## 5月30日01HTML5BASIC-Day01

1 第一阶段

HTML5基础(HTML5Basic)

CSS3基础

PROJECT1 (京东首页)

WEB UI (PS)

PROJECT2 (京东详情页)

CSS3核心

Javascript 基础

2、JS（Javascript）

3、html5，服务器端 ... ...

1、W3C

World Wide Web Consortium

万维网联盟

1、什么是WEB

WEB 就是无数 网页文档的集合

2、作用

能够将各类信息和服务进行无缝连接，并提供生动的图形用户界面

WEB - 网页

3、WEB 工作原理

基于 浏览器/服务器(B/S) 模式

互联网程序模式：

1、C/S

Client / Server

客户端 服务器

只能通过指定的客户端去访问服务器程序的一种方式

QQ, ... ...

2、B/S

Browser / Server

浏览器 服务器

通过 浏览器 去访问服务器程序的一种方式

1、组成

1、WEB 服务器

2、浏览器

用户代理(User Agent : UA)

3、通信协议

http : Hyper Text Transfer Protocal

超 文本 传输 协议

规范了数据是

如何进行打包的

如何进行传递的

请求：

响应：

2、WEB 服务器

1、功能

存储 WEB 上的信息

响应客户端的请求

2、主要服务器产品

TOMCAT

IIS

APACHE

3、WEB 浏览器

1、功能

代理用户 提交请求

作为 HTML 代码的解释器

以图形化方式显示HTML文档

2、主流的浏览器

IE

Google Chrome

Apple Safari

Mozilla Firefox

Opera

3、相关技术

1、服务器端技术

php

jsp

asp

aspx

2、客户端技术

HTML

CSS

Javascript(JS)

2、HTML 快速入门

1、HTML概述

1、什么是HTML

就是一个网页文档，WEB页面

Hyper Text Markup Language

Markup：标签，html文档内容组成的最基本方式。

Language：语言

2、HTML语法

1、标签(标记)

1、什么是标签

就是描述功能的一个符号<p></p>

2、语法

标签在使用时，必须用<>括起来

标签分为

封闭类型

非封闭类型

3、封闭类型标记-双标记

必须成对出现，一个表示开始，一个表示结束

<标签>内容</标签>

注意：

封闭类型标记，必须要有结束，否则会产生意想不到的效果

4、非封闭类型标记-单标记(空标记)

一个标签，既能表示开始，也能表示结束。没有内容可以控制。

每一个单标记，都是一个独立的功能

<标签>或<标签/>

2、元素

1、什么是元素

就是标签。

标签及其控制的内容，可称之为 元素

2、元素嵌套

在一个元素(标签)中嵌套了另外一个元素(标签)

注意：

1、嵌套顺序

2、代码规范-缩进

3、属性 和 值

属性以及值，是用来修饰元素的。

<标记 属性名称="值" 属性名称="值"></标记>

<p align="center"></p>

标准属性：

id:定义元素再文档中的唯一名称

title:鼠标移入到元素时所显示的文字

class:关联类选择器时使用

style:定义元素的内联样式

3、注释

语法：<!-- 注释内容 -->

特点：

1、不会被浏览器所解析

2、不允许被嵌套

3、不允许出现在 < 不能出现注释 >

3、HTML 版本

1、HTML4.01

2、XHTML1.0

3、HTML5

2014年10月29日 发布第一版本的规范

1、支持了更多规范

2、代码更加简洁

<p align="center"></p>

<p align=center></p>

<input readonly="readonly">

<input readonly>

4、文档结构

1、文档组成

1、文档类型声明

指定当前网页的类型和版本

html5文档类型声明：

<!doctype html>

位于 整个网页文档的最上方

2、html页面

表示 整个 网页文档的内容

位于 文档类型之下，由一对html标记组成

<!doctype html>

<html>

</html>

2、文档内容

1、网页头部

1、作用

用于定义页面的全局信息，比如：标题，描述，关键字，编码格式，定义JS，定义CSS

2、语法：

由 一对 head 标记表示

<head></head>

位于 html 根标记之内

3、子标记

1、网页标题

<title>标题内容</title>

2、编码格式

<meta charset="utf-8">

3、其他子标记

<style></style>

<link />

<script></script>

<meta name="" content="">

2、网页主体

1、作用

描述 展示给用户去看的东西

2、语法

<body></body>

位置：

<html>之内

<head>之下

3、文本标记

1、特殊字符的处理

1、空格

&nbsp;

2、<

&lt;

3、>

&gt;

4、©

&copy;

5、￥

&yen;

2、文本标记

1、文本样式

<b>...</b> 加粗

<i>...</i> 倾斜

<s>...</s> 删除线

<u>...</u> 下划线

<sub>...</sub> 下标

<sup>...</sup> 上标

2、标题元素

突出显示内容标题(改变字体，以及加粗)

<h#></h#>

#:1-6

<h1></h1> 一级标题(最大)

...

<h6></h6> 六级标题(最小)

3、段落元素

表现形式：上下会具备垂直的空白距离

<p></p>

4、换行元素

<br>

5、分割线元素

<hr>

属性：

size：尺寸

width：宽度

align：水平对齐方式 (left/center/right)

color：颜色

6、分区元素

1、行分区元素

<span></span>

作用：用于自定义文本效果

2、块分区元素

<div></div>

作用：布局

7、行内元素 块级元素

1、块级元素

独占一行

<div></div>

<p></p>

<h1></h1>

...

<h6></h6>

作用：布局

注意：

1、p元素中，不允许嵌套其他的块级元素(包含自己)

2、非布局标记，不要嵌套div

允许嵌套div

<li>

<td>

<dt>

<dd>

结构化标记

3、行内元素不要嵌套块级元素

2、行内元素

特点：多个元素会在一行内显示

作用：用来修饰文本的不同表现形式

<span></span>

<i></i>

<u></u>

<s></s>

<b></b>

...

8、预格式化

保留 源文件中的 空格 和 回车

<pre></pre>

练习1：

创建一个网页，设置标题为我的第一个html页面，并且将编码更改为 utf-8，在页面显示 Hello World

1、创建一个文本文件，并且修改后缀为 html

2、打开文件 ， 编写

1、文档类型声明

2、html 标记

3、在 html 中 编写

1、head

2、body

4、在head 中 编写

1、<title></title>

2、<meta charset="utf-8">

5、在body 中 编写

Hello World

## 5月31日01HTML5BASIC-Day02

http://code.tarena.com.cn

用户名：tarenacode

密码：code\_2013

HTML5 Day02

1、图像和链接(URL)

1、URL

1、目录 和 目录结构

1、目录

web 站点中 文件夹的名字

2、目录结构

文件所在 目录的完整表示方式

2、URL

1、什么是URL

Uniform Resource Locator

统一资源定位器，简称 路径

作用：用来表示 网络中 要用到的资源的位置

2、URL(路径)分类

1、绝对路径

1、什么是绝对路径

资源文件的地址，是从资源文件所在的最高级目录开始算起

2、本机中的绝对路径

从盘符处开始，一直到文件名称为止

E:/赵旭/01\_HTML5BASIC/Day02/jd/Images/jd\_logo.png

3、网络资源中的绝对路径

完整URL:

协议:http / https

主机名:www.baidu.com

目录结构:img

文件名:baidu\_log.png

https://www.baidu.com/img/bd\_logo1.png

2、相对路径(重点)

1、什么是相对路径

查找的资源文件的位置 是相对于 当前文件所在路径出开始出发 所经过所有的目录结构，就是相对路径

1、同目录中

当前文件 与 资源文件 在同一文件夹中

直接通过文件名称进行引用。

eg: jd\_logo.png

2、资源文件在 子目录中

解决：先进入到子目录中，再查找资源文件名称

Images/jd\_logo.png

Images/index/jd\_logo.png

3、当前文件在 子目录中，资源文件在 父目录结构下

解决：

先 后退 ../(向上一级)

再 查找

eg: ../Images/jd\_logo.png

3、根相对路径

永远都是从 网站所在的服务器 根目录下开始查找

/ : 表示服务器的根

eg: /Images/index/jd\_logo.png

4、注意

1、路径描述过程中，最好不要出现中文

2、尽量严格遵循路径的大小写方式去描述路径

以上两点，本地测试中，没有问题的。一旦到达服务器上时，必须遵循以上两点。

2、图像

1、图像格式

1、jpeg

图形图像联合专家组

\*.jpg

\*.jpeg

2、gif

图形接口格式

\*.gif

特点：动图

3、png

可移植网络图形

\*.png

特点：支持 透明背景

注意：png24位不支持透明色

2、图像元素 - <img>

语法：<img> 单标记

属性：

1、src (必须的)

要显示图像的URL

2、width

宽度

3、height

高度

注意：

1、width 和 height 如果只设置其中一个属性的话，那么图片将 "等比缩放"

2、width 和 height 赋值时，可以只给数字而不给单位 ，html中，数字的默认单位为 px(像素)

4、title

作用：鼠标移入到图像上所提示的文字

3、链接

1、链接的作用

允许用户在页面中完成“点击”并且跳转网页的行为

2、语法

<a>内容</a>

属性：

1、href

链接URL

注意：只有设置了 href 属性之后，才允许被用户点击

2、target

目标，即打开新页面的方式是什么

取值：

1、\_blank

在新标签页中打开页面

2、\_self

默认值，在自身页面中打开新页面

3、链接的表现形式

1、形式为 下载资源

设置 href 属性值 为 rar或zip等格式的文件，就可以实现下载而不是跳转

2、形式为 电子邮件链接

<a href="mailto:邮箱地址">联系我们</a>

3、返回页面顶部的 空链接

<a href="#">返回顶部</a>

4、链接到Javascript

作用：点击 链接 完成对一段Javascript代码的调用操作

<a href="javascript:js代码">JS功能</a>

4、锚点

1、作用

为了在页面的某个位置处，做一个标识，方便网页随时随地的跳转到这个 标识处上来

2、锚点的使用

1、定义锚点(做记号)

语法：<a>内容</a>

属性：

name ： 锚点的名称

2、跳转到锚点(找到锚点)

<a href="#锚点名称">内容</a>(本页)

<a href="页面url#锚点名称">内容</a>

3、id 属性 定义锚点

任何一个元素的id属性值，都可以作为一个锚点。

2、表格

1、作用

按照一定的格式来摆放数据

表格 是由 一组 "单元格"按照 从左到右 从上到下的顺序 进行排列的。

数据是保存在 "单元格" 中的

2、创建表格

1、定义表格

<table></table>

2、定义表行

<tr></tr>

注意：

1、tr 标记 要放在table中

2、一对 tr 表示的是一行数据。

3、定义 单元格(列)

<td></td>

注意：

1、td标记 要放在 tr 中

2、一对td 表示的是一个单元格

<th></th> : 文本加粗，并且水平居中。

4、默认的表格，每行单元格(列)的数量都是统一的。

3、表格的属性

1、table 属性

1、width ：宽度

2、height: 高度

3、align

控制表格在其父元素中的水平对齐方式

取值：left,center,right

4、border

设置表格的宽度

5、cellpadding

内边距：单元格的边框与内容之间的距离。

6、cellspacing

外边距：单元格与单元格之间的距离

7、bgcolor

表格的背景颜色

2、tr

1、align

所在行数据的水平对齐方式

2、valign

所在行数据的垂直对齐方式

top,middle,bottom

3、bgcolor

所在行的 背景颜色

3、td

1、align

2、valign

3、width

4、height

5、colspan ：跨列

6、rowspan ：跨行

4、表格标题

表格标题，最多只能有一个

标记：<caption>表格标题</caption>

位置：位于 table 内容中的第一行

5、复杂应用

1、行分组

1、表头行分组

<thead></thead>

2、表主体行分组

<tbody></tbody>

注意：在一个 table 中，允许出现多对 <tbody></tbody>

3、表尾行分组

<tfoot></tfoot>

2、不规则表格的实现

1、解决

通过设置 td 的 属性来完成不规则表格的创建

1、colspan ：跨列

从指定单元格位置处开始，横向向右，合并掉几个单元格(包含自己)

2、rowspan ：跨行

从指定单元格位置处开始，纵向向下，合并掉几个单元格(包含自己)

3、表格的嵌套

嵌套的表格需要放在td中

<table>

<tr>

<td></td>

<td>

<table>

<tr>

<td></td>

</tr>

</table>

</td>

<td></td>

</tr>

</table>

3、列表

1、列表的作用

按照一定的格式 显示数据

通常情况下，会将具备相似特征或前后顺序的一组内容进行排列显示。

默认格式：从上到下

2、列表的使用

1、列表的组成

每个列表由两部分组成

1、列表类型

指定列表是有序的还是无序的。

有序：<ol></ol>

无序：<ul></ul>

2、列表项

指定 具体的列表显示的内容

注意：列表项只能出现在列表中(ol,ul)

列表项：<li></li>

2、有序列表

语法：

<ol>

<li></li>

</ol>

属性:

1、type

作用：指定有序列表前的标识的类型

取值：

1 ：数字（默认值）

a : 小写英文

A : 大写英文

i : 小写 罗马 数字

I : 大写 罗马 数字

2、start

作用：指定 有序标识符的 起始编号

取值：数字

3、无序列表

1、作用、

是以没有顺序的标识符展现在列表项之前

2、语法

<ul>

<li></li>

</ul>

3、属性

1、type

指定 标识符 的类型

取值：

disc:实心圆(默认)

circle:空心圆

square:实心方块

1none:取消标识符(重点)

3、列表的复杂应用

列表的嵌套

被嵌套的内容要出现在 li 中

允许嵌套 div

4、定义列表<dl>

语法：

<dl>

<dt>独立术语</dt>

<dd>对 独立术语 的解释说明 </dd>

</dl>

使用场合：

图文混排

## 6月1日01HTMiL5BASIC-Day03

1、结构标记

1、作用

用于页面的布局，提升了 标签的 “语义化”

2、常用的结构标记

1、<header>元素

作用：用于定义页面文档的页眉(靠上方的内容部分-头部)

注意：header 元素 在页面中 允许出现多次

语法：<header></header> 取代 <div id="header"></div>

2、<nav>元素

作用：用于定义页面的导航链接部分

语法：<nav></nav> 取代 <div id="nav"></div>

3、<section>元素

作用：用于表示页面中的主体部分

注意：在页面中允许出现多次

语法：<section></section> 取代 <div id="main"></div>

4、<article>元素

作用：用于论坛的帖子、新闻的具体信息、博客/微博条目、用户评论

语法：<article></article> 取代 <div id="news"></div>

5、<footer>元素

作用：用于定义页面底部信息(页脚)

语法：<footer></footer> 取代 <div id="footer"></div>

6、<aside>元素

作用：用于定义页面内容中的边栏部分

语法：<aside></aside> 取代 <div id="left\_\*"></div>

7、注意： 能使用 结构标记的场合尽量使用结构标记。如果不适合使用 结构标记的场合，可以采用 <div id=""></div> 来使用

2、表单(重难点)

1、表单的作用

用于 显示、收集用户的信息，并将信息提交给服务器

2、表单的组成

1、表单 ：负责提交数据后的表单处理

2、表单控件：实现 能够与用户交互的可见的界面元素

3、表单元素 - <form>

1、作用

收集 表单控件 中的信息，并且提交给服务器

2、语法

<form></form>

注意：表单元素只负责提供功能，不提供界面显示效果。

属性：

1、action

作用：后台中，处理表单数据的应用程序的 URL(由后端工程师负责提供)

2、method

作用：表单提交数据的方式

常用取值：

1、get

默认值

显示提交数据，被提交的数据是可以显示在地址栏上的

不安全的：涉及到用户隐私数据时如，密码，身份证号，银行卡号等，是不能使用get提交方式的。

特点：

1、有大小限制，所提交的数据最大不能超过 2KB

2、向服务器要数据时，使用get提交方式

2、post

隐式提交数据：不会将用户的信息显示在地址栏上

安全的:有关用户的隐私数据，推荐使用 post 提交方式 进行提交。

特点：

1、无大小限制

2、让服务器去处理你的数据的时候，使用post

3、enctype

作用：指定表单数据进行编码的方式。(表单允许将什么样的数据提交给服务器)

1、application/x-www-form-urlencoded

默认值，允许将表单中的字符，数字，特殊字符，一起提交给服务器

2、multipart/form-data

只在 上传 文件 时使用

3、text/plain

只允许将普通字符、数字 提交给服务器。特殊字符不可以。

4、name

定义表单的名称

4、表单控件

1、表单控件的作用

接受用户输入的数据

2、表单控件分类

1、input元素 : <input>

1、语法

<input />

2、属性

1、type：根据不同type属性值，可以创建各种各样的输入控件(重点)

2、value : 控件的值，要提交给服务器处理程序的数据

3、name : 控件的名称，要提交给服务器处理程序的名称

4、disabled : 禁用控件，允许不给值

3、具体控件

1、文本框与密码框

文本框:type="text"

密码框:type="password"

2、属性

1、maxlength

限制输入的字符数

2、readonly

只读，允许不给值

3、name

采用 "匈牙利" 命名法

使用 ：

控件缩写 + 功能名称 组成

文本框，密码框缩写：txt

2、单选按钮 和 复选框

单选按钮:type="radio"

复选框： type="checkbox"

属性：

1、name

定义名称，分组

缩写：

单选按钮：rdo

复选框：chk

一组中，只能有一个按钮被选中

2、checked

设置默认被选中，可以不给值。

3、按钮

1、提交按钮

功能：将表单中的数据提交给服务器

type="submit"

2、重置按钮

功能：将表单的数据恢复到初始化的状态

type="reset"

3、普通按钮

功能：可通过JS自定按钮功能

type="button"

属性：

value ：定义按钮上的文本

提交注意：

1、submit 只负责提交自己所在的表单的数据，与其他表单(form)没有关系

2、submit,reset,不要脱离而独立存在，否则没有功能实现

3、只有在表单内的控件的值才会被提交，表单之外无法提交

4、隐藏域和文件选择框

1、隐藏域

type="hidden"

作用：在页面中看不到，但是允许被提交的数据。

2、文件选择框

作用：提供了选择文件的相应操作

type="file"

注意：

1、<form> 中的 method 必须为 post

2、<form> 中的 enctype 属性值 必须为 multipart/form-data

2、textarea元素

中文名：多行文本域

语法：

<textarea>显示的内容</textarea>

属性：

name : 控件的名称，服务器用

cols : 指定文本域的列数，一行内能够显示多少个字符(中文减半)

rows : 指定文本域的行数，初始化时，默认能够显示多少行的数据。

readonly:只读

3、select,option 元素

中文名：选项框

分类：

1、下拉选项框

2、滚动列表

1、语法

<select></select> :选项框

缩写：sel

属性：s

1、name ：名称

2、size : 大于1，则为滚动列表

3、multiple : 这是多选，不需要给值。(即：multiple=”multiple”)

<option></option> : 选项框中的选项

属性：

1、value ：值

2、selected ：预选中

4、其他元素

1、 <label>元素

作用：关联文本以及表单控件。关联之后，点击文本时就如同点击表单控件。

语法：

<input id ="txtName" type="text">

<label for="txtName">文本</label>

属性：

for ：表示与该元素相关联的表单控件的id值。

2、为控件分组

语法：

<fieldset></fieldset> 为控件分组

<legend>标题文本</legend> 为分组指定标题

3、其他标记

1、浮动框架

作用：在一个网页中嵌入另一个网页

标记：

<iframe>该浏览器不支持iframe</iframe>

属性：

1、src ：被引入网页的url

2、width ：宽度

3、height ：高度

4、frameborder ：浮动框架的边框，设置为0，则取消默认边框

2、摘要与细节

作用：用于 展开 和 收缩 网页内容

标记：

<details></details> 定义细节

<summary></summary> 定义细节中的标题

3、度量元素

作用：用于在页面中定义一个度量衡

标记：<meter></meter>

属性：

1、min ：范围的最小值，默认为0

2、max ：范围的最大值，默认为1

3、value ：当前度量元素上显示的值，默认为0

4、时间元素

作用：关联时间日期的不同表现形式。

标记：<time>文本</time>

属性：

datetime : 用于关联真正的时间

年月日 与 时分秒之间 用 T 作为分隔

5、高亮文本显示

功能：在标识出来的文本上，以突出的背景颜色进行显示。

标记：<mark>文本</mark>

http://www.baidu.com/?txtUsername=mingren&txtUserpwd=xuanwo&rdoGender=female&chkHobby=eat&chkHobby=happy

http://www.baidu.com/?txtUsername=mingren&txtUserpwd=xuanwo&rdoGender=male&chkHobby=eat&chkHobby=drink

http://www.baidu.com/?txtUserpwd=admin&rdoGender=female&chkHobby=drink

http://www.baidu.com/?txtUsername=admin&txtUserpwd=admin&rdoGender=male&chkHobby=drink&chkHobby=play&txtHidden=110120119

