11.8

今天我开始入手制作一个属于我的网页，我今天学习了HTML的骨架与基本内容

如图所示，

<!DOCTYPE html>是声明这个是HTML文件

<head>内部主要放的是一些基本信息

<body>里面才是我们的主体部分

其实最需要记忆的是一些标签及其属性

|  |  |
| --- | --- |
|  |  |
|  | |
|  | |
|  | |

总结：

一、<a>中的href属性是超链接默认属性，通常配合target，用来配合控制打开窗口方式，注意target=\_blank 这是最常用的

|  |
| --- |
|  |

常见的href链接对象：

1. 链接到其他网页
2. 连接到站内其他的页面
3. 连接到页面内的特定位置，通常是自己设置的锚点

|  |
| --- |
|  |

请以此为依据，学习一下定义锚点的方式，id通常是有这种一一对应的关系

1. 邮箱地址
2. 电话联系方式
3. input之后的type属性可以有多种，常用的：password，text，radio，checkbox，date，均有不同的作用，我在今天的案例中也有所体现，可以看具体的代码实现；而value则是初始内容和提交值，两者是有所搭配的；id可以说是一个独一无二的idcard，在超链接的时候也可以用得上，在pwd的时候也可以用上，作用是提供一种独特的特征识别，全文id不能出现两个一模一样的；placeholder就是不输入时存在，用户输入时就会隐藏。
4. div的作用是创建一个可以包含其他HTML元素的容器块，没有任何语义，创建页面布局结构：页眉、导航栏、页脚之类。
5. span是用来内联样式化文本，后面学CSS时候是可以用到的。

11.9

今天我开始学习CSS，主要学习了CSS的导入方式和选择器的使用方法

|  |  |
| --- | --- |
|  | |
|  |  |

1. 可以看到一共有三种导入CSS的方式，要注意它们之间的优先级关系，优先级高和优先级低的导入方式在作用于同一个文本时，前覆盖后。
2. 选择器的种类是很多的，注意在不同的情况下选择不同的选择器。其中，对于伪类做一下了解，这表示一种临时性的状态。

|  |
| --- |
|  |

|  |  |
| --- | --- |
|  | 7687b7a23628e438bb58bdb8f46d1778 |

1. 然后又复习一下这个块之类的，前两个可以说是比较常用的形式。

11.10

今天我继续学习了复合属性的概念，学习了盒子模型，初步了解了网页内容布局的原理以及三种方法：标准流（网页按照元素的书写顺序依次排列）、浮动、定位。下面我再细化分析一下。

1. 首先关于复合元素有一个例子便是font，这个的用法就是，CSS中有上百种元素，但其实并不是需要记住每一种，如”font：50px ‘KaiTi’;”其实就相当于同时调用了大小和字体这两种属性，所以我们可以通过这种复合属性来减少我们记忆的属性个数。
2. 下面是对于昨天的块元素，行内元素，行内块元素的分析：值得注意的就是inline无法去控制宽高，inline-block却可以；

|  |  |  |
| --- | --- | --- |
| Block | Inline | Inline-block |
|  |  |  |

如果你想改变某一种标签的属性，你可以使用display，具体的语法实现是如图，只需要display：后面加上你的目标元素即可，改变之后原标签属性完全改变。

|  |
| --- |
| f8966beee9262d1ed9199421dce20a64 |

三、第三个介绍盒子模型

|  |  |
| --- | --- |
| ef4df1a07e07fb8d75f0c91f2e90b75a | a8eda141ab23183e72ea51135dcefd36 |

盒子模型是我们的一种页面布局的基本概念，页面的布局效果和上图类似，分为四个部分，我们可以分别对这四个内容进行属性的改变，从而影响最终的页面布局效果

|  |  |
| --- | --- |
| b40ffa177239069a94e91531f38f7d0e |  |

从上图可以明确对盒子模型四个内容的添加属性方式，并且作用域也根据调用方式有所不同。

四、下一个部分是有关于网页布局方式，其中标准流是CSS默认的方式，是由块级元素和行内元素按照默认规定的方式来排列，块级就是占一行，行内元素一行可以放好多个元素，一行满了之后才会换行；在这里我们还要介绍浮动和定位，语法实现如图。

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| 02297f6ec3751aee747e4eb06dcc9f0d | |  | |
|
|  | | | |
|  |  | | 通过图片可以看到，一共有relative，absolute，fixed，三种，其中relative是依照目前标准流位置做出调整，后二者都是跳脱出了标准流的内容，而fixed更是以浏览器为范围去调整固定位置的。 |
|  | | | |
|  | | 我认为，前两种不太直观，所以我又辨析了一下，这个position：relative，也就是说，绝对定位是需要设置一个祖先元素作为基准的，手动设置，定位祖先的标准请参考上图，且关键点在于找到最近的，如果一直没有会找到body。 | |
| 浮动会脱离标准流，并且特点是在同一行上，可以选择靠近哪一侧。  并且介绍两种解除浮动的方法，伪元素清除法和overflow，这样就可以消除浮动，防止高度被破坏。 | |
| 什么是粘性定位？  粘性定位是relative和fixed定位的混合体，元素在滚动到特定阈值前相对定位，之后变为固定定位。 | |

11.11

今天我开始接触了JavaScript，并且学习了解了他的导入方式，作用，和基础语法。

1. 首先明确JavaScript是什么和它的作用：

|  |
| --- |
| bbe4a571cee1aