部分

# 前端

[李然----g-liran@tedu.cn](mailto:李然----g-liran@tedu.cn)

## 一、学习体系

## 1.HTML

用来勾勒出网页的结构和内容-骨架

html保存在服务器，运行在浏览器

html--超文本标记语言

### **语法**

1. **标签**

<标签></标签>

ex:<div></div>

1. **属性**

<标签 属性1=’值1’ 属性2=’值2’ …></标签>

ex:<span id=’s1’ style=’..’></span>

1. **标签与标签之间的嵌套关系**

<div>

<span id=’s1’></span>

</div>

1. **标签分类**
   1. 闭合标签/双标签

<body>内容<body1>

* 1. 非闭合标签/单标签

<标签>或

<标签/>

<meta charset=’utf-8’>

\*\*单标签没有内容这个概念

总结：学习html就是学习固定标签，固定属性以及标签之间的嵌套关系

### **HTML用法**

html经历了5个版本

html4之前版本很宽泛，宽松

html4非常严谨

html5是最新也是最流行的版本，比html4宽泛

总结：doctype用于约束HTML文档内部的标签，属性和嵌套关系

不写doctype，同一篇代码会在不同浏览器中有巨大的显示差异



### **HTML常见标签**

1. **文本标签--跟论文格式很相似**

1.1标题 <hn></hn> n = 1~6

ex:<h1></h1>

<h2></h2>

…

<h6></h6>

总结：h1最大，h6最小

字体较粗

垂直方向，上下都有空间

1.2段落 <p></p>

垂直方向，上下都有空间

1. **课堂案例**

使用hn标签和p标签，写一份简易的简历

属性align=‘center’可以使用在标题标签中

1. **列表标签--使文本显得有条理**

3.1无序列表

u—unordered

l—list

i--item

<ul>

<li></li>

<li></li>

…

</ul>

3.2有序列表

<ol>

<li></li>

<li></li>

…

</ol>

1. **分区**

<div></div>

div看不见，功能是为了逻辑分区

<div id='header'></div>

<div id='nav'></div>

…

H5添加了很多和div一样，但是更容易读容易理解的标签

<header></header>

<nav></nav>

<footer></footer>

\*块级分区 div在这个区域独占一行

\*行级（行内）分区

<span></span>

如果只是一行文字内部的区域进行分区设置属性或者修改样式，此时使用行内分区

行内分区不会换行，会和其他文字混合在同一行上

1. **元素（标签）的显示方式**
   1. 行级元素（行内）

可以和其他行级元素位于同一行显示

span

* 1. 块级元素

自己独占一行

hn p li ul ol div

1. **其他行内元素**

i/em 斜体

b/strong 粗体

u 下划线

del/s 删除线

1. **实体引用，实体替换符号**

http://www.w3school.com.cn/html/html\_entities.asp

行内元素的空格折叠现象

写代码的时候，不管连写多少个空格和回车，都显示成一个空格

<br>或<br/>🡪回车

&nbsp；🡪空格

<>🡪&lt（<） &gt（>） &amp(&) &#39（'） &apos（”）

1. **图片标签**

单标签，最重要的属性是src

<img src=’图片路径’>

支持图片格式：png jpg gif

图片路径的写法

主机、路径、文件名 绝对路径

在一个页面中 显示其他网站中的图片，这种行为成为图片盗链，通常使用绝对路径，称提供网页图片的网站称为图床，节省了自己服务器的存储空间，坏处是当图床网站将图片路径改变，自己的网站就显示不出来了

<http://cdn.tmooc.cn/tmooc-web/imgad/2018/6/29//1A69E1CFF81D4D4AA90AAAC08720E75F.png>

相对路径🡪参照物🡪当前.html文件

img的其他属性：为行级元素，自带了width和height属性，通常只写一个尺寸属性，另外一个自适应

title：鼠标悬停在图片上，显示的文本

alt：找不到图片的时候显示的文本

1. **超链接a标签**

用于跳转页面

链接到网页、图片、下载

<a href="01\_img.html">跳转到本机Troye链接</a>

<a href="img/troye1.jpg">图片链接</a>

<a href="http://www.tmooc.c n/">跳转到网页</a>

a标签的属性

target：有两个取值：\_self,\_blank默认为\_self

\_self：链接页面在当前选项卡打开

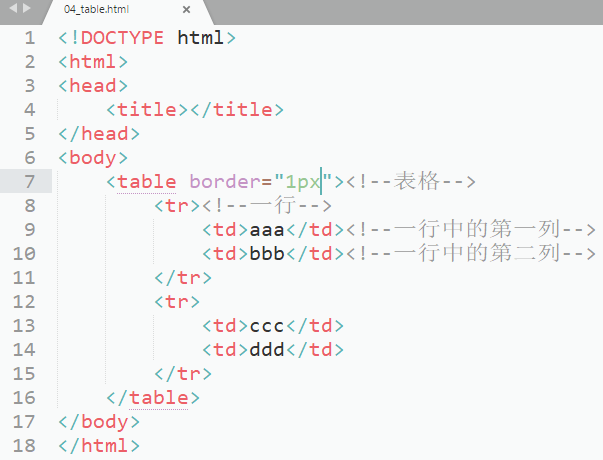
\_blank：链接页面在新的选项卡打开

\*\*两个页面之间的跳转，是超链接

一个页面内部的位置跳转，是锚点

H5锚点的用法：  
在需要跳转的位置标签内添加id属性，比如id=‘abc’，在a标签内的href属性中添加值href=‘abc’，这样点击a标签，页面就会自动跳转到锚点的位置，另：href=‘#’会让页面直接跳转到顶部

1. **网页中的表格标签**





table的常用属性

width：宽度

height：高度

border：边框宽度

algin：表格对齐页面

cellpadding：边框与内容之间的距离

cellspacing：边框之间的距离

colspan：跨列，一个单元格占据多列

<td colspan=’2’></dt>从左往右跨两列

rowspan：跨行，一行单元格占据多行

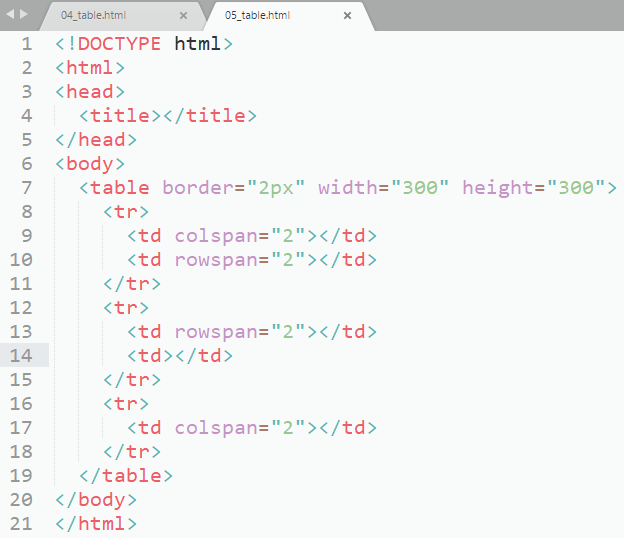
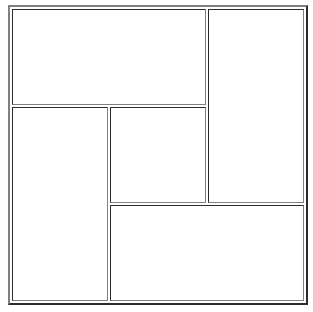
<td rowspan=’2’></dt>从上往下占2行，由于占据了下面单元格的位置，下面只用写n-1列，都减少一个td

1. 表分组

<thead></thead>

<tbody></tbody>

<tfoot></tfoot>

🡪

1. **表单标签**

用于在页面中显示一个可以收集用户填写的数据的控件

总结：想通过网页吧用户填入的信息发送给服务器处理

* 1. 所有的标签必须在form表单中放置
  2. form表单中的标签必须有name属性才可以传
  3. 使用submit将数据提交给服务器

form表单的主要属性

action：表单提交数据的url

method：表单提交数据的方法（get，post）

enctype：表单数据的编码格式

**1.文本框和密码框**

<input type=’text’ name=’…’> 文本框

<input type=’password’ name=’…’> 密码框

value属性：值

placeholder：输入文本提示

maxlength：限制输入的最大字符数

redonly：只读

lable标签：与span类似，但只用在表单中

**2.radio：单选框**

<input type="radio" name="gender">

<input type="radio" name="gender">

\*\*name属性决定一组的单选

\*\*单选框只有有value属性才能传值

\*\*checked=”checked”属性表示默认选中的单选框

<input type="radio" name="gender" value="男" checked="checked"><label>男</label>



**3.checkbox：多选框**

\*\*name属性，决定了一组多选

\*\*有value属性才可以提交值

\*\*checked=”checked”默认选中



**4.submit 按钮：提交**

<input type="submit" value="提交">

**5.button按钮**

<input type="button" value="click" onclick="alert('fdadg')">

**6.重置按钮：reset，将表单内的内容清空重置**

<input type="reset" name="重置">

**7.隐藏域**

<input type="hidden" name="id" value="123456789">

在form表单中存在，但是对用户隐藏，包含着不给用户看的信息，可以提交给服务器

**8.文件选择器-上传文件用**

<input type="file" name="up" value="选择文件">

属性multiple：多重的，可以选择多个文件，不写则只能选择一个

**9.文本域**

<textarea>大量文本</textarea>

rows cols描述的是文本区域内可见的宽度和高度，原意为行数和每一行显示的字符数，但由于系统编码不同，浏览器解析也不同，导致显示字符数差异很大

**10.下拉列表**

<select>

<option></option>

<option></option>

……

</select>

总结：

1.当option没有value属性时，select标签的value为选中的option的内容

2.当option有value属性时，select标签的value为选中的option的value值

**表单标签总结：**

1.只有含有name属性的表单标签才可以提交数据

2.radio和checkbox必须设置value属性才可以提交数据

3.text的value是直接设置值，placeholder是设置提示

4.radio和check默认选中为checked

5.select默认选中为selected

## 2.CSS

用来美化网页-皮肤-化妆

1. 内联样式

<span style="color:#f00;font-size:30px">内联样式的使用</span

在标签的style属性中写样式

特点

内联样式使用不多

1.不能重用

2.内联样式优先级最高

项目代码完全不使用内联

只有在学习和调试过程中

1. 内部样式

在head标签中，写style标签

在style标签内部写样式

选择器{

属性：值;

}

样式代码可以重用,但只能在本html中重用

在项目使用不多

1. 外部样式

创建一个.css文件

在html文件中，head标签中引用这个css文件

<link href="03.css" rel="stylesheet">

1. 层叠样式表的规则特性

1.层叠效果

多个样式作用在同一个标签上

如果样式属性不重复

那么这些样式都会产生效果

2.继承

父级元素的样式会继承给子元素

子元素可以重新写样式，

不使用父级元素的样式

3.优先级

优先级最低的 浏览器默认样式

优先级最高的 内联样式

内部样式和外部样式--就近优先

以被影响标签为基准，哪个样式离这个标签近

就使用哪个样式的效果

1. 选择器

选择器的作用，为了选出当前页面中符合要求的一个或者多个标签

样式属性的作用是给选出来的标签设置样式

选择器{样式属性：值;}

* 1. 通用选择器

\*{}：尽量少用，写通用样式的时候使用

* 1. 标签选择器

选出当前页面中标签名称符合要求的标签

ex：div{} span{} P{}

使用范围；设置某个标签的通用样式

* 1. id选择器

唯一不重复

#p2{

border: 2px solid #b00;

}

id选择器 #id值

同一个html中，id值是惟一的，不可重复

同一个html中使用一个id选择器，只能找到一个标签

* 1. 类选择器（class选择器）

如果编写html标签时，需要让某些标签应用一类样式，就使用类选择器，标签的属性class为类名

<div class="d1"></div>

.d1{

width: 200px;

height:100px;

background-color: #006699;

border: 1px solid #f00;

}

多类选择器：当一个标签想匹配多个样式的时候，使用多类选择器

<div class="w200 h50 bg-c"></div>

* 1. 分组选择器

把一系列选择器当成一组，设置共有样式

div,p,h1,ul,li{

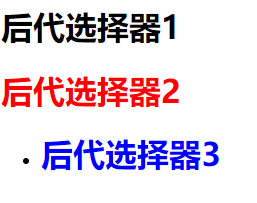
background-color: #f00;