01.html?txtName=mary&rdoGender=male&chkPublic=on&btnSave=%E4%BF%9D%E5%AD%98

## 6月2日

## 02CSS3BASIC-Day01

1、表单控件 name属性 value属性

1、name 值重复的问题

<input type="radio"> name属性值可以重复

<input type="checkbox">name属性值可以重复

除以上两组控件外，其余的控件name值不允许重复

所有表单元素(input,textarea,select) 都必须设置 name属性

2、value 属性值的问题

<input type="text">

<input type="password">

<textarea>value</textarea>

以上三个控件在创建时可以不给value

其余的控件，在创建时就必须赋予value属性

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CSS3基础

1、CSS概述

1、问题

1、设置页面中所有的文本的颜色为红色

2、所有的div都变成 蓝色(blue)

3、将所有的div都变成 绿色(green)

页面中，html属性，有以下问题

1、相同的效果需要通过不同的属性来实现

<body text="red">

<font color="red">

2、属性的方式不够通用

可维护性 以及 可重用性不高

2、CSS

通过 css 方式 解决以上的问题

1、什么是CSS

CSS:Cascading Style Sheets

层叠样式表，级联样式表，通常称为：样式表

作用：定义页面中元素的样式

1、实现了内容与表现的分离

2、实现了代码的可重用性和可维护性

2、CSS 与 HTML 之间的关系

HTML:内容体现

CSS:样式的修饰

如果 CSS样式 属性 与 html属性相冲突时：

W3C建议尽量使用CSS属性来取代html属性

3、使用CSS样式表

1、使用CSS样式表的方式

1、内联方式

又称为：内联样式，行内样式

实现：将 样式 定义在 html元素style属性 中

语法：

<p style=""></p>

style属性值：由若干样式"属性"以及"值"来组成

属性:值

多个属性以及值之间，用 ; 进行分割

属性 : 值 ; 属性 : 值 ; 属性 : 值 ;

常用属性以及值:

color : 表示的是文本的颜色，取值 可以为颜色的英文表示(red,green,blue)

background-color:表示的背景颜色，取值同上

font-size : 文字大小，取值：以px为单位的数值，以pt为单位的数值

2、内部样式表

将样式定义在html文档的头部<head></head>元素中

使用步骤:

1、在<head>元素中增加一对<style>元素

2、在 <style>元素中 增加 若干 "样式规则"

样式规则：

由 选择器 和 样式声明来组成

选择器：规范了页面中哪些元素能够使用声明好的样式.

如 标签选择器 ， 由标签名称作为选择器，相当于定义了该页面中这个标签的样式。

样式声明：具体的样式，属性:值;

完整语法结构：

选择器{

属性:值;

属性:值;

...

}

3、外部样式表

将样式定义在外部的css文件中

在页面对css文件进行引入

步骤：

1、创建 css 文件(.css结尾)

2、在页面中 引入 css 文件

<link rel="stylesheet" href="引入文件url" />

在 <head>中进行引入

4、练习

1、使用内联样式 为p元素设置样式

2、使用内部样式表，为h1元素设置样式

3、使用外部样式表，为h2元素设置样式

Unknown property name : 样式属性名称 写错了

Invalid property value:样式属性值 写错了

2、CSS语法(重点，难点)

1、CSS样式特征

1、继承性

大部分的属性可以被继承

2、层叠性

为一个元素定义多个样式规则

样式规则中的样式不冲突时，多个样式将层叠为一个

3、优先级

样式冲突时，会按照不同样式的优先级来应用样式。

1(低)、浏览器缺省设置(UA Style)

2(中)、外部样式表与内部样式表

-就近原则

3(高)、内联样式

练习：

1、为 <h3>元素定义样式

2、使用内联样式为其定义字体颜色为灰色(gray)

3、在内部样式表中，为其定义字体颜色为红色(red),并且字体大小为30pt

4、在外部样式表中demo.css，定义其字体颜色为黄色(yellow)，字体大小为10pt，且背景为银灰色(silver)

5、改变内部样式表和外部样式表位置，查看效果

2、CSS基础选择器(重点,难点)

选择器作用：

匹配页面中的元素

1、通用选择器

作用：匹配页面中所有的元素

语法：\*

eg:

\*{

color:red;

}

body{

color:red;

}

2、元素选择器

别名：标签选择器，标记选择器，标签样式，标记样式，元素样式

作用：定义页面中某个指定标记的默认样式。比如：div，p ...

语法：由元素名称作为选择器

p{}

div{}

span{}

body{}

3、类选择器

作用：通过元素所附带的 class 属性值，对类选择器进行引用

语法：.类选择器名称{}

注意：名称不能以数字开头

可以将多个类选择器应用在同一个元素中-多类选择器

语法：class 属性值 是以 空格 来分开的值列表

可以将类选择器和元素选择器结合使用,目的是为了对某种元素中不同样式的细分控制-分类选择器

语法：元素选择器.类选择器{}

4、ID选择器

作用：通过元素所附带的 id 属性值，对样式进行声明定义。

语法：#ID值{}

注意：不能以数字开头

元素选择器#ID选择器{}

5、群组选择器

作用：为了定义一组选择器的公共样式

语法：以 , 隔开的选择器列表

sel1,sel2,sel3 ... {}

p,.redBack,.important,#header{

font-size:24px;

}

p{

background-color:red;

}

6、后代选择器

多于一级以及一级的层级关系：后代

子代也可以称之为后代

语法：selector1 selector2{}

在 满足 selector1 选择器的元素内，再去匹配 满足 selector2的选择器的元素

7、子代选择器

只有一级层级关系：子代

语法：selector1>selector2{}

> : 表示的是子代关系

目的：缩小匹配范围

8、伪类选择器

作用：匹配元素不同的状态

语法：

:伪类选择器{}

选择器:伪类选择器{}

分类：

1、链接伪类

目的：匹配超链接元素

:link,适用于尚未访问的超链接样式

:visited,适用于访问过的超链接的样式

2、动态伪类

:hover,适用于，鼠标悬停在元素上时的状态

:active,适用于，元素被激活(点击)时的状态

:focus,适用于，元素获取焦点时的样式。

3、目标伪类

4、元素状态伪类

5、结构伪类

6、否定伪类

3、选择器优先级-权值

元素选择器 1

类选择器 10

伪类选择器 10

ID选择器 100

内联样式 1000

选择器的权值加到一起，大的优先。

如果权值相同，以 后定义 为主。

3、尺寸与边框的属性(重点)

1、CSS单位

1、尺寸单位

% : 百分比

in : 英寸 --> 2.54cm

cm : 里面

mm : 毫米

pt : 磅 1pt=1/72in ,多用于文字大小

px : 像素,常用

em : 倍数,多用于 文字大小设置 1em相当于原始大小。2em当前字体的2倍大小。

注意：尺寸相关属性在设置值的时候，必须加单位。

2、颜色单位(颜色取值)

1、rgb(x,x,x)

x:0-255

rgb(0,0,0) : 黑色

rgb(255,0,0) : 红色

rbg(128,72,61) : ~

2、rgb(x%,x%,x%)

3、#rrggbb

由6位16进制数字组成

0-9 A-F

#000000:黑色

#ffffff:白色

#111111 ~ #eeeeee : 灰色

4、#rgb

#rrggbb中 每两位数字一样的情况下，可以使用#rgb的方式来取代

#000000 --> #000;

#aabbcc --> #abc;

#111 ~ #eee : 灰色

5、颜色的英文表示法

2、尺寸属性

1、什么是尺寸属性

用于设置元素的宽度和高度

默认尺寸：

块级元素：

宽度 ：占据父元素的 100%

高度 ：自适应

行内元素：

宽度：自适应

高度：自适应

2、宽度

属性：

width

max-width

min-width

3、高度

属性：

height

max-height

min-height

4、哪些元素允许设置尺寸属性

1、块级元素允许设置尺寸属性

2、非块级元素中，本身就具备 width和height 属性的 html元素，可以设置尺寸属性

5、溢出处理

1、什么时候会产生溢出

当使用尺寸属性限制元素框大小时，并且内容所需空间大于元素框本身空间。会导致 溢出

2、溢出处理属性

属性:

overflow

overflow-x : 横向溢出处理

overflow-y : 纵向溢出处理

取值:

1、visible ：默认值，可见的

2、hidden ：隐藏

3、scroll ：滚动条，在元素上显示滚动条，溢出时，滚动条可用，不溢出时滚动条不可用(可见)

4、auto ：自动，溢出时显示滚动条并且可用，非溢出时，不显示滚动条

3、边框属性

1、边框

属性：

1、简写方式

border:width style color;

width：边框大小(粗细)

style：边框样式(实线，虚线)

实线：solid

虚线：dashed,dotted

color：颜色

除正常颜色值之外，还可取值为 transparent(透明的，有宽度，但是不可见的边框使用)

注意：border属性 设置 四个边框的统一效果。

2、单边设置

属性：

border-方向:width style color;

方向：

top : 上

bottom : 下

left : 左

right : 右

3、单属性设置

属性:

border-属性:值;

属性：

width : 宽度

style : 样式

color : 颜色

注意：设置四个边框的某一个属性

4、单边单属性设置

border-方向-属性:值;

border-top-color:#123456;

border-bottom-style:solid;

5、边框组成

边框是由四个三角形 或 梯形组成

2、边框倒角

属性：border-radius

取值：数值/百分比;

border-radius:简写属性，设置四个角的圆角半径

单独定义：

border-top-left-radius:左上角倒角半径

border-top-right-radius:右上角倒角半径

border-bottom-left-radius:左下角倒角半径

border-bottom-right-radius:右下角倒角半径

## 6月3日

## 02CSS3BASIC-Day02

1、边框阴影

1、什么是边框阴影

为元素增加阴影

2、属性

box-shadow

取值：用空格分开的值列表

h-shadow:必须，阴影的水平偏移位置(距离),取值为正，向右偏，取值为负，向左偏

v-shadow:必须，阴影的垂直偏移位置(距离),取值为正，向下偏，取值为负，向上偏

blur:可选，模糊距离

spread:可选，阴影尺寸

color:可选，颜色

inset:值，可选值，将默认的外部阴影，改为 内部阴影

2、轮廓

1、什么是轮廓

轮廓是绘制于元素周围的一条线，位于边框之外。

2、属性

简写：outline:width style color;

outline-width:轮廓的宽度

outline-style:轮廓的样式

outline-color:轮廓的颜色

常用方式：

outline:0;

outline:none;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1、框模型(重点,难点)

