

1、html（超文本标记语言）

xml（可扩展的标记语言）

4、html5(h5)

语义化的标签

- `<header>`：页眉通常包括网站标志、主导航、全站链接以及搜索框。
- `<nav>`：标记导航，仅对文档中重要的链接群使用。
- `<main>`：页面主要内容，一个页面只能使用一次。如果是web应用，则包围其主要功能。
- `<article>`：定义外部的内容，其中的内容独立于文档的其余部分。
- `<section>`：定义文档中的节（section、区段）。比如章节、页眉、页脚或文档中的其他部分。
- `<aside>`：定义其所处内容之外的内容。如侧栏、文章的一组链接、广告、友情链接、相关产品列表等。
- `<footer>`：页脚，只有当父级是body时，才是整个页面的页脚。
- `<small>`：呈现小号字体效果，指定细则，输入免责声明、注解、署名、版权。
- ``：和 `em` 标签一样，用于强调文本，但它强调的程度更强一些。
- ``：将其中的文本表示为强调的内容，表现为斜体。
- `<mark>`：使用黄色突出显示部分文本。
- `<figure>`：规定独立的流内容（图像、图表、照片、代码等等）（默认有40px左右margin）。
- `<figcaption>`：定义 `figure` 元素的标题，应该被置于 `figure` 元素的第一个或最后一个子元素的位置。
- `<cite>`：表示所包含的文本对某个参考文献的引用，比如书籍或者杂志的标题。
- `<blockquote>`：定义块引用，块引用拥有它们自己的空间。
- `<q>`：短的引述（跨浏览器问题，尽量避免使用）。
- `<time>`：datetime属性遵循特定格式，如果忽略此属性，文本内容必须是合法的日期或者时间格式。
- `<abbr>`：简称或缩写。
- `<dfn>`：定义术语元素，与定义必须紧挨着，可以在描述列表dl元素中使用。
- `<address>`：作者、相关人士或组织的联系信息（电子邮件地址、指向联系信息页的链接）。
- ``：移除的内容。
- `<ins>`：添加的内容。
- `<code>`：标记代码。
- `<meter>`：定义已知范围或分数值内的标量测量。（Internet Explorer 不支持 meter 标签）

`<progress>`：定义运行中的进度（进程）。

5、表单域

文本框，密码框，单选，复选，下拉列表，隐藏域，文本域

```
1 <form action="02.html" method="get">
2   文本框: <input type="text" value="111111"><br>
3   密码框: <input type="password"><br>
4   单选按钮: 男<input type="radio" name="sex">女<input type="radio" name="sex"><br>
5   复选按钮: 足球<input type="checkbox">篮球<input type="checkbox"><br>
6   下拉列表: <select>
7               <option>aaaaa</option>
8               <option>bbbbbb</option>
9               <option>cccccc</option>
10            </select><br>
11   隐藏域:<input type="hidden" value="1234"><br>
12   文件上传: <input type="file"> <br>
13   普通按钮: <input type="button" value="普通按钮"> <br>
14   提交按钮:<input type="submit" value="提交按钮"> <br>
15   重置按钮:<input type="reset"> <br>
16 </form>
```

-----CSS-----

1、css引入

1) 内联

2) 内部引用

```
1 <style type="text/css">
2   css
3 </style>
```

3)外部引入

```
1 <link rel="stylesheet" href="first.css">
```

2、常见选择器

id，类，元素，属性

#id, .class, 标签名, 标签名[属性名称='属性值']

选择器 返回值 描述 * 返回集合 选取页面中的所有元素 .class 返回集合 选择指定的class类的所有元素 #id 返回单个元素 选择指定的ID的元素 element 返回集合 选择所有匹配标签的元素

2、字体，背景，文本，盒子模型、定位（绝对定位，相对定位），鼠标的操作（类型），浮动，显示隐藏

```
1  /* 字体大小 单位像素 */
2      font-size: 20px;
3      /* 字体加粗 默认normal*/
4      font-weight: bold;
5      font-weight: bolder;/* 更粗，部分浏览器不支持 */
6      font-weight: lighter;/* 变细，部分浏览器不支持 */
7      font-weight: normal;
8      /* 字体设置 */
9      font-family: "黑体";
10     /* 字体样式 */
11     font-style:italic;/* 斜体 */
```

```
1  /* 添加背景图片 */
2      background-image: url("../picture/zhouxingchi.jpg");
3      /* 设置图片尺寸 */
4      background-size:300px 300px;
5      /* 设置图片不重复 */
6      background-repeat:no-repeat;
```

```
1  /* 文本左右居中 */
2      text-align: center;
3      /* 文本上下居中 */
4      line-height: 30px;/*行高的值等于元素的高度，则文本垂直居中*/
5      /* 默认left左对齐 */
6      text-align:right;
7      text-align: center;
8      /* 文本修饰 */
9      text-decoration: none;/* 无任何修饰 */
10     text-decoration: underline;/* 下划线 */
11     text-decoration: overline;/* 上划线 */
12     text-decoration: line-through;/* 删除线 */
13     /* 文本颜色 */
14     color: red;
```

盒子模型：

margin：外边距，指元素边框距相邻元素或上级元素边框的距离，有上下左右4个值。

border：边框宽度，有上下左右4个值

padding：内边距，有上下左右4个值

```

1  /* 单独设置外边距 */
2      /*
3      margin-left: 50px;
4      margin-top: 50px;
5      margin-bottom: 50px;
6      */
7      /* 批量设置外边距 */
8      /* margin: 50px; */    /* 上下左右各50px */
9      /* margin: 10px 30px; */    /* 上下10px 左右30px */
10     /* margin: 0px auto; */    /* 上下0px 左右居中 */
11     margin: 10px 20px 30px 40px;    /* 上10px 右20px 下30px 左40px 顺时针 */

```

定位:

相对定位: position: relative 元素不会脱离文档流 初始位置大小不会改变, 坐标位置相对于自己的初始位置的四个边, 相邻元素位置不变。

绝对定位: position: absolute 绝对定位会脱离文档流 (相对于窗口左上角或祖先元素中距离最近的做过非static定位的元素的位置), 初始位置会改变, 相邻元素位置会变。

固定定位: position: fixed 元素脱离文档流。固定在窗口的某个位置, 位置坐标相对于窗口的左上角。

应用场景: 元素位置相对于窗口, 并且不会随着内容位置改变而改变。

浮动定位: float: left/right 元素脱离文档流, 只能在元素所在行内左右浮动, 当撞到父元素边框或另一个浮起来的元素时停止。如果一行装不下浮动的元素会自动折行 (换行的原则为不能被挡住, 需要完全显示)。

应用场景: 当多个纵向排列的元素, 需要变成横向排列时, 使用浮动定位。

-----js-----

js基础

1、js中的引用方式

1) 内联

```

1  <body>
2      <input type="button" onclick="alert('第一种引入方式')" value="我是按钮">
3  </body>

```

2) <script> 添加

```

1  <script type="text/javascript">
2      function a(){
3      }
4  </script>

```

3) 外部引入

```
1 <script type="text/javascript" src="first.js">
2     /* 这里面不能再写js代码 */
3 </script>
```

(css一般放在上面, js一般放在底部)

2、全局变量

1、在函数外部用var声明的变量 2、在函数内部不用var声明的变量 局部变量 1、在函数内部用var声明的变量

3、数据类型

undefined, number, string, boolean, object Object, Array, Date、Function

typeof: 查看数据类型

1、隐式转换

自动转换

2、显示转换

1、转换为number

parseInt parseFloat Number(i);

2、转换成字符串

String(): 对null和undefined不会报错 toString(): 对null和undefined会报错

3、转换成boolean

Boolean():非0转为true, 0, null,undefined为false

4、与=

==: 等值符 ===: 等同符

5、Object

1、创建: A、new的形式 var o = new Object(); o.name = "xx"; B、字面量表示法 var o = {id:10,name:"zs"}; 2、访问属性: A、对象名.属性 B、对象名["属性"] 推荐使用对象名.属性, 但是一种情况下(当属性是变量的时候)必须使用第二种形式

3、添加属性 对象名.属性 = 值 4、删除属性 delete 对象名.属性 5、对象的冻结 Object.freeze(对象名)

6、Array

1、创建 1、new 2、字面量表示法[], [{},{}]

2、常见方法 concat: 连接 join: 对数组中的元素使用某个连接符连接起来 pop:从数组中弹出最后一个元素, 返回值: 弹出的元素 元素中的最后一个元素会被删除 push: 向数组的末尾添加元素 返回值代表新添加进去之后数组的长度 原数组是会改变的 shift:删除元素 从数组的第一个元素开始删 返回值是删除的元素, 原数组会改变 unshift:向数组的开始添加元素 返回值是新数组的长度 slice: 截取数组中的元素 [start,end) 返回值是截取到的元素 原数组是不会改变的 splice: 向某个位置添加元素 返回值是原数组中被删除的元素 原数组改变 sort: 排序 reverse: 逆序

7、Function

1、创建 1、声明式方式 function a(){} 2、表达式方式 var a = function(){} 3、new 2、声明式方式和表达式方式的区别?

