""

> 关联选择器(后代，子代，组合(且)).header .nav ul li

> ID选择器

> 类选择器

> HTML选择器

day06 CSS兼容性问题

第一部分：

1，浏览器默认样式

问题：

有些浏览器默认会给一些标签添加一些样式，并且不同浏览器添加的样式不相同，这样会导致我们布局的页面在不同浏览器发生错乱

解决：

清除默认样式，保证在每个浏览器样式统一

body,p,ul,h1,h2,h3,h4,h5,h6,ol,dl,dd,form,input,ul,ol{

marigin:0;

padding:0;

list-style:none;

font-weight:normal;

}

img{border:none;}

2，img标签底部间隙问题

问题：

div中包含一张图片，底部可能有2px,4px或更多的间隙，不同的font-size会影响这个间隙的大小。

解决：

1.将图片的垂直对其方式vertical-align,值为top或者bottom

2.将图片转换为块元素display:block

3.将包含图片的父容器的字体大小设置为0，font-size:0

3，img标签IE下图片有边框

问题：

html图片img加了超链接之后产生蓝色边框（IE6~10）

解决：

img{border:none;}

4，margin上下边框合并问题

问题：

两个div容器，如果同时给上下两个div都加外边框的话，会发生边框合并

div{

border: 1px solid gray;

width: 100px;

height: 100px;

}

div.one{

margin-bottom: 30px;

}

div.two{

margin-top: 50px;

}

one和two相距50px

解决：

只给一个容器调整外边框即可，不要同时给两个

5，IE6双倍边距bug

问题：

当我们给元素添加浮动的并指定左外边距的时候，IE5,6会出现双倍边距

div.outer{

width: 100px;

height: 100px;

margin-left: 10px;

float:left;

display: inline;

}

解决：

给浮动的元素指定display:inline;

6，父容器（子元素浮动）高度为0

问题：

父元素的高度不确定，且子元素个数不确定，而且还是float会引发父元素高度为0问题，浮动的子元素层级高于父元素导致撑不开父元素的高度

解决：

在使用float元素的父元素结束前加一个高度为0宽为0且有clear:both样式的空DIV

<div style="clear:both;"></div>

在父元素上添加overflow:hidden

8，IE6不支持固定定位

解决：

div{

width:100px;

height:100px;

border:1px solid gray;

position:fixed;

bottom:20px;

right:30px;

/\*---ie6---\*/

position:absolute;

top:expression(eval(document.documentElement.scrollTop+200));

}

\*html{

background-image:url(blank:about);

background-attachment:absolute;

}

9，CSS hack

.bb{

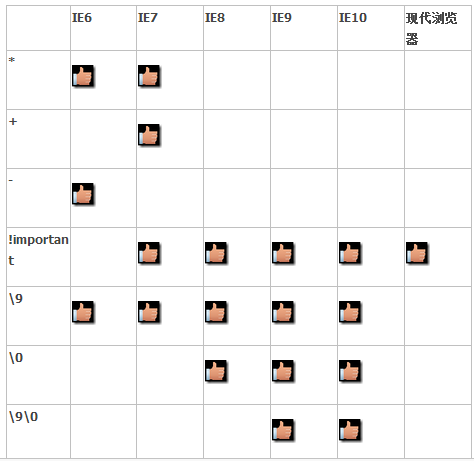
background-color:#f1ee18;/\*所有识别\*/

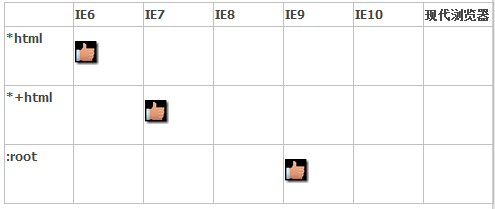
background-color:#00deff\9;/\*IE6,7,8识别\*/

+background-color:#f1ee18;/\*IE6,7识别\*/

\_background-color:#f1ee18;/\*IE6识别\*/

}





01. 块级元素，行内元素有哪些？特点是什么？

块级：

独占一行空间

指定宽高

div

h1~h6

ul ol dl

table

form

行内：

不独占一行空间

不能指定宽高

span

img

i em strong b ..