1、框模型

Box Model : 框模型，盒子模型，方框属性

框模型 定义了元素框处理 内容，内边距，外边距以及边框的方式。

2、详解

1、width与height

负责处理元素 内容区域 的宽度和高度

2、增加 边框，内边距，外边距后

元素 内容区域 不会改变，但是，元素整体的占地尺寸会发生改变。

3、增加内边距

元素 内容区域 不会改变，但是，元素框内(边框内)的占地尺寸会发生改变

4、框模型介入到元素中

一旦框模型中的任何一个属性介入到元素中后，那么元素的整体占地尺寸会发生改变

元素实际宽度=左右外边距+左右边框+左右内边距+width;

元素实际高度=上下外边距+上下边框+上 下内边距+height;

3、外边距

1、什么是外边距

位于元素边框之外的距离。

2、语法

属性：margin

取值：value;

单位：

1、px ：具体数值

2、% : 宽或高的占比

3、auto : 只对左右外边距有效

4、负值 : 移动元素

数量：

value：四个方向设置统一外边距

v1 v2：v1 表示上下外边距 ，v2 表示左右外边距

v1 v2 v3：v1 表示上外边距 ，v2表示左右外边距，v3表示下外边距

v1 v2 v3 v4：v1:上外边距，v2：右外边距，v3：下外边距，v4：左外边距

单边属性：

margin-left:左外边距

margin-right:右外边距

margin-top:上外边距

margin-bottom:下外边距

1、取值为auto时

当块级元素设置宽度时，同时设置左右外边距为auto时，当前元素可以实现水平居中的位置摆放。

2、取值为负值时

目的：向相反的方向去移动元素

eg:

margin-left:50px; 向右移动50px

margin-left:-50px; 向左移动50px

通常会设置 margin-top 或 margin-left 负值，来实现元素上或左的移动

3、默认外边距

html元素中，有些元素是有默认外边距的

4、外边距合并

1、什么是外边距合并

两个垂直外边距相遇时，它们将形成一个外边距

垂直后，外边距以 较大者 为准。

5、外边距溢出

父子元素中，设置子元素的上下外边距时，某些情况下(父元素无边框，设置第一个(最后)一个子元素时)，有可能会作用在父元素上

解决方案：

1、为父元素加边框

2、设置父元素的上(下)内边距，替代子元素的上(下)外边距

3、待续... ...

4、内边距

1、什么是内边距

元素边框距离内容区域之间的空白距离就是内边距

注意：设置了元素的内边距之后，元素边框的占地区域会变大

2、语法

属性：padding

取值：

单位：

1、px ：具体像素值

2、%:占比

数量:

value : 四个方向内边距

v1 v2 : v1:上下 v2:左右

v1 v2 v3 : v1:上 v2:左右 v3:下

v1 v2 v3 v4 : 上 右 下 左

单边设置：

padding-top:

padding-bottom:

padding-left:

padding-right:

2、背景属性

1、背景

主要由 背景颜色 以及 背景图像组成

背景颜色：以单一颜色作为背景的填充

背景图像：以图片作为背景填充

2、背景属性

1、背景颜色

属性：background-color

取值：任何合法的颜色值

注意：背景颜色以及图像，默认是从元素的边框处开始填充。

2、背景图片

属性：background-image

取值：url(图像路径)

eg:background-image:url(images/xx.jpg);

3、背景重复

属性：background-repeat

取值：

1、repeat：垂直，水平方向全平铺，默认值

2、no-repeat:不平铺，只显示一次背景图像

3、repeat-x:横向重复

4、repeat-y:纵向重复

4、背景图片尺寸

属性：background-size

取值：

1、value1 value2

宽度 高度

2、value1% value2%

宽度占比 高度占比

3、cover

将背景图像扩充至足够的大，以使背景图像完全覆盖到背景区域

4、contain

将背景图扩充至足够的大，直到背景图像的宽度或高度适应元素为止。

5、背景图片固定

属性：background-attachment

取值：

1、scroll

默认值，滚动的。

2、fixed

固定的，不会随着滚动条而发生滚动。

6、背景图片定位

1、什么是背景图片定位

改变背景图在元素中的默认位置。

2、语法

属性：background-position

取值：

1、x y :x表示背景图像的水平位置偏移。x取值为正，向右偏，取值为负，向左偏。y表示背景图像的垂直偏移。y取值为正，向下偏，取值为负，向上偏。

2、x% y%

3、

x : left,center,right

y : top,center,bottom

3、CSS Sprite

将 网页中的若干小图，合并成一幅大图。从而减少http的请求次数。

7、背景属性

目的：将若干背景子属性，声明在一个属性中

语法：

background:color url(图像) repeat attachment position;

常用用法：

background:url repeat position;

3、渐变

1、什么是渐变

两种或多种的颜色之间形成平滑过渡的效果

关键：颜色 以及 颜色出现的位置--> 色标

分类：

1、线性渐变(重点)

2、径向渐变

3、重复渐变

2、语法

允许出现背景图像的地方，就可以设置渐变效果

属性：background-image

取值：

linear-gradient() : 线性渐变

radial-gradient() : 径向渐变

repeating-linear-gradient():重复线性渐变

repeating-radial-gradient():重复径向渐变

3、线性渐变

语法：

background-image:linear-gradient(angle,color-point1,color-point2,...);

angle:指渐变的方向或角度,取值可以为关键字，也可以是具体的角度值。

关键字：

to top : 向上显示渐变效果

to right : 向右显示渐变效果

to bottom : 向下显示渐变效果

to left : 向左显示渐变效果

角度:0deg ~ 360deg

0deg : to top

90deg : to right

180deg : to bottom

270deg : to left

color-point:颜色,以及颜色所出现的位置,值有 颜色 以及其 位置联合组成，中间用 空格 区分，位置可以省略。

eg :

red 20px : 20px位置为红色

green 50% : 元素中间位置处为绿色

4、径向渐变

语法：

background-image:radial-gradient([size at position],color-point,color-point,...);

5、重复渐变

repeating-linear-gradient():重复线性渐变

repeating-radial-gradient():重复径向渐变

6、浏览器兼容性

对不支持的版本的浏览器，可添加浏览器前缀实现兼容。

Firefox : -moz-

Chrome、Safari : -webkit-

Opera : -o-

background-image:linear-gradient(..);

background-image:-moz-linear-gradient(..);

background-image:-webkit-linear-gradient(..);

background-image:-o-linear-gradient(..);

4、文本格式化

## 6月6日

## 02CSS3BASIC-Day03

1.文本格式化

1、文本格式化

包含

控制字体：大小...

文本格式：颜色...

2、控制字体

1、指定字体系列

属性：font-family

取值：字体1、字体2、字体3......;

注意：字体值中包含中文或特殊字符时必须使用“”引起来

2、字体大小

属性：font-size

取值：px或pt

3、字体加粗

相当于：<b></b>

属性：font-weight

取值：

normal:正常

bold：加粗（b,strong,h# 默认值）

400-900：整百倍数的数字

4、字体样式（斜体）

相当于：<i></i>

属性：font-style

取值：

normal:正常

italic:斜体，i标记的默认值

5、小型大写字母

Aa Bb Cc

作用：将匹配元素中所有的小写英文字符全部转换为大写，但是大小与小写一样。

属性：font-variant

取值：normal:正常

small-caps:小型大写字符

6、字体综合属性

将所有的字体属性值声明在一个字体中。

属性：font

取值：font-style font-variant font-weight font-size font-family;

3、文本格式属性

1、文本颜色

属性：color

取值：合法的颜色值

2、文字排列方式（水平对齐）

属性：text-align

取值：left,right...

3、文字修饰（线条样式）

属性：text-decoration

取值：

none:无样式/无下划线

underline:下划线

overline:上划线

line-through:删除线

4、行高

作用：控制匹配元素中一行数据的高度

属性：line-height:

取值：以px为单位的数值

缺点：仅局限于一行数据

font:12px/24px

表示:字体大小/行高

【?】5、文本阴影

属性：text-shadow

取值：h-shadow v-shadow blur color

6、首航文本缩进

属性：text-indent

取值：以px为单位的距离

2.表格

1、表格的常用样式属性

1、尺寸属性

width,height

2、边框 border

3、内边距 padding 【【td中使用margin无效！！】】

4、文本和字体的相关属性

font-\*

text-\*

5、vertical-align:单元格内容的垂直对齐方式

取值：top,middle bottom

2、表格特有的样式属性

1、【【边框合并】】

属性：border-collapse

取值：

separate:默认值，分离边框

collapse:合并边框

2、边框边距

相当于:table元素的cellspacing属性

属性：border-spacing

取值：

制定一个值：水平、垂直方向相同

指定两个值：分别表示水平、垂直间距，中间用逗号“，”隔开

【注意】：仅在border-collapse为

3、显示规则

表格布局：浏览器以什么样的方法显示表格

属性：table-layout

取值：

auto:自动，列宽度由单元格内容决定。【默认值】，称为【自动表格布局】。

fixed:【固定表格布局】，列宽由属性值来确定。

【自动表格布局】：

1、单元格大小会适应内容大小

2、表格复杂时，加载较慢

3、不确定每列大小时使用

【固定表格布局】

1、单元格宽度由设置的数值为主，与单元格内容无关

2、加载时不需要计算每列的宽度，加载速度快

3、不够灵活

3.浮动（重点、难点）

1、定位

定位方式：

1、普通流定位（默认定位方式）

又称为“文档流定位”

特点：

1、块级

上下排列，每个元素独占一行

2、行内

左右排列，一行内允许显示多个

2、浮动定位（【【常用】】）

1、作用

让块级元素显示在一行

2、定义

1、将元素排除在文档流之外，即脱离文档流。意味着不再占据文档空间。

2、浮动元素会停靠在包含框的左边或右边，或者是其他已经浮动起来的元素的边缘上。

3、属性

属性：float

取值：

none:默认值，无浮动

left:左浮动

right:右浮动

4、【注意】

1、如果包含框（父元素）中无法显示所有的已浮动元素，那么最后一个将实现换行

【【2、所有的元素 浮动之后 都将变成 块级元素】】，如<span></span>行内元素不能更改宽高，但浮动之后，可以

3、浮动元素对（未浮动的父元素）的高度所产生的影响：

父元素中，包含浮动元素的话，那么高度将以未浮动元素的高度为准。若所有子元素都是浮动元素，那么父元素高度将变为0。

【解决方案】

1、直接设置父元素的高度

2、为父元素增加

overflow:hidden;属性

【【弊端】：如果有要溢出显示的内容，也会被隐藏。】

3、通过 【【清除浮动】】 的方式

4、

5、清除浮动

1、什么是清除浮动

清除当前元素之前或之后的浮动元素对其带来的影响

2、属性

clear

取值：

none:默认值，无清除效果

left:清除【之前】【左边】的浮动元素所带来的影响

right:清除之前右边......

both:清除左右两边的浮动......