声明式方式在预执行期产生, 表达式在执行期产生。

3、js中的函数没有重载

8、事件

常见事件: onclick,onfocus,onblur,onchange,onselect,onload,onunload,onSubmit

a.窗口事件,只在body和frameset元素中才有效 onload 页面或图片加载完成时 onunload 用户离开页面时

b.表单元素事件,在表单元素中才有效 onchange 框内容改变时 onsubmit 点击提交按钮时 onreset 重新点击鼠标按键时 onselect 文本被选择时 onblur 元素失去焦点时 onfocus 当元素获取焦点时

c.键盘事件,在base,bdo,br,frame,frameset,head,html,iframe,meta,param,script,style,title元素里都无效 onkeydown 按下键盘按键时 onkeypress 按下或按住键盘按键时 onkeyup 放开键盘按键时

d.在base,bdo,br,frame,frameset,head,html,iframe,meta,param,script,style,title元素里都无效 onclick 鼠标点击一个对象时 ondblclick 鼠标双击一个对象时 onmousedown 鼠标被按下时 onmousemove 鼠标被移动时 onmouseout 鼠标离开元素时 onmouseover 鼠标经过元素时 onmouseup 释放鼠标按键时

dom部分

1、获取节点

直接获取 id: document.getElementById(""); name: document.getElementsByName(""); tagName: document.getElementsByTagName(""); 间接获取 根据父亲获取儿子: 获取所有的儿子: parent.childNodes 获取第一个儿子: parent.firstChild 最后一个儿子: parent.lastChild 根据儿子获取父亲节点: child.parentNode 找兄弟节点: 节点.nextSibling 节点.previousSibling

2、节点的类型

nodeType 1: 元素节点 3: 文本节点 nodeName:元素结点返回结点类型,文本结点返回"#text" nodeValue: 文本节点调用nodeValue会拿到文本值 如果是元素节点拿到的值为null

3、属性的操作:

```

1  设置属性:
2      节点.setAttribute(属性名,属性值)
3      节点.属性名 = 属性值;
4
5  获取属性:
6      节点.getAttribute(属性名);
7      节点.属性名
8  移除属性:
9      节点.removeAttribute(属性名);
10     节点.属性 = null

```

4、innerHTML, innerText, outerHTML,outerText

innerHTML:节点内部所有元素和文本 innerText: 节点内部的所有文本 outerHTML: 节点内部的所有元素和文本 包括节点本身 outerText: 节点内部的所有文本 包括节点本身

5、元素节点增删改替换

创建元素节点 var divElement = document.createElement("div"); 创建文本节点 var text = document.createTextNode("Hello javascript"); 添加节点 parentNode.appendChild(childElement); 插入子节点 parentNode.insertBefore(newNode,beforeNode); 取代子节点 parentNode.replaceChild(newNode,oldNode); oldNode必须是已经存在, 不然会发生异常 删除子节点 parentNode.removeChild(childNode);

6、表单的处理

- 1、获取表单 dom节点的获取方式有效 document.formName
- 2、表单的属性 action:form的提交地址 method: get:默认get提交 提交的数据的内容会展现地址栏上 post:提交的数据不会显示在地址栏上 name: length:表单域的个数 elements: 所有的表单域
- 3、表单的方法 submit() reset()
- 4、事件 onSubmit() onreset()

7、表单中的元素的处理

1、文本类(text,password,textarea,hidden) 获取值:value 设置值:对象.value=值 2、单选、复选 (遍历) 3、下拉列表 获取所有的option :下拉列表对象.options 获取选中的option的Index:下拉列表对象.selectedIndex 获取选中的option: 下拉列表对象.options[下拉列表对象.selectedIndex] 获取选中的值:下拉列表对象.value 设置某个option选中:下拉列表对象.value=值

级联

bom部分

1、document、window、location、history、navigator 2、window: confirm() setTimeout() clearTimeout() setInterval() clearInterval()

location: href reload history: go(1),go(-1),go(n),back(),forward()

navigator: userAgent

面向对象

1、函数没有重载，函数可以作为参数传递，函数可以作为返回值 2、arguments：代表的是实参构成Array arguments.callee:代表的是当前这个函数 length:形参的个数 3、call和apply：可以将某个函数动态的绑定到某个对象上，相当于就在这个对象上面定义了这个方法一样。call：参数得列举出来 apply：封装成数组的形式 4、深刻理解prototype function Person(){}: 在Person中有一个属性prototype指向Person的prototype，在Person的prototype里面有一个属性constructor指向Person函数

```
1 Person.prototype.name = "zs";
2 Person.prototype.age = 20;
3 Person.prototype.say = function(){}
4 向Person的prototype上面添加了2个属性(name,age)和一个say方法
5
6 var p = new Person();
7 在p对象中有个属性_proto_ 指向Person的prototype
8
9 p.say();
10 调用p的say方法，但是p没有say方法，这时就会上prototype中去找say，如果有就完成调用
```

5、作用域链

1、js中没有块级作用域 2、作用域链

6、闭包

```
1 function a(num){
2     function b(a){
3         return num + a;
4     }
5
6     return b;
7 }
```

7、匿名函数

没有名字的函数称作“匿名函数”。

```
1 (function(){
2     //这里忽略jQuery所有实现
3 })();
4 (function(){ //这里忽略jQuery所有实现 })();
```