行内与块级之间的转换

display:

block

inline

inline-block

...

float:left

02. 精确计算宽高，子元素有边框，在同一行布局的时候当浏览器缩放会发生错乱，该如何解决?

<div class="content">

<div class="left"></div>

<div class="content"></div>

</div>

.content{

width:1200px;

margin:0 auto;

}

.left,right{

border:1px solid #ccc;

}

.left{

width:400px;

float:left;

}

.right{

margin-left:410px;

width:788px;

}

03. 表格的设计？

为什么要重置css样式

04. 表单元素有哪些？

<form action="login.action" method="" enctype="">

<input type="text"> val输入

password val输入

radio val

checkbox val

img

button val

hidden val

file val输入

submit val

reset val

<select name="">

<option value="">请选择</option>

<option value="1001">web1601</option>

</select>

<textarea name="" id="" cols="30" rows="10">hello</textarea>

</form>

get

1.url?key1=val1&key2=val2

2.携带数据的量 小

post

1.url

2.携带数据量 大

05. 列举三种盒子居中方式？

<div class="content">

<div class="inner"></div>

</div>

原则：

inner指定绝对宽度

不能浮动

margin:0 auto;

.inner{

width:200px;

height:100px;

margin-left:auto;

margin-right:auto;

}

计算

1000px

200px;

margin-left

绝对定位

.content{

position:relative;

}

.inner{

position:absolute;

left:50%;

margin-left:-100px;

}

相对定位

.inner{

position:relative;

left:50%;

margin-left:-100px;

}

06. 网页中引入CSS的方式？

<div style=""></div>

<head>

<style></style>

</head>

<link rel="stylesheet" href="style.css">

07. link与@import的区别？

link用于html文件中，css->html

@import用于css中， css->css

08. CSS选择器有哪些？优先级？

层级选择器

selector1 selector2

table td

tr td

table>tr

复合选择器

tr.one

tr#one

id选择器

#id

class选择器

.className

标签选择器

div{}

09. !important 与内嵌式的css优先级谁高？

div{

background:red;!important

background:blue;

}

<div style="background:gray"></div>

10. 布局的方式有几种

行级布局

列级布局

其他布局

position

由上而下，由左而右，由外而内

定位元素

能够使用定位属性 left right top bottom

position:

static(默认值)

relative 相对于原来所在位置

absolute 相对于离他最近定位父级

fixed 相对于浏览器

11. 图片精灵技术

background:url repeat x y

12. css中常见的兼容性问题以及解决方案

css hack

13. JavaScript组成？

ECMAScript（标准）

核心语法

数据类型

语句

表达式

核心api Array Object Math RegExp

DOM(H5)

HTML

BOM

14. ECMAScript中数据类型（typeof a）

基本数据类型 typeof

null object

undefined undefined

boolean boolean

number number

string string

引用数据类型

object object

function(){} function

[]

15. 数据类型之间的转换

弱类型语言

var a = 3;

var a = true;

var a = "12";

number

Number()

parseInt()

parseFloat()

+

//100px+100

var left = parseInt($(".car").position().left())+100+"px"

string

String()

toString()

boolean

var a = "false";

!!a;=>Boolean(a);

16. 短路运算符

var foo = NaN;

var bar = false;

var foo = foo||bar;

var foo ;

if(foo){

foo = foo;

}else{

foo = bar;

}

var foo = foo?foo:bar;

16.2 核心语句

分支

if(){}else{}

var a = "1";

switch(a){

case 1:

alert("周一");

break;

}

循环

for(){}

while(){}

do{}while();

for(var key in arr){}

编程语言

数据类型

表达式

语句

函数，对象

应用

17. 对象的创建，检测

1)构造函数

var obj = new Object();