3、相对定位

4、绝对定位

5、固定定位

2、浮动定位

4.显示

（下一天）

## 6月7日

## 02CSS3BASIC-Day04

1、显示

1、显示方式

1、什么是显示方式

显示方式决定了的页面上的元素的显示方式（块级、行内、行内块......）

2、如何修改显示方式？

属性：display

取值：

1、none

让生成的元素没有框，即不显示（隐藏）

会使元素脱离文档流，既不占据页面空间

2、block

让 当前元素 表现的和 块级元素一致 ，即将当前元素变成 块级元素

3、inline

让 当前元素 表现的和 行内元素一致 ，即将当前元素变成 行内元素

【注意】：尽量不要将块级元素改编成 行内元素。

4、inline-block

【行内块】：本身是行内元素，但又具备块级元素的特征 。即一行内允许显示多个元素，并【可以修改宽和高】。

2、显示效果

1、visibility属性

1、属性

visibility

2、取值

1、visible：可见的，默认值

2、hidden：隐藏，元素不可见，但依然占据页面空间

3、collapse：用在表格上，在删除表格上的一行或一列时，表格整体布局不受变化

3、面试问题

【问】解释display:none;与visibility:hidden;的不同

【答】前者不显示生成的元素，并且脱离文档流（即不占据页面空间）。后者隐藏元素，但没有脱离文档流，依然占据页面空间

2、opacity属性

1、作用

透明度

2、属性

opacity

取值：0.0（完全透明）-1.0（不透明）之间的数

3、颜色取值

rgba(r,g,b,0.0-1.0);

【注意2、3两者的区别】

3、vertical-align属性

1、设置td的垂直对齐方式

2、可以设置 行内块（img等） 元素的垂直对齐方式

给img 设置vertical-align，设置图片两端的文本相对于图片的垂直对齐方式

3、属性

vertival-align

取值：

top,middle,bottom,baseline

3、光标

1、属性

cursor

取值：

default: 默认值 ‘

pointer；(小手)

crosshair: +

text: I

wait :(等待)

help : (帮助) ?

2、列表属性

1、列表项标志

属性：list-style-type

取值：

1、none:没有

2、disc:实心圆

3、circle:空心圆

4、square:实心方块

......

2、列表项图像

属性：list-style-image

取值：url(图像路径)

3、列表项位置

属性：list-style-position

取值：

1、outside:标记位于文本之外，默认值

2、inside:标记位于文本之内

4、列表属性

属性：list-style:

取值；type url() position;

【常用方式：list-style:none;】

列表的使用场合：

从上到下的排列 或者 从左到右的排列 方式，都适合列表

3、定位属性

1、定位方式

普通流定位

浮动定位

相对定位

绝对定位

固定定位

2、定位属性

1、position

作用：指定元素的定位方式

取值：

1、static:静态的，默认定位方式，即文档流定位

2、relative:相对定位

3、absolute:绝决定位

4、fixed:固定定位

2、偏移属性

top属性：

bottom属性：

left属性：

right属性：

3、z-index

堆叠顺序

3、定位方式

1、相对定位

1、什么是相对定位？

相对于元素本身的位置去偏移某个距离

2、设置相对定位

属性：posotion

取值：relative

配合 偏移属性 实现位置调整

3、使用场合

1、【实现元素位置微调】时，使用相对定位

2、【【绝对定位】】

1、特点

1、绝对定位的元素会脱离文档流

2、【【【【绝对定位的元素会变成块级元素】】】】

3、绝对定位的定位位置

相对于 最近的 已定位的 祖先元素 来实现定位。

若元素没有已定位的祖先元素，那么他的位置就相对于最初的包含框，去实现定位

【注意】：relative、absolute、fixed中的任何一种定位方式可以称为【已定位元素】

2、属性

属性：position

取值：absolute

配合...

3、使用场合

1、实现堆叠效果时使用

2、【弹出菜单】

【【【弹出菜单一定使用的是绝对定位！！！！】】】

3、固定定位

1、什么是固定定位

将元素固定在页面的某个位置处，不会随着滚动条发生位置的改变

2、特点

脱离文档流，不占据页面空间

3、属性

position:fixed;

配合 偏移属性 实现位置的调整

4、堆叠顺序

改变层叠元素的叠放顺序

属性：z-index

取值：无单位的 数值。

【【值越大，越靠上】】

如果为负数，显示在网页正常内容之下

【注意】：

1、【必须是已定位元素】才能使用

已定位

：relative,absolute,fixed

2、【父子元素间，无法用z-index改变堆叠顺序】。父子元素间，不能改顺序，永远都是 子压父

6月8日京东项目首页

6月9、10、11日端午节

6月12、13、14日京东首页项目

6月16日京东详情页

京东蓝：#005aa0

6月16日京东详情页制作

重点：1、商品展示下图片的边框为1px与hover下边框为2px时的设置。

1.先为标签设置border:1px solid #ddd;

2.为该标签设置padding:1px;。目的是为该标签额外撑出1px的空间，以备hover下的border:2px使用

3.为该标签设置hover{

border:2px;

padding:0; /\*很重要\*/

}

2、收货地址选择处的 二级弹框 的设置：

1.设置父级元素为相对的定位点

2.

## 6月21日

## 06CSS3CORE-Day01

1、复杂选择器

1、兄弟选择器

兄弟：具备 相同父元素 的一组元素称之为兄弟元素

兄弟元素：

1、相邻兄弟选择器

1、什么是相邻兄弟选择器

匹配指定元素的相邻兄弟

相邻：下一个

2、语法：

结合符：+

结 构：选择器1 + 选择器2

eg：

p + div{}

#content + .important{}

2、通用兄弟选择器

1、什么是通用兄弟选择器

匹配指定元素下面所有的满足选择器的元素

2、语法

结合符：~

结 构：选择器1 ~ 选择器2

2、属性选择器

1、作用

允许通过元素所附带的属性及其值，来匹配页面上的元素

比如：获取页面上的所有文本框

<input type="text" />

2、语法规范

[]

1、[attr]

匹配元素中附带 attr 属性的元素

eg:

[title]:匹配所有附带 title 属性的元素

[class]:匹配所有附带 class 属性的元素

2、elem[attr]

作用：匹配具备 attr 属性的elem元素

eg:

div[title]

3、elem[attr1][attr2]

作用：匹配 同时具有 attr1 属性以及 attr2 属性的elem元素

4、elem[attr=value]

作用：匹配 attr 属性的值等于 value 的 elem 元素

eg:

[class=c1]:匹配class属性的值为c1的元素

div[title='xxx']：匹配 title 属性值为xxx的元素。

【"此处的引号可有可无，也可为双引号"】

p[id="p1"][class=important]:匹配 p 的 id 值为p1且class属性值为important的元素。

5、elem[attr~=value]

作用：匹配 attr 属性值为 值列表，并且 value 是该列表中的一个值的 elem 元素

【值列表】：

<p class="p1 important rf"></p>

eg:

p[class~=important]

匹配 class 属性值为列表值，并且important时该列表中的一个值的p元素

6、elem[attr^=value]

^ ：表示【开始】

作用：匹配 attr 属性值 是以 value 作为开始的 elem 元素

eg：

p[class^=c]:

(匹配到p中class值以"c"字母开始的元素)

7、elem[attr$=value]

$ ：表示【结束】

作用：匹配 attr 属性值是以 value 作为结束的 elem 元素

eg：

[class$=t]

(匹配到页面中class值以"t"字母结束的元素)

8、elem[attr\*=value]

\* ：表示【包含】

作用：匹配 attr 属性值中包含"value"字符的元素

eg：

div[class\*=ant]

（匹配所有class值中包含字母"ant"的元素）

3、伪类选择器

作用：匹配 元素 不同的状态

已学：

1、链接伪类

:link

:visited

2、动态伪类

:hover

:active

:focus

新伪类：

1、目标伪类

1、作用：突出显示活动的HTML锚元素

2、语法

:target

2、元素状态伪类

元素状态：启用的

禁用的

被选中的

使用场合 ：表单元素中

1、语法

1、:enabled

匹配每个已启用的元素

2、:disabled

匹配每个被禁用的元素

3、:checked

匹配每个被选中的元素

3、结构伪类

1、:first-child

作用：匹配属于其父元素中的首个子元素

2、:last-child

作用：匹配属于其父元素中的最后一个子元素

3、:empty

作用：匹配没有子元素的每个元素

eg:

<div></div> -- 即无子元素

<div>

</div> -- 即有子元素

4、:only-child

作用：匹配属于其父元素中的唯一子元素

<div><p>Hello World</p></div>

5、:nth-child(n)

作用：获取属于其父元素中的第n个子元素

eg:

:nth-child(5){} :获取第5个子元素

4、否定伪类

作用：在获取到的元素中进行进一步的筛选或排除。

语法：:not(选择器)

选择器:not(选择器)

eg:

input:not([type="text"])

4、伪元素选择器

与伪类的区别：

伪类：匹配 "元素（不同的状态）"

伪元素：匹配 元素中的"内容的一部分"

1、:first-letter (或) ::first-letter

first:第一个

letter:字符

作用：获取 匹配元素 的首字符。（如：首字下沉）

2、:first-line (或) ::first-line

3、::selection

作用：匹配被用户选取的部分

eg:

::selection{

background:#f00;

color:#0f0;

}

【【【 : 和 :: 的区别】】】：

在 CSS2.0 中，伪元素和伪类选择器统一使用 ：来表示。

从 CSS3 开始，W3C规范所有的伪类选择器一律使用 ：表示，所有的伪元素选择器一律使用 :: 表示。

【为了实现兼容性，尽可能使用 : 表示伪元素】，:: selection 除外。

2、内容生成

1、什么是内容生成

通过CSS生成一部分新的内容，插入到网页元素中。

2、伪元素选择器（需要用到）

1、:before 或 ::before

作用：匹配 指定元素 的内容区域之前

2、:after 或 ::after

作用：匹配 指定元素 的内容区域之后

3、属性：content

作用：生成的内容是什么

属性：content

取值：

1、字符串，使用 " " 引起来的纯文本

2、图像，url(...)