}

逗号分隔，可以使用标签选择器，id选择器，class选择器混合编写，一般用于设置公共样式

一般项目中分组选择器写在最开始的部分

* 1. 后代选择器（派生选择器）



div内部只要存在h1，就可以定位到h1

* 1. 子代选择器

div>h1{

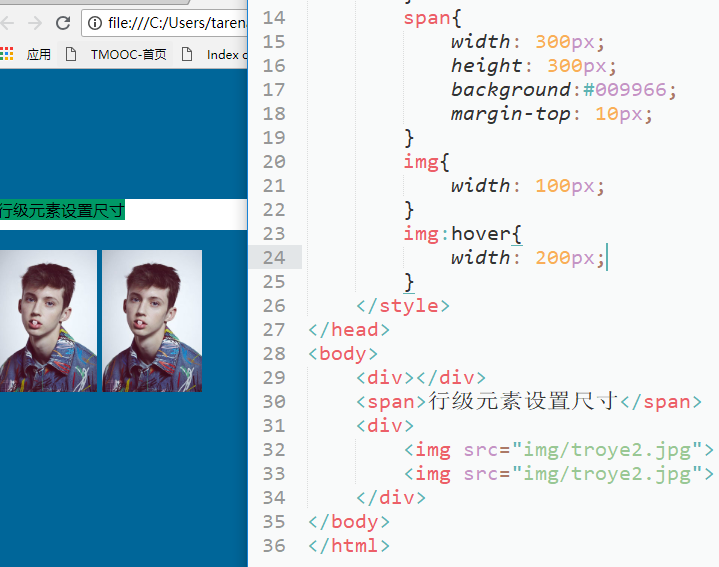
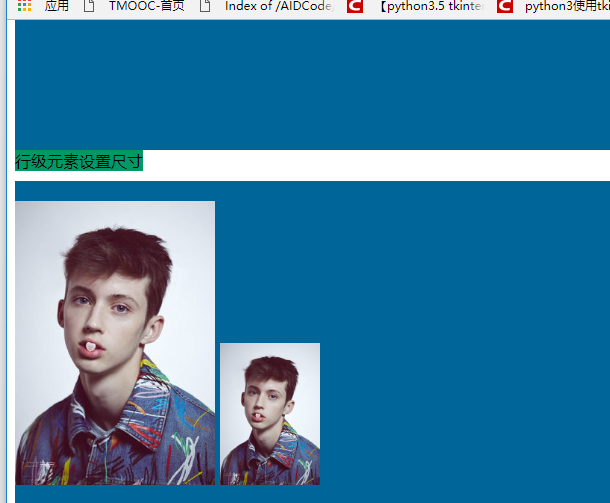
background-color: pink;

}

h1必须是div的儿子才可以匹配

* 1. 伪类选择器

标签名：hover{}：设置鼠标悬停时的样式

1. css常用样式属性
   1. 布局 相关属性

1.width：宽

2.height：高

单位：px cm 100%

3.margin：设置外边距

margin：10px，同时设置四方外边距

margin；10px 20px，上下为10，左右为20

margin；0px auto，块级元素水平居中

margin：10px 20px 30px 40px；上、右、下、左

margin-top：10px；上边距

margin-bottom：10px；下边距

margin-left：10px；左边距

margin-right：10px；右边距

4.padding：10px；设置4个方向的内边距

padding：10px 20px；上下为10，左右为20

padding：10px 20px 30px 40px；上、右、下、左

padding-top：10px；上内边距

padding-bottom：10px；下内边距

padding-left：10px；左内边距

padding-right：10px；右内边距-

改变内边距，视觉效果，元素大小改变

\*\*\*块级元素和行级元素的区别\*\*\*

块级元素可以设置宽高

行级元素，除了自带宽高属性的标签，其他元素宽高设置无效

margin-top，margin-bottom对行级标签无效

3.盒子模型（框结构）

css渲染页面时，使用盒子模型

对元素之间的留白，元素背景填充范围，元素大小的控制规则，就是盒子模

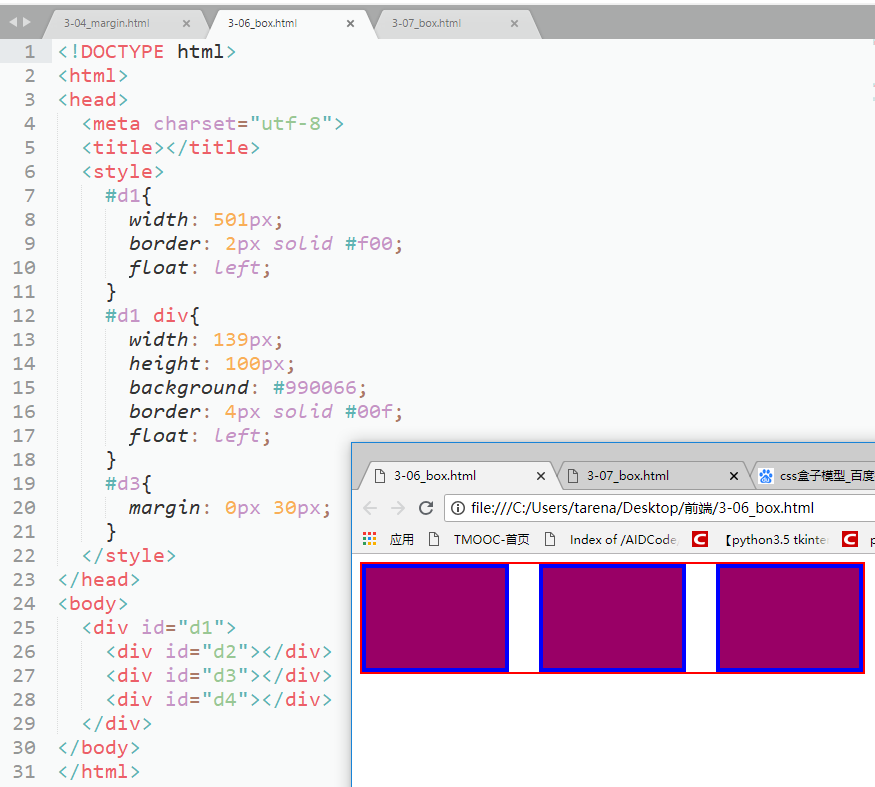
型

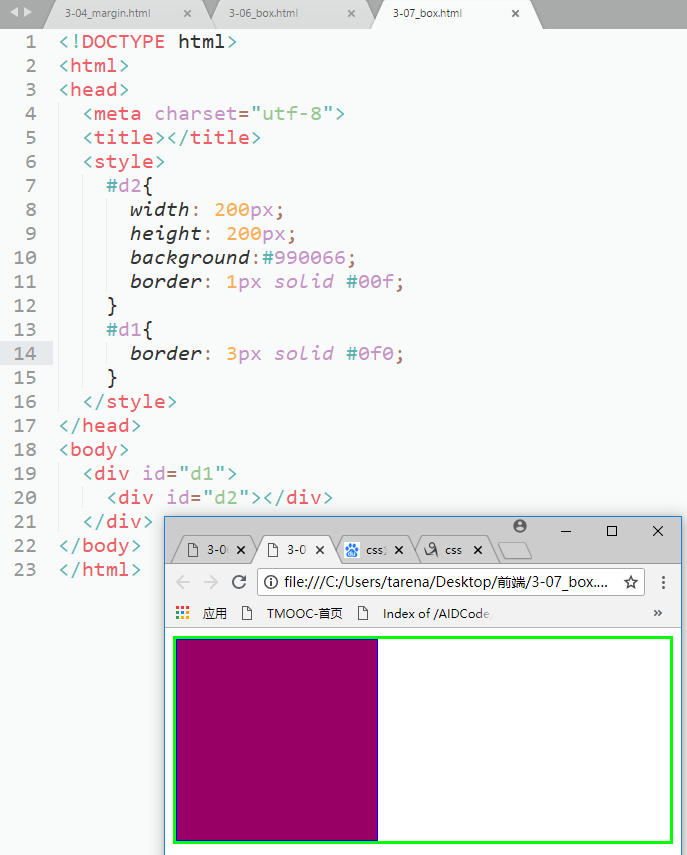
主要关注宽度

左外边距+左边框+左内边距+内容区域+右内边距+右边框+右外边距

总结：div默认占据的宽度是父容器的内容区域100%

div默认的高，是内部内容的高，如果内部没有内容，或没有高，则高度为0





* 1. 背景属性

1.background-color：合法的颜色值

background：合法的颜色值

合法的颜色值：

1.颜色的单词：red、pink…

2.#ffffff：6位16进制字符串

3.#aabbcc—>#abc

4.rgb（255,255,255）；

5.rgba（255,255,255,0~1）a为不透明度

2.背景图片

background-image：url（”图片路径”）

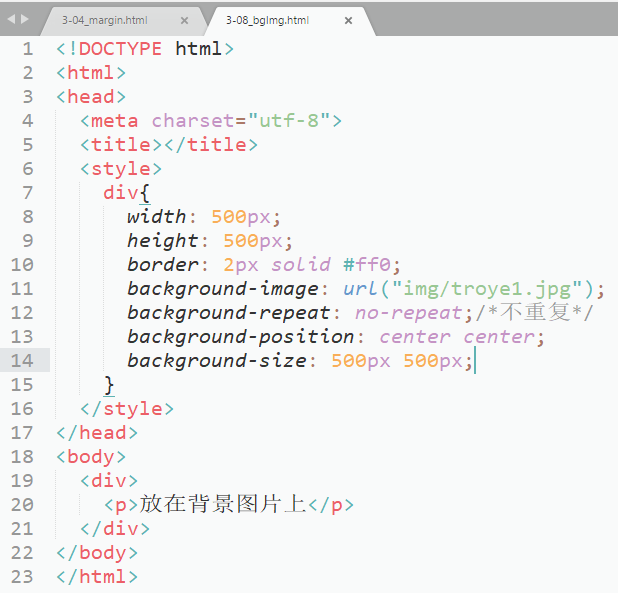
background-repeat：平铺方式

no-repeat repeat repeat-x repeat-y

不重复 默认，重复 水平平铺 垂直平铺

background-position: center center;（left，right，top，bottom可选，也可直接写数值，如100px）

background-size: 500px 500px;设置背景图片的大小



### 课堂练习-学子商城首页的中间部分-简易版

1、计算标签宽高

2、margin+padding

写html和css的节奏

* + - 1. 确认html代码的结构
      2. 通过盒子模型，计算标签所占的位置
      3. 通过定为和浮动确定元素位置
      4. 写css代码
         1. 宽高
         2. 背景
         3. 文本样式、对齐
         4. 微调
  1. 文字属性
     1. font-size

单位：px cm em

* + 1. font-family

font-family：“微软雅黑”、“黑体”、“文泉驿正黑”

* + 1. 字体粗细

font-weight：lighter（细）、normal（正常）、bold（粗体）、bolder（大粗体）

* + 1. font-style：字体样式

font-style：italic（斜体）、normal

* 1. 文本属性
     1. color：文本颜色
     2. 文本的对齐方式：

text-align：left center right

* + 1. 行高属性

line-height：

div中文本默认情况下会垂直处于行高中间的位置，如果希望文本在div中垂直居中，只需要把行高属性与div的高度设置一致

\*\*\*但如果字体过多，折行之后行高效果会在每一行生效

* + 1. 文本装饰

text-decoration：none underline（下划线） line-through（删除线） overline（上划线）

* 1. 边框属性border

border：宽度 样式 颜色

border-width:

border-color:

border-style:solid（实线） dotted（点） double（双实线） dashed（虚线）

border-top: 2px solid #f00;

border-top-color：

border-top-width：

border-top-style：

border-left同上top

border-bottom同上top

border-right同上top

border:none;去掉边框，文本框也可

边框的圆角

border-radius：3px；（圆角半径）

* 1. 常见复杂属性
     1. 溢出overflow

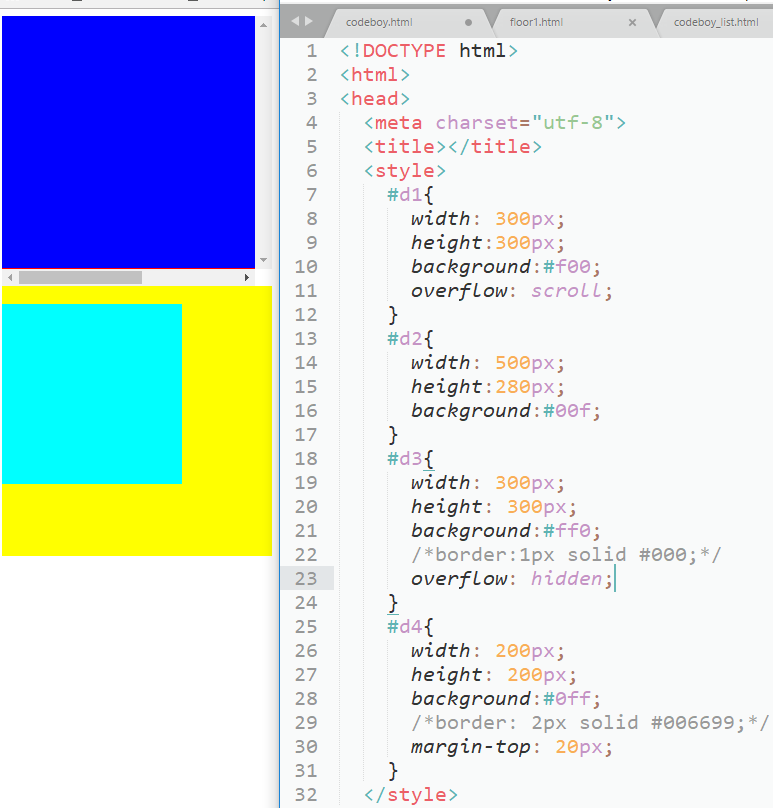
overflow：visiable（溢出属性可见）

hidden（不可见）

scroll（溢出部分变成滚动条）

div的bug：当两层div嵌套，内层div与外层div的内容区域的上线齐平（无边框的情况下），内层div如果有margin-top属性，外层div的margin-top会跟随内层的margin-top值显示

解决方案：需在外层div中加上overflow：hidden样式



* + 1. 控制元素的显示方式display

display：block；（让元素按照块级元素的规则显示）

inline；（让元素按照行级元素规则显示（一般不用））

inline-block；（行内块）

none；隐藏标签

总结：

1.行级元素特点：可以与其他行级元素共占一行，不能改变元素的height和width值（自带宽高属性的除外，如img）；行级元素的大小由内容撑开；可以使用padding、margin-left和margi-right，margin-top和margin-bottom不起作用

2.块级元素特点：独占一行；若不设置width，默认充满父级元素的宽度，若不设置height，靠内容撑开；可以正常地设置宽高；可以设置margin和padding的各个属性

3.行内块特点（不建议初学时使用）：与其他的行级元素和其他的行内块共用一行；可以设置宽高；可以设置margin和padding的所有属性；

使用行内块需要注意很多细节

1.每一个行内块都需要设置宽度

2.如果html代码中有空格，行内块会产生消除不了的空隙

* + 1. 定位
       1. 默认定位

文档流定位🡪默认文档流

* + - 1. 浮动

float：left；元素浮动起来，往左放

right；元素浮动起来，往右放

clear: both;当前元素不会被任何浮动元素遮挡

总结

1.一旦元素设置了浮动，当前元素将脱离默认文档流，在默认文档流之上进行渲染，原来的位置将被其他元素占用

2.元素只会在当前行浮动（注意判断当前行的位置）

3.浮动最重要的作用就是块级元素横向显示

* + - 1. 高度坍塌

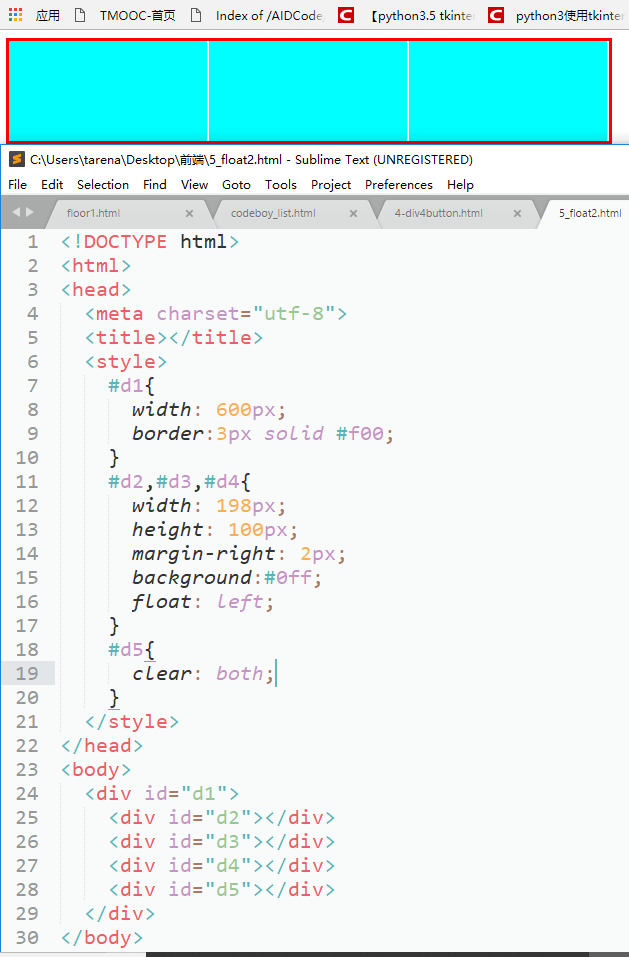
1.原理：外部div没写高度，靠内部高度撑起，而内部div都浮动了，脱离默认文档流，外部div内部认为内部没有元素了，高度为0

2.解决方案

1.给外部div加上高度，但需要提前知道高度是多少

2.给 外部div加浮动属性，但会影响外部的结构

3.在内部浮动的div下面加一个div，样式全无，仅加一条clear：both；（原理：没有内容，所以高为0，没有写宽高，所以用户不可见，所以宽为父元素的宽度，且有clear：both；样式，那么这个div就一直在浮动元素的下方显示，则父元素的高度自动匹配这个div的位置高度）



* + - 1. css中定位

position：relative；相对定位

absolute：绝对定位

1.relative：相对定位

1.定位之后，原来的位置不被占用

2.top/right/bottom/left为距离原来位置的距离

3.如果不写top/right/bottom/left，效果相当于没有定位

2.absolute：绝对定位

1. 定位之后，原来的位置会被占用

2.如果子元素有position：absolute，会向上级元素查找有position的元素，如果有一级元素带有position属性，就以这个元素的左上角为参照定位，如果没有，就以body左上角定位

3.一般情况下，absolute上级的标签会使用relative来定位，很少使用absolute

3.fixed；

## 3.JavaScript

吴华：wuhua@tedu.cn

让网页呈现出动态的数据和效果

是一种运行在客户端的解释性脚本语言

编译型：将程序的源代码编译成二进制机器语言，形成编译文件（.dll或.exe），以后再执行时可以直接调用变异的结果，优点是一次编译，多次执行

解释型：在每次执行时都需要进行编译、执行的过程，而且需要编译器进行编译、执行的操作

解释型脚本是程序执行的一种方式

说明：

JS严格区分大小写

每行JS代码必须以分号结尾

文件扩展名为js

1.创建JS文件并写入相关代码

2.在网页文件内调用相关的js文件

CSS文件合并js文件合并

1.为了减少http请求次数，减轻服务器的请求压力

2.前端框架（如天猫的Kissy）可以实现合并

### 变量

指存储在计算机内存中有名称的信息，其值在程序运行期间可以发生变化

声明变量：var 变量名称；或 var 变量名称=值；

### 数据类型

1. 字符型（string）
2. 数值型（number），包括整型和浮点类型，可以带有符号位，可以存储八进制、十进制、十六进制的数值，八进制的表示方法：0[0-7]+，十六进制的表示方法：0[x|X][0-9a-fA-F]+
3. 布尔类型（Boolean），只能存储true或false

\*\*\*在控制台输出结果console.log(typeof(age))

1. 未定义（undefined），值也只为undefined

\*\*\*以下变量的值为undefined

* 1. 声明但没有赋值的变量 （var n；console.log（typeof（n）））
  2. 声明但赋值为undefined的变量（var n=undefined；console.log（typeof（n）））、
  3. 引用对象不存在的属性时（var obj={}；console.log（typeof（obj.username）））

1. 对象（object），属性与方法的集合体

{

method：function（[parameter,…]）{

……

}

}



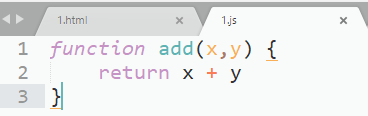
1. null（空对象）,其作用是为了断开继承链；类型为对象，值为null

var n=null;

console.log(typeof(n));---->object

console.log(n); ---->null

1. function（自定义函数），可以重复执行的代码段；



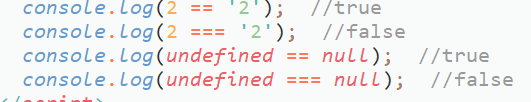


### 运算符

按类型分

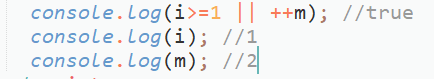
1. 字符运算符：+
2. 数字运算符：+（正号）、-（符号）、\* / % + -
3. 比较运算符：> >= < <= !=（不等于） ==（等于） ===（全等） !==

全等（===）：值相等、数据类型相同



1. 逻辑运算符：! && ||

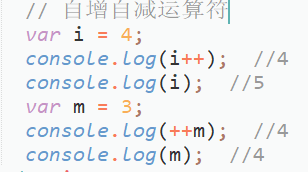
短路现象：在进行&&或||运算时，只要第一个表达式的值可以确定整个表达式结果时，后续表达式将忽略执行；



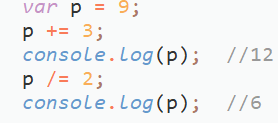
1. 自增/自减运算符：

i++ i--，后缀形式：先使用，后加减

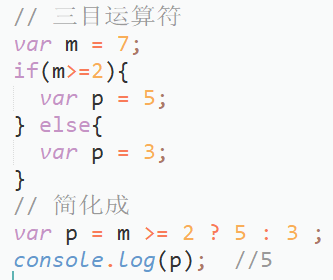
++i --i，前缀形式：先加减，后使用



1. 赋值运算符：= += -= \*= /= %=



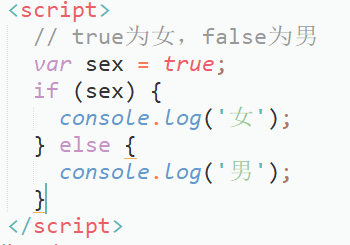
按参加运算的操作数的数量

1. 单目运算符：只有一个操作数，如负数（-），逻辑非（!）
2. 双目运算符：有两个操作数
3. 三目运算符：expression ? value1 : value2

运算符优先级：一二三赋值，

算数比较逻辑

### 流程控制语句

1. if语句

1.if（condition1）{

….条件1成立

}else{

…条件1不成立

}

2.if else语句

if（condition1）{

….条件1成立

}else if（condition2）{

…条件1不成立

}else{

…条件2不成立

}

1. switch语句

switch（expression）{

case 值1：

…

break；

case 值2：

…

break；

default

…

break

}

说明：

* 1. if语句的条件表达式只能是布尔型（可能会存在数据类型的自动转换）

switch语句的表达式可以为任意数据类型（布尔除外）

* 1. if语句每次执行时都将对变量进行取值操作，而switch只进行一次取值操作
  2. switch语句进行的是全等（===）运算

1. for语句

for（expr1；expr2；expr3）{

……

}

说明：

expr1，在循环之前无条件执行一次，一般用于变量的初始化

expr2，在循环前执行一次，以判断能否进入循环体；

expr3，在循环后执行一次，一般用于步长的计算；

for语句的表达式可以省略任意一个或全部；

1. while语句

while（条件表达式）{

……

}；

说明

* 1. 死循环：指条件永远为真的循环语句
  2. for语句和while语句同属于“当型循环（先判断，后执行）”，所以可以互换；

1. do…while循环

do{

…

…

} while（条件表达式）；

说明

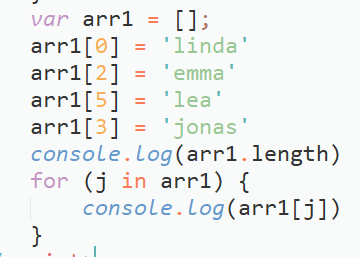
* 1. do…while语句为先执行后判断；
  2. do…while语句至少执行一次，而当型循环（for、while）至少执行零次；

1. for…in…语句

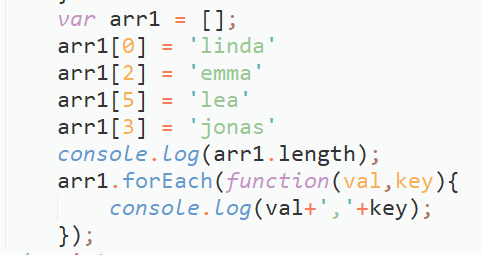
for （key in array）{

……

}



1. forEach（）语句



1. break语句：跳出本次的switch及循环语句
2. continue语句：跳过当前的循环，进行下一次循环

### 内置对象

1. String对象
   1. 构建String对象
      1. 直接量方式：’string’或”string”
      2. 构造函数：new String（'string’）
   2. 属性

length

描述：返回字符串的长度

语法：int ,object.length

* 1. 方法
     1. toLowerCase（）

描述：将字符转换为小写字母

语法：str.toLowerCase（）

* + 1. toUpperCase（）

描述：将字符转换为大写字母

语法：string object.toUpperCase（）

* + 1. substr（）

-描述：截取字符串，取length位

语法：string object.substr（start[,length]）

说明：

1.字符串从0开始编号

2.如果省略length参数，则返回start到字符串结尾之间的字符

3.如果start参数为负数，则起始值从结尾倒数

* + 1. substring

描述：截取字符串

语法：string object.substring（int start[, int end]）

说明：

1.返回的字符串包含起始位，不包含结束位

2.如果省略end参数 ，则返回到字符串结尾之间的字符

* + 1. indexOf

描述：返回字符串第一次出现的位置，如果没有则返回-1

用法：string object IndexOf（string）

console.log(str.indexOf('a')); //1

console.log(str.indexOf('k')); //-1

* + 1. lastIndexOf

描述：返回字符串最后一次出现的位置，如果没有则返回-1

string object lastIndexOf（string）



* + 1. split（）

描述：使用分隔符拆分字符串

语法：

array object.split（delimiter）

array object.split（RegExp）

## extra

UUID:通用唯一识别码，其形态是8-4-4-4-12，其作用是为保证文件名的唯一性

webp格式是Google推出的一种图像格式，其仅被Google Chrome支持，其特点是系相比jpeg文件来说，字节数更小；

京东商品展示，采用的是动态图片缩放的奇数

火狐MDN里有很多前端知识

思维导图软件：xmind

* + 1. replace（）

描述：字符串替换

语法：

string object.replace（string search，string replcement）

\*\*仅替换第一次出现的字符

string object.replace（string RegExg，string replcement）

* 1. ES6新增方法
     1. trim（）

描述：删除第一个之前及最后一个字符之后的空白（去掉两边空白字符）

语法：string object.trim（void）

* + 1. startsWith（）

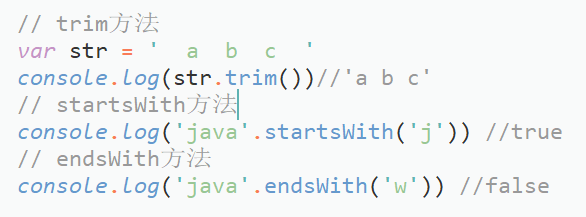
描述：检测字符是否以指定的字符开头

语法：bool object.startsWith（string）

* + 1. endsWith（）

描述：检测字符是否以指定的字符结尾

语法：bool object.endsWith（string）



1. Math对象

Math对象是一个静态对象，

* 1. 属性

Math.PI：圆周率

Math.SQRT2：2的平方根

* 1. 方法
     1. Math.ceil（）
     2. Math.floor（）
     3. Math.pow（）
     4. Math.sqrt（）
     5. Math.min（）
     6. Math.max（）
     7. Math.random（）产生0~1之间的随机数

Math.random（void）；

* + 1. Math.round（）四舍五入，仅保留到整数位

1. Date对象
   1. 构建日期对象

new Date（）

* 1. 方法
     1. getYear（）

描述：获取年份（从公元1900年至今的年份）

语法：int object.getYear（）

* + 1. getFullYear（）

描述：获取年份

语法：int object.getFullYear（）

* + 1. getMonth（）

描述：获取月份（取值范围为0-11）

语法：int object.getMonth（）

* + 1. getDate（）

描述：获取日期（xx号）

语法：int object.getDate（）

* + 1. getDay（）

描述：获取星期的第几天（0为星期天）

语法：int object.getDay（）

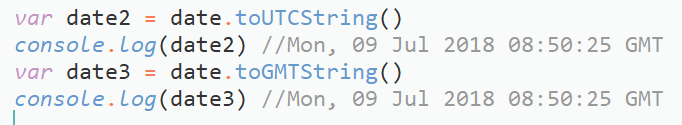
* + 1. getHours（）
    2. getMinutes（）
    3. getSeconds（）
    4. getMilliseconds（）获取毫秒
    5. getTime（）返回从公元1970年到现在的毫秒数（时间戳）



* + 1. toUTCString（）/toGMTString（）

描述：将日期转换成UTC/GMT格式

语法：object.toUTCString（）/ object.toGMTString（）



* 1. ES6新增的方法
     1. Date.now（）静态方法

描述：返回从公元1970年至今的毫秒数

语法：int Date.now（）

说明：该方法为静态方法

1. Array对象
   1. 构建Array对象
      1. 直接量方法

new Array[value1,…]

* + 1. 构造函数

new Array（length）

new Array（value1,…）

* 1. 属性
     1. length

描述：获取数组成员的数量

语法：int object.length

* 1. 方法
     1. unshift（）

描述：在数组的开头添加一个或多个成员，并返回数组的新长度

语法：int object.unshift（）

* + 1. push（）

描述：在数组的末尾添加一个或多个成员，并返回数组的新长度

语法：int object.push（）

* + 1. shift（）

描述：在数组的开头添加一个或多个成员，并返回数组的新长度

语法：mixed object.shift（）

* + 1. pop（）

描述：在数组的开头添加一个或多个成员，并返回数组的新长度

语法：mixed object.shift（）

* + 1. join（）

描述：将数组成员使用指定分隔符连接成字符串

语法：string object.join（delimiter分隔符）

* + 1. isArray（）

描述：判断是否为数组，静态方法

语法：bool Array.isArray（obj）



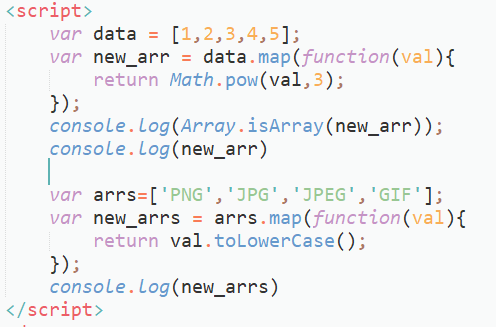
* + 1. map（）

描述：数组成员一次调用回调函数，并且返回新数组

语法：array object.map（function（value[,index]）{

…

}）



1. Regular Expression对象（正则表达式对象）
   1. 构建正则表达式对象
      1. 直接量方式

/正则表达式/修正符

* + 1. 构造函数

new RegExp（”正则表达式”[，”修正符”]）

* 1. 正则表达式
     1. 元字符
        1. \d,匹配数字，等价于[0-9]
        2. \D,匹配非数字，等价于[^0-9]
        3. \s,匹配空白字符，等价于[\n\r\t\v\f]
        4. \S匹配非空白字符，等价于[^\n\r\t\v\f]
        5. \w,匹配任意的数字字母下划线，等价于[0-9a-zA-Z\_]
        6. \W,匹配任意的非数字字母下划线，等价于[^0-9a-zA-Z\_]
        7. . ，匹配除换行符外的任意字符
     2. 量词
        1. ?，出现零次或一次，等价于{0,1}
        2. \*，至少出现零次，等价于{0，}
        3. +，至少出现1此，等价于{1，}
        4. {m}，出现m次
        5. {m,}，至少出现m次
        6. {m,n}，至少出现m次，至多出现n次
     3. 其他
        1. |，或者
        2. ^，以指定的字符开头
        3. $，以指定的字符结尾
        4. []，表示范围，如[0-9]
        5. [^]，表示不在范围内
     4. 转义符
        1. \n,\r,\t,\b,\f,\\
        2. \.
        3. \\*
        4. \?
        5. \|
     5. 捕获组（…）与非捕获组（?:…）

* + 1. 修正符
       1. g（global），全局
       2. i（ignore），忽略大小写



* 1. 方法
     1. test（）

描述：检测字符串是否符合正则表达式规范

语法：bool object.test（）



### DOM编程

JavaScript由ECMAScript（简称ES）、DOM和BOM组成；

ECMAScript：关键字、保留关键字、流程语句、数据类型、内置对象；

DOM：提供HTML/XML文档的相关API；

DOM由DOM Core、HTML DOM和XML DOM组成；

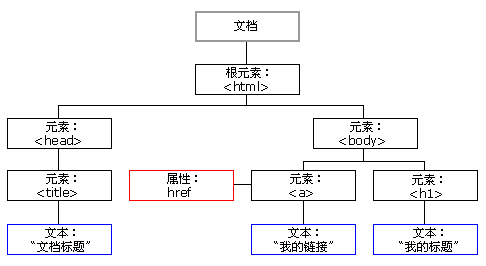
DOM Core：提供处理HTML和XML文档的API；

XML DOM ：提供处理XML文档的API

html DOM：提供处理HTML文档的API

#### 节点（Node）

在DOM树中一切皆节点



1. 节点类型
   1. 1，代表元素节点（HTML标记）
   2. 2，代表属性节点（HTML标记的属性）
   3. 3，代表文本节点（纯文本）
   4. 8，代表注释节点（HTML注释）
   5. 9，代表文档节点（整个HTML文档）
2. 节点关系
   1. 子节点：某节点是另一个节点的直接下一级节点；如<a><b>…</b></a>中，a为b的父节点，b为a的子节点；
   2. 父节点
   3. 兄弟节点：同一个父节点的所有子节点互为兄弟节点
3. 节点的属性
   1. nodeName

描述：获取节点的名称

语法：string node.nodeName

* 1. nodeValue

描述：获取节点值

语法：string node.nodeValue

* 1. nodeType

描述：获取节点类型

语法：int node.nodeType

* 1. parentNode

描述：返回节点的父节点

语法：node node.parentNode



* 1. childNodes

描述：返回节点内所有的子节点形成的集合(数组)

语法：NodeList node.childNodes

* 1. firstChild

描述：返回指定节点的第一个子节点

语法：node node.firstChild

* 1. lastChild

描述：返回指定节点的最后一个子节点

语法：node node.lastChild

* 1. nextSibling

描述：返回当前节点的下一个兄弟节点

语法：node node.nextSibling



* 1. previousSibling

描述：返回当前节点的上一个兄弟节点

语法：node node. previousSibling

1. nodeValue（）

描述：

语法：



1. 节点方法
   1. appendChild（）

描述：追加子节点

语法：Node（新插入的子节点） node.appendChild（Node）

* 1. removeChild（）

描述：删除子节点

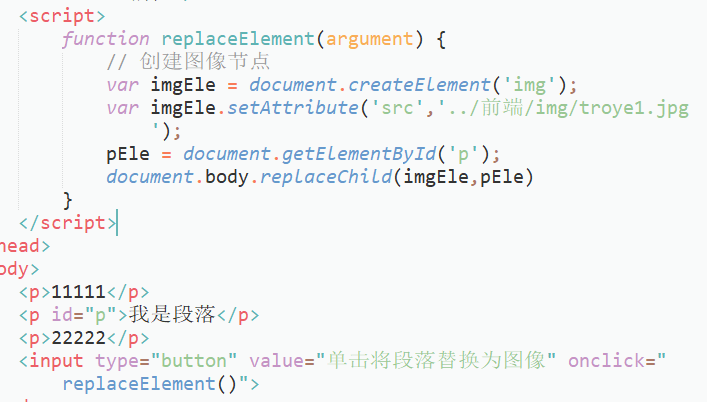
语法：Node node.removeChild（Node）



* 1. replaceChild

描述：节点替换

语法：node node.replaceChild（newChild，oldChild）



#### document对象（继承自Node）

1. 属性

documentElement

描述：返回文档的根元素

语法：Element document.documentElement

1. 方法
   * 1. getELementById（）

描述：根据对象ID来获取元素

语法：Element document.getELementById（string id）

* + 1. createElement（）

描述：创建元素节点

语法：Element document.createElement（string tagName）

* + 1. getElementByTagName（）

描述：获取指定元素中包含指定标记名称所形成的集合（数组）

语法：NodeList Element.getElemenByTagName（string TagName）（单个对象内）

NodeList document.getElemenByTagName（string TagName）（整个文档）

* + 1. createTextNode（）

描述：创建文本节点

语法：TextNode document.createTextNode（string）



#### Element对象（继承自Node）

1. 方法
2. getAttribute（）

描述：获取属性值

语法：string Element.getAttribute（string name）

1. setAttribute（）

描述：设置属性

语法：void Element.setAttribute（string name，string value）

1. removeAttribute（）

描述：删除属性

语法：void Element.removeAttribute（string name）

1. hasAttribute（）

描述：返回是否存在属性

语法：Boolean Element.hasAttribute（string name）

1. getElementByTagName（）

描述：获取文档中指定标记名称所形成的集合（数组）

语法：NodeList document.getElemenByTagName（string TagName）（整个文档）

NodeList Element.getElemenByTagName（string TagName）（单个对象内）



### HTML DOM

HTML DOM：提供处理HTML文档的API

1. 选取对象

Element doucment.getElementById（string id）

NodeList doucment.getElementsTagName（string TagName）

NodeList Element.getElementsTagName（string TagName）

NodeList document.getElementsByName（string name）

1. 属性控制
   1. 对于单个单词的HTML标记属性即HTML DOM对象属性（class属性除外，因为为关键字）

pEle = document.getElementById（’p’）;

pEle.align = ‘center’;

* 1. 对于合成词的HTML标记属性（如table标记的cellspacing等）在HTML DOM编程时，采用“小驼峰标记法”命名（如cellSpacing）

tableEle = document.getElementById('table');

tableEle.cellPadding = '10px';

* 1. 对于HTML标记的class属性，在HTML DOM编程时使用className取代

var pEle = document.getElementById('p')

pEle.className = 'f1';

* 1. 对于HTML标记的不二属性（如单/复选框的checked，列表项的selected属性）在HTML DOM编程时采用boolean类型表示

checkEle = document.getElementById('check');

checkEle.checked = true;

checkEle.disabled = true;

* 1. 在HTML DOM编程时，某些HTML DOM对象有自己特有的属性或方法；

1. HTMLDOM方法
   1. getElementByName（）方法

描述：获取文档中name属性相同的对象的组成的集合（数组）

语法：NodeList document.getElementsByName（string name）

## 4.jQuery-js框架

提高JavaScript开发效率

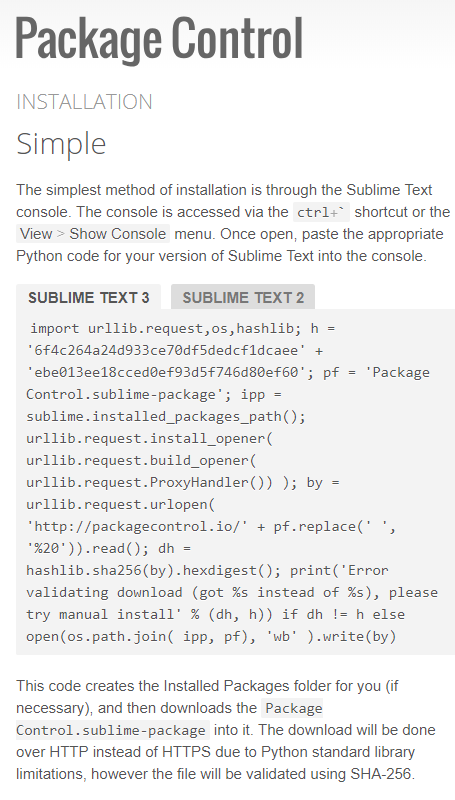
## 二、浏览器介绍

Chrome/Safari🡪webkit

Firefox🡪mozilla

IE

opera



## sublime插件安装

1. 安装packagecontrol

https://packagecontrol.io/installation

* 1. 复制相关版本代码
  2. 打开sublimeText3控制台（ctrl+`）
  3. 粘贴代码

1. 安装插件
   1. ctrl+shift+p，输入install回车
   2. 输入插件名称
2. 常用插件
   1. Emmet
   2. AutoFilename
   3. jsformat