2)字面量

var name = "terry",

var age = 12;

var obj = {

name:name,

age:age

};

var key = "name";

var val = "terry";

{name:"terry"}

var obj = new Object();

obj[key] = val;

var arr = new Array();

arr instanceof Array

arr instanceof Object

arr instanceof RegExp

19. 数组的定义，访问，检测，常见方法，自定义排序，遍历

var arr = new Array();//

var arr = [1,2,3,3];

arr[index] index从0开始

Array.isArray(arr);

push()

pop()

shift()

unshift()

slice()

splice()

forEach()

map()

some()

every()

filter()

...

API

1. 函数意义

2. 参数

3. 返回值（原值改变了吗）

var arr = [3,10,34,15,20,81,9];

console.log(arr)

var result = arr.sort();

console.log(arr);

arr.sort(function(a,b){

if(a < b){

return -1;

}else{

return 1;

}

});

20. 正则表达式的贪婪与非贪婪匹配，exec() test()

var pattern = new RegExp("正则","igm");

var pattern = /"正则"/igm

var str = "get-element-by-id";

// getElementById

21. 包装器类型，String的常用方法

String

Number

Boolean

var a = 3;

//自动装箱 a = new String(a);

a.toString(); //为什么可以调用

//自动拆箱

String.prototype

slice

substr

substring

split

charAt

charCodeAt

toLowerCase

toUpperCase

concat

indexOf

lastIndexOf

length 属性

函数内部属性

1)只能在函数内部访问

2)内部属性在函数调用的时候初始化

var length= 1;

function add(){

arguments

var length = 2;

console.log(this.length);

}

obj{

length:3,

add:add

}

obj2 = {length:4};

var arr = [add,add,add,1,2,3];

add(1,2,3);

obj.add(1,2,3);

add.call(obj2,1,2,3);

arr[1]();

var a = new add();

函数属性

add.length

21. JS内置函数

Object

Array

Function

RegExp

Date

Number

String

Boolean

22. 使用函数创建对象的方式

工厂模式

构造函数模式

function Person(name,age){

//obj <- this

this.name = name;

this.age = age;

this.name = function(){

alert(this.name);

};

//return obj

}

var person = new Person("terry",12);

var person2 = new Person("larry",13);

var person = Person("terry",12);

原型模式

方法

Person.prototype.sayName = function(){

alert(this.name);

}

var p = new Person();

23. 原型链继承与借用构造函数继承

借用构造函数（代码级别）

function Animal(name){

this.name = name;

}

Animal.prototype.sayName = function(){

alert(this.name);

}

function Dog(name,age){

//借用构造函数

Animal.call(this,name)

this.age = age;

}

//原型链继承 子类的原型指向父类对象

//Dog Dog.prototype

Dog.prototype = new Animal();

//天大 金毛

var dog = new Dog("一休",12);

dog.sayName();

原型链继承（内存级别）

24. DOM追加节点，替换节点，删除节点，复制节点，选择节点，节点树关系

children

firstElementChild

nextElementSibling

25. 事件捕获？事件冒泡？DOM事件流

26. 事件绑定方式？事件对象?事件代理

table>tr>td

27. 超时调用与间歇调用

28. 递归与闭包

函数与它能访问的外部函数的变量，组成了闭包,函数能访问外部函数变量的最新值

29. ajax原理，json,跨域之jsonp-

前端

数据展示

数据收集

|

|ajax

|

XMLHttpRequest request = new XMLHttpRequest();

request.open("addWork.action","post");

request.setRequestHeader();

request.send({})

request.onreadystatechange = function(){

if(request.status == 200 && request.state == 4){

request.responseText

}

}

30. 怎么理解jQuery, jQuery与js之间的关系

---------------------------

31. jQuery选择器，过滤器

32. jQueryDOM操作

33. jQuery事件处理

34. jQuery Ajax

35. jQuery 动画

36. jQuery 扩展与插件开发