"解决问题"：

1、【【浮动元素的父元素高度】】

解决方案：

1、设置高度

2、overflow:hidden;

overflow:auto;

3、父元素浮动（有局限）

4、增加 子元素 并且 清除浮动

5、

eg:

...

.div:after{

content:"";

display:block;

clear:both;

} html代码:

<div class="div">

<div></div>

<div></div>

<div></div>

</div>

2、【【外边距的溢出】】

解决方案：

1、父元素加上边框

2、通过父元素的 padding-top 代替子元素的 margin-top（有局限）

3、追加一个子元素：<table></table> --table是 行内块元素

4、

eg:

div:before{

content:"";

display:table;

}

3、计数器

1、什么是计数器

通过CSS生成一组有顺序的数字，并且添加到元素的内容区域中。

2、语法

1、声明一个计数器

【counter-reset】 属性

作用：声明/复位 一个计数器

语法：

counter-reset:计数器名称 初始值;

初始值：

默认为0，

可以设置为正数，也可为负数

一次性声明多个计数器：

counter-reset:计数器1 值1 计数器2 值2；

声明计数器的位置：

不能放在使用计数器的元素中

【【【推荐】】】：将计数器声明在 使用元素 的父元素中。

如页面中所有的元素使用相同的计数器的话，可以将其声明在body中。

2、设置计数器增量

counter-increment 属性

作用：设置 某个选择器 每次使用计数器的增量，默认值为1

取值：

正数：递增

负数：递减

语法：

counter-increment:计数器名称 增量;

位置：哪个元素使用计数器，就在哪个元素中声明增量

3、使用计数器

counter()函数

已学的常用函数：url()、linear-gradient()等

【注意】：配合content属性一起使用

语法：content:counter(计数器名称)

content:"~"counter(计数器名称)"~";

3、多列

1、分割列

作用：将文本分成几列

属性：column-count

取值：数字

2、列间隔

作用：每两列间间隔的距离是多大

属性：column-gap

取值：px

3、列规则

作用：设置每两列之间的边框的大小、样式、颜色

属性：column-rule

取值：width、style、color ...

4、【兼容性】

IE10+ , Opera :支持多列属性

Firefox : 需要加上 -moz- 前缀

Chrome , Safari :加 -webkit- 前缀

4、CSS Hack

1、CSS Hack 原理

CSS Hack 是使用 选择器或样式属性的【优先级】来解决兼容性问题的

2、实现方式

1、CSS类内部Hack

在样式属性名称前 或者 属性值之后 增加前后缀，解决兼容性。

2、选择器Hack

在 选择器 名称前 增加 前缀 解决兼容性。

3、HTML头部引用Hack

通过HTML中的 【条件注释】 判断浏览器的版本，解决兼容性问题。

【【条件注释的语法结构】】：

<!--[if 条件 IE 版本]>

待执行的内容（html、css、js...）

<![endif]-->

版本：

6 ：表示 IE6

7 ：表示 IE7

...

省略 ：表示判断是否为 IE 浏览器

【条件】：

1、gt ：表示 选择条件 版本以上的浏览器版本，不包含条件版本

eg:

<!--[if gt IE 6]>

只在IE6（不包含）以上的版本上执行

<![endif]-->

2、gte : 大于等于指定版本

3、lt ： 小于指定版本（不包含）

4、lte : 小于等于指定版本

5、! : 判断是否为指定版本

6月22日

06CSS3CORE-Day02

1、转换

1、什么是转换

转换就是改变元素的形状、尺寸和位置的一种效果

2D:在平面上发生的转换效果 x轴、y轴发生的转换效果

3D:x、y、z轴上发生的转换效果

转换类型：旋转、缩放、移动、倾斜

2、转换属性

属性：transform

取值：

1、none ，默认值，表示当前元素无转换效果

2、transform-function

可以指定 一个 或 多个 转换函数，中间用 空格 隔开

缩放：scale()函数

位移：translate()函数

3、转换的原点

属性：transform-origin

作用：指定元素的转换原点

取值(方式)：

1、默认情况：转换远点存在于整个元素的中心位置处。

2、数值/百分比/关键字

100px/ 50%/ center

给定两个值：x轴、y轴的位置

给定三个值：x轴、y轴、z轴的位置

【注意】：一般在旋转操作时，可能要更改转换的原点。其他转换操作基本不用。

4、2D转换

1、什么是2D转换

在二维空间（平面中），围绕着x轴和y轴所做的转换操作。

2、2D-位移

1、位移

改变元素在x轴和y轴上的位置

2、属性和函数

属性：transform

函数：

translate(x):改变元素在x轴上的位置

translate(x,y):改变元素在x轴和y轴上的位置

取值：数值 / 百分比

可以取负值

x 为正，向右移动

为负，向左移动

y 为正，向下移动

为负，向上移动

单向位移函数：

translateX(x):此处括号外的"X"也可以小写

translateY(y):

3、2D-缩放

1、什么是缩放

改变元素的显示比例（大小）

2、属性和函数

属性：transform

函数：scale(x): x轴与y轴等比例缩放（x与y时相等的）

scale(x,y):

取值：

1、默认值：1

2、放大：大于 1 的值

3、缩小：0 到 1 之间的小数

单向缩放函数：

scaleX(x):只在x轴上的缩放

scaleY(y):只在y轴上的缩放

4、2D-旋转

1、什么是旋转

围绕着一个点（原点），产生角度的变化

2、属性和函数

属性：transform

函数：rotate(ndeg)

n : 为旋转角度

取值为正：按顺时针方向旋转

取值为负：按逆时针方向旋转

【注意】：转换过程中，连同坐标轴一起旋转

eg:

transform:rotate(45deg) translate(200px);

多个转换操作同时进行的时候，方式如上所示，即先旋转，再顺着旋转后的坐标轴位移。

5、2D-倾斜

1、什么是倾斜

让元素按照一定的角度产生倾斜的效果。倾斜会改变元素的形状。

2、属性和函数

属性：transform

函数：

skew(x): x 取值为角度，即ndeg

skew(x,y):

单向倾斜函数：

skewX(x):

skewY(y):

x轴倾斜：让元素向左或向右产生倾斜的变换，

但是 是 y轴 产生了倾斜角度

y轴倾斜：让元素向上或向下产生倾斜的角度变换，实际上是 x 轴按照顺时针产生了

5、3D转换

1、什么是3D

2D与3D之间的差别：空间距离

3D：x轴、y轴、z轴（空间轴）

2、perspective 属性

作用：假定人眼到投射平面的距离

【注意】：

1、该属性要定义在3D变换元素的父元素上。

元素本身不会有3D的效果，而子元素会呈现出3D的变换效果

2、Chrome 和 Safari 需要使用 -webkit-perspective 来替代

eg:

-webkit-perspective:100px;

3、3D-旋转

1、什么是3D旋转

让元素在x轴，y轴以及z轴上分别产生旋转的效果。

即以x轴，y轴或z轴为中心轴所做的旋转操作。

2、x轴旋转

属性：transform

函数：rotateX(xdeg)

3、y轴旋转

函数：rotateY(ydeg) 负值表示往左转

4、z轴旋转

函数：rotateZ(zdeg)

5、rotate3d函数

函数：rotate3d(x,y,z,deg)

x,y,z 表示的不是角度，而是该轴是否要参与到旋转操作中来。

取值：

0 ：该轴不参与旋转

大于0 ：即参与到旋转中来

eg:

rotate3d(1,0,0,45deg) -> rotateX(45deg)

rotate3d(1,1,0,45deg) -> rotateX(45deg) rotateY(45deg)

4、3D-位移

3D位移可以改变元素在z轴上的位置

属性：transform

函数：translateZ(z) // z:100px

translate3d(x,y,z);

5、transform-style 属性

作用：规定如何 在3D空间中 呈现 被嵌套 的元素。

取值：

1、flat : 默认值，子元素将不保留其3D位置（子元素按照2D的方式进行显示）

2、preserve-3d : 子元素将保留其3D位置（子元素会以3D的方式呈现）

eg:

-webkit-transform-style:preserve-3d;

2、过渡

1、什么是过渡

使得CSS属性值在一段时间内平滑的过渡出来。强调的是 一个值 到 另外一个值 变换的过程。

2、如何触发过渡

现阶段：使用 鼠标悬停时 触发过渡操作

点击、鼠标的移入、移出 等，都可以触发该操作

3、过渡相关属性

1、过渡属性

作用：指定 哪个CSS属性值在变化的时候使用过渡的效果来展示。

属性：transition-property

取值：

1、none : 无过渡效果

2、all :全部，任何属性值在发生变化的时候都将使用过渡的效果来展示。

3、property : 具体的 属性 名称

eg：

transition-property:width;

宽度 属性值在发生变化时将使用过渡的效果进行展示

transition-property:transform;

转换 属性值在发生变化时将使用过渡的效果进行展示

【以下属性可以设置过渡的效果】：

1、颜色属性

2、取值为数值的属性：height width margin ...

3、转换属性：transform

eg:

transition-property:transform;

【注意】：rotatex ... 之类的不能用过渡，统一用transform

eg:

transition-property:rotatex; 写法错误

4、渐变属性

5、visibility

6、阴影属性

2、过渡时间

1、什么是过渡时间

规定完成过渡效果所需要花费的时间。

以 s / ms 作为单位。 1 s = 1000 ms

2、属性/语法

transition-duration

取值：以 s 或 ms 为单位的数值

【注意】：

该属性值默认为0，意味着不会有过渡效果。如果先显示出过渡效果，该属性必须设置。

eg:

transition-duration:1000ms;

3、过渡的速度时间曲线函数

1、作用

过渡过程中的速度是什么样的

2、属性

transition-timing-function

取值：

ease : 默认值，慢速开始，快速变快，再慢速结束

【linear : 匀速】

ease-in : 慢速开始，加速结束效果

ease-out : 快速开始，慢速结束

ease-in-out ：慢速开始和结束，中间先加后减

4、过渡延迟

1、作用

用户激发过渡效果后，等待多长时间后再显示过渡效果

2、属性

transiton-delay

取值：

以 s / ms 为单位的数值，默认值为 0

5、过渡综合属性

属性：transition

取值：

porperty duration timing-function delay;

eg：

transition:background 1s linear,width 2s ease-in;

3、动画

1、什么是动画

动画（animation），通过 关键帧 控制 动画 的每一种不同的状态

动画能使得元素从一种样式逐渐变化为另一种样式

兼容性：

Chrom & Safari ： -webkit-

Firefox : -moz-

Opera : -o-

IE : -ms-

2、动画的使用步骤

1、声明动画

1、为动画指定名称

2、指定 关键帧 的状态

语法：

@keyframes 动画的名称{

关键帧的声明：

1、时间点 （"时间点是由 %表示的"）

2、在该时间点上的状态

时间点{

状态：若干CSS属性

}

eg:

0% | from{

动画开始时的状态

width:400px;

......;

}

......

50%{

动画执行到一半时的状态

width:200px;

......;

}

......

100% | to{

动画结束时的状态

}

}

【浏览器兼容性】：

Chrome $ Safari : @-webkit-keyframe 名称{}

Firefox : @-moz-keyframe 名称{}

2、调用动画

属性：

1、animation-name

为元素指定调用动画的名称，类似于过渡中的 property

2、animation-duration

动画执行时长

3、animation-timing-function

动画执行时的速度时间曲线函数

取值：

ease

linear

ease-in

ease-out

ease-in-out

4、animation-delay

动画执行的延迟时间

5、animation-iteration-count

作用：规定动画的播放次数

取值：

1、具体的数值

2、infinite

无限次播放

6、animation-diraction

作用：动画播放方向

取值：

1、normal

正向播放，从0%到1000%

2、reverse

逆向播放，从100%到0%

3、alternate

奇数次数 正向播放

偶数次数 逆向播放

7、animation 综合属性【不包括第8、9个属性】

animation:name duration

timing-function delay;

iteration-count direction;

8、animation-fill-mode

作用：指定动画播放前或播放后的填充状态

取值：

1、none

默认值，无填充状态

2、forwards

动画完成后，将保持在最后一帧的状态上

3、backwards

动画显示之前，将保持在第一帧的状态上

【注意】：

只有在有延迟的时候，才会显示在第一帧的状态上

4、both

动画开始之前和结束之后，都应用在对应的状态上

9、animation-play-state

作用：控制动画播放还是暂停

取值：

1、paused 暂停

2、running 播放

## 6月23日07JAVASCRIPTBASIC-Day01

1、JavaScript概述

1、什么是JavaScript

简称JS

一种运行于 JS解释器/引擎 中的解释型脚本语言

1、JS解释器/引擎

2、解释型脚本语言

1、编译型

C/C++

先编译再执行

2、解释型

无需编译，直接执行，遇到错误则停止

JS解释器：

1、独立安装的JS解释器

NodeJS

2、嵌入在浏览器内核中的JS解释器

2、JS发展史

1、1992年 Nombas公司 --> CMM(C--) --> ScriptEase

2、1995年 Netscape网景 -->

3、1996年 ...

4、1997年 ...

完整的JS由三部分组成

1、JS核心-ECMAScript（ES）

2、文档对象模型（DOM ：Document Object Model）

3、浏览器对象模型（BOM : Browser Object Model）

3、JS特点

1、任何文本编辑工具都可以编写

2、无需编译，由JS解释器负责运行

3、弱类型语言

由数据来规范语言

强类型语言：

由数据类型来规范数据

4、【面向对象】

4、JS用途

1、客户端JS（前端）

1、数据计算

2、表单验证

3、异步提交数据（Ajax）

4、事件的触发和处理

5、特效

2、服务器端JS（后端）

1、分布式运算

2、实时服务器

3、窗口应用

4、网络应用

2、JS语法

1、使用JS

1、浏览器内核

作用：负责渲染页面。由两部分组成

1、内容排版引擎：解析HTML与CSS

2、脚本执行引擎：解析JS

"浏览器 内核 排版引擎 脚本引擎"

IE Trident - Chakra

Firefox Gecko - Monkey~

Safari Webkit Webcore Nitro

Chrome Webkit Webcore V8

Opera Presto - Carakan

Opera(2013) Webkit Webcore V8

2、环境搭建-运行

1、独立安装的解释器 - NodeJS

1、开始菜单中 --> cmd

2、嵌入在浏览器中的引擎

1、使用浏览器控制台输入代码并执行

2、将脚本嵌入在HTML页面中【重要】

1、直接在HTML元素事件中执行JS脚本

onclick事件：单击元素时所做的操作。

eg:

<button id="btn" onclick="console.log("JS脚本")">

测试

</button>

【

2、在HTML页面中，通过

<script></script>

编写JS脚本

3、将JS脚本 编写在外部脚本文件中（js文件）

代码：

document.write("output...");

】

代码： console.log("待输出内容");

作用：在控制台上输出一句话

......

2、调试JS

3、语法规范

1、基本规范

1、语句：组成JS的最基本单位

console.log();

document.write();

var a = 10;

var r = a + b / c;

1、由 表达式、关键字、运算符

关键字：由JS定义，具备特殊功能的字符，称为关键字

2、严格区分大小写

3、语句使用 ; 表示结尾

2、注释：不会被执行

单行注释： //

多行注释： /\*\*/

3、变量和常量（重点）

1、什么是变量

内存：临时性保存。

保存程序正在运行过程中所需要用到的数据。

硬盘：永久性保存数据。

内存空间：内存中，保存数据的一个空间

变量：内存中的一个存储空间，一个可以变化的数据。

变量名：一个变量（内存空间）的别名

值：变量中保存的数据

2、变量的声明

1、什么是变量的声明

如何到计算机内存中申请一段空间用来保存数据

2、语法

var 变量名;

var 变量名1,变量名2,...;

eg:

1、声明一个空间，用于保存用户的名称

var userName;

2、声明几个空间，用于保存用户的年龄，性别，工资

var age , gender , salary ;

3、赋值

语法：

变量名=值;

1、var age;

age = 18;

var age = 18;

注意：

1、值为 数字

var age = 18;

var score = 98.5;

2、值为 非数字，即字符串时

var userName = 'hello';

4、打印输出变量

console.log(变量名);

document.wirte(变量名);

eg:

var age = 18;

console.log(age); //18

document.write(age); //18

5、变量名命名规范

1、不允许使用 JS关键字、保留关键字 做变量名

2、可以包含 字母、数字、下划线\_ 、$ 符号

3、不能以 数字 开头

4、最好见名知意

5、命名方式

1、匈牙利命名法

txt : 文本框、密码框、...

rdo : 单选按钮

chk : 复选框

...

在获取页面元素时使用

2、下划线命名法

var \_userName;

3、驼峰命名法

变量名称由多个单词组成，第一个单词所有字母全部小写，从第二个单词开始，每个单词的首字符变大写。

/\*var userName;\*/

3、变量的使用

1、变量未经初始化就使用

只声明，未赋值的变量，结果为 undefined

2、未经声明的变量

报错

3、声明变量不使用 var 关键字

gender = "男"; //可以

如果声明变量时不使用var关键字而直接赋值的话，声明的是一个全局变量，容易引发"全局污染"的效果。

声明变量时，尽量使用 var 关键字。

4、【变量的存取操作】

1、存 - 为变量赋值

变量 出现在 = 的左边，一定是赋值

2、取 - 获取变量的值

1、放在函数中使用

document.write(变量);

console.log(变量);

2、变量出现在 = 的右边，一定是取值

var userName = "张三"; //赋值

var newName = userName;

newName : 取值操作

newName : 赋值操作

有 = 的时候，一定是【先计算 = 右边的表达式或变量的值】，

再将其赋值给 = 左边的变量。而且 = 的左边必须是【变量】。

4、运算符

1、赋值运算符

= ：将等号右边的内容赋值给左边的常量

由赋值运算符组成的表达式称为 赋值表达式

赋值表达式的值 为 右侧表达式计算出来的结果

2、算数运算符

+ , - , \* , / ,%

1、%

取余，俗称：模

eg：

2 % 5 ：结果为 2

【场合】：

1、判断某数字的奇偶性

55 % 2 结果为 1，所以为奇数

2、判断某数字是不是 n 的倍数

判断 188 是不是 3 的倍数

判断 188 和 3 取余，看结果是否为 0 。

3、获取某数字的最后几位

获取 1234 的最后一位

eg：

1234 % 10 结果 4

1234 % 100 结果 34

2、+

作用：

1、做 加法 操作

要求：+ 两边的操作数必须是数字

2、做 连接 操作

如果 + 两边的操作数，有一个是字符串类型的，那么起连接的效果

3、优先级

先算 \* / % , 再算 + - 。平级时：从左向右算

可以通过 () 提升优先级

由算数运算符组成的表达式称之为算数表达式。

5、常量

1、什么是常量

一经声明就不允许被修改的数据，称之为常量

2、目的

为了保证数据的准确性

3、语法

const 常量名称 = 值;

注意：常量名一般采用全大写方式表示

eg:

圆周率符号 用 PI

4、数据类型

问题：想在内存中保存一个数字，开辟多大空间比较合适。

1、什么是数据类型

保存的数据类型是什么

2、数据类型的分类

1、原始类型（基本类型）

1、number 类型

数字类型

可以表示32位(4字节)的整数，或64位(8字节)的浮点数（小数）

2、string 类型

字符串类型

注意：使用时，必须用 "" 或 '' 引起来

eg:

var num = 18; //number类型

var num = '18'; //string类型

转义字符

\n :换行

\r :回车

\t :制表符

\\ : \

\' : '

\" : "

3、

2、引用类型

6月24日 07JAVASCRIPTBASIC-Day02

js-basic Day-01 前情回顾

1、回顾

1、

2、调试

3、语法规范

1、JS代码组成

1、语句

console.log();

document.write();

var userName = "";

......

表达式、关键字、运算符 ...... 区分大小写。都是 以 ；结尾

4、变量

1、什么是变量

一个可以变化的数据

变量名：数据的名称、内存空间的别名。

值：保存在变量中的数据，可变。

2、声明变量

3、为变量赋值

使用 = 为变量赋值

4、变量名命名规范

1、不能使用JS关键字和保留关键字

2、由字母、数字、下划线\_、$ 组成

3、不能以数字开头

4、最好见名知意

5、可以采用"匈牙利"、"下划线"、"驼峰"命名法

5、变量的使用

1、使用未初始化的变量

2、使用未声明过的变量

3、声明变量不带 var

4、变量的存取操作

1、存 - 为变量赋值

特点：变量出现在赋值符号 = 的左边

2、取 - 使用变量的值

1、配合函数使用

console.log(userName);

document.write(userAge);

2、变量出现在赋值符号的右边

5、常量

1、什么是常量

一旦声明之后不允许被修改的数据就是常量

2、目的

为保护数据的正确性

3、声明常量

const 常量名 = 值;

6、数据类型

1、什么是数据类型

保存在内存中的数据的类型

2、JS数据类型

1、原始类型（基本类型）

1、number

数字类型

2、string

字符串

注意：使用时，必须使用 "" 或 '' 引起来

转义字符：

\n : 换行

\r : 回车

\t : 制表符

\\ : \

\" : "

\' : '

3、boolean

布尔类型

只负责描述 真 或 假 的结果

取值：

true : 真 ，1

false : 假 ，0

boolean 类型是允许参与到 number 类型的运算中。

4、undefined

声明未赋值

访问对象的属性不存在

5、null

2、引用类型

js-basic Day-02

2、数据类型转换

1、弱类型

变量的 数据类型 取决于 值

2、typeof 函数（运算符）

作用：查看当前变量的数据类型

语法：

1、typeof(变量名)

2、typeof 变量名

3、隐式转换

1、问题

var userName = "Mary";

var userAge = 18;

var result = userName + userAge;

2、什么是隐式转换

不同类型的数据在做运算的时候，可以进行默认的数据类型的转换。

3、隐式转换

NaN ：Not a number (不是一个数字)

函数：isNaN(变量) ：是 不是一个数字

false : 是数字

true : 不是数字

1、string 类型转换

var a = "18";

var a = "18";

var r1 = a + b; //1818 字符串

var r2 = a - b; //0 数字

var r3 = a \* b; //324 数字

var r4 = a / b; //1 数字

var r5 = a % b; //0 数字

【值 为 数字 的字符串，在做 - \* / % 运算时，

会自动转换成 number 类型进行运算，+ 仍然是 连接 操作。】

2、在JS中，所有用户手动输入的数据，默认都是 string 类型。

1、window.prompt("提示的文本"[,默认值]);

2、prompt("提示的文本"[,默认值]);

4、显示转换

由 转换函数 完成 显式转换

1、toString()

转换成 字符串，将任意类型的数据转换成 string 类型

var num = 15;

num.toString();

2、parseInt(参数)

将 参数 转换成 整数

碰到第一个非数字字符，则停止转换

3、parseFloat(参数)

将 参数 转换成 小数

3、函数

1、使用过的函数

console.log();

document.write();

window.prompt();

parseInt();

parseFloat();

2、为什么要使用函数

1、放在<script>中的js脚本，在网页加载时就执行 -- 执行时机没法选择。

2、代码冗余 -- 将代码写在函数事件中。

3、可重用性不高

3、什么是函数

函数（Function）:也被称之为 方法（Method），或者过程（Procedure）。

函数是一段被预定义好，并且可以被反复使用的代码块。

预定义：先编写完成，但不会在被网页加载时就执行的代码。

反复使用：可以被无限次的使用。

代码块：将多行代码放在一起去管理。

函数其实就是一个完整的且功能独立的功能体。

4、函数的声明

使用关键字 function 声明一个函数

语法：

function 函数名称(){

代码块;

可执行的操作语句;

}

5、函数的调用

JS脚本中，通过 函数名() 进行函数的调用

练习：

声明一个函数，在函数体内，要求用户输入两个数字，计算两个数字的加减乘除结果，并且打印出以上结果。

在页面上定义一个按钮，实现对以上函数的调用。

//见06.html

6、定义参数

参数：函数在运行过程中，需要用到的数据进行辅助操作，数据就是参数。

1、语法：

function 函数名(参数列表){

//函数体

}

function 函数名(参数名1,参数名2){

}

function zha(fruit){

console.log(fruit);

document.write(fruit);

}

形参：定义函数的时候声明的参数。

调用带参函数：

函数名(参数值列表);

【注意】：调用带参数的函数时，参数的顺序、数量 要与声明时保持一致

实参：调用函数的时候传递的值。

4、分支结构

1、流程控制

程序 = 数据结构 + 算法

任何程序都由以下三种结构组成：

1、顺序结构

2、选择结构（分支结构）

根据“条件”，选择某一段代码去执行。

3、循环结构

根据“条件”，选择某一段代码反复执行。

2、分支结构

1、什么是分支结构

根据不同的条件，选择不同的代码执行

if 结构：

当条件满足时，运行指定的语句，

当条件不满足时，则不运行这些语句

if...else 结构：

当条件满足时，运行指定的一组语句，

当条件不满足时，运行另一组语句

2、if 结构

语法：

if(条件){

语句块;

}

条件：boolean 类型的结果或表达式

> , < , >= , <= , ==

注意：

1、if (条件){}

条件：必须是boolean类型

【如果不是boolean类型的话，会产生隐式转换】

以下情况会默认转换成 假：

if(0){}

if(0.0){}

if(""){}

if(null){}

if(undefined){}

if(NaN){}

eg ：

if(1){} //真

if('今天下雨了吗？'){} //真

2、if 后的 {} 可以省略

如果省略，那么 if 只控制他下面的第一条语句

3、if-else 结构

语法：...

...

4、多重 if 结构

1、问题

编写一个函数，让用户输入一个...

2、语法

if(条件1){

语句块1;

}else if(条件2){

语句块2;

}else if(条件3){

语句块3;

}...else{

语句块n;

}

特点：

从上到下判断条件，满足哪个条件则执行对应的语句块，

执行完成后，就跳出该结构，即便满足其他条件，也不会判断。

练习：

计算 可莱托 指数

公式：体重(kg)/(身高(m)\*身高(m))

20-25:正常

20以下：偏瘦

25以上：偏胖

1、回顾

1、数据类型转换

1、什么是数据类型转换

2、数据类型转换分类

1、隐式转换

自动完成

2、显使转换

转换函数

1、toString()

将任意类型的数据转换成string类型

2、parseInt()

将指定的数据转换成【整数】。如果碰到第一个非数字字符则停止转换。

eg:

parseInt(53.5) -> 53

parseInt("73") -> 73

parseInt("12Hello") -> 12

parseInt("H23") -> NaN

3、pareFloat()

将指定的数据转换成小数

2、函数

1、什么是函数

函数(Function)：又称为方法(Method)、过程(Procedure)

2、定义函数

语法：

function 函数名(){

函数体;

}

3、调用函数

函数调用的几种方式：

1、函数名();

2、<button onclick="函数名()"></button>

3、<script>

函数名();

</script>

4、function 函数1(){

函数2();

}

5、function 函数3(){

函数3();

}

4、带参函数

1、声明

语法：

function 函数名(参数列表){

函数体;

}

2、调用

函数名(参数值列表);

3、分支（选择）结构

1、程序的流程

1、顺序结构

自顶向下，逐条语句执行

2、分支结构

根据指定的条件，选择某段代码去执行

3、循环结构

2、分支结构

......

练习 ：

编写一个“个人所得税计算器”函数。该函数 接受一个参数值，来表示税前工资，在函数内将工资所对应的税额计算出来，并打印输出。

个税计算梯度如下：

3500以下：免征

3500-5000的部分：缴纳3%个税

5000-9000的部分：缴纳10%个税

9000以上部分：缴纳20%个税

2、循环结构

1、问题

1、在控制台上打印输出一遍 Hello World

2、在控制台上打印输出十遍 Hello World

3、在控制台上打印输出100遍 Hello World

4、将 Hello 改成 您好

5、在输出的 您好 前加上 序号

2、循环

1、什么是循环

循环：重复执行相同或相似的事情（代码）

3、循环特点

1、循环条件

2、循环操作 又称为循环体

3、while循环

1、语法

while(条件){

循环操作;

}

流程：...

eg:

while(true){

} //死循环

2、注意：

一定要在循环体内更新循环条件，否则会进入死循环

i = i+1;

i += n;

i ++;

i --;

3、break

1、作用

用于循环中，终止循环的执行

4、for循环

1、while循环

eg:

打印1-100之间所有的数字

var i=1;

while(i<=100){

console.log(i);

i++;

}

2、for循环

表达式1 表达式2 表达式3

for(声明循环变量;循环条件;更新循环变量){

循环体;

}

3、使用场合

1、for

已经明确了循环次数时，优先使用 for 循环。

2、while

不确定循环次数时，相比之下可以优先使用 while 循环。

4、for 中三个表达式的特殊用法

1、第一个表达式为空时

var i=1;

for(;i<=100;i++){

console.log(i);

}

注意：可以省略第一个表达式，但是必须要在循环之外将第一个表达式内容补齐，否则就是语法错误。

2、第二个表达式为空时

for(var i=1;;i++){

console.log(i);

}

省略第二个表达式时，默认其为真。会导致死循环。

如果省略第二个表达式，一定要在循环体内将条件补足。否则就是死循环。

3、第三个表达式为空时

for(var i=1;i<=100;){

console.log(i);

}

省略第三个表达式时，结果为死循环。如果要省略，需要在循环体内补足第三个表达式的值 或 给定退出条件

4、省略所有表达式

for(;;){

}

省略所有表达式，导致死循环。补足所有的内容即可。

5、表达式1、表达式3的位置处，允许接受多个表达式。中间用，隔开即可。

3、数组

1、问题

2、什么是数组

多个数据组成的集合

将多个数据保存在一个变量名中

下标：数组中每个元素的序号。下标：0 ~ (数组个数-1)

3、定义和初始化数组

语法

var 数组名=[]; //声明一个空数组

var 数组名=[值1、值2、值3]; //声明一个包含三个元素的数组

var 数组名=new Array(); //声明一个空数组

var 数组名=new Array(值1，值2，值3)； //声明一个包含三个元素的数组

4、访问数组

1、访问数组元素的方式

index:下标/索引,...

数组名称[下标];

2、存 - 设置数组元素的值

数组名称[下标]=值;

注意：如果下标所对应的元素已经存在，就会实现覆盖

3、取值

数组名称[下标];

注意：访问数组中不存在的元素，结果为 undefined

4、获取数组长度

数组长度：即数组中元素个数

语法：

数组名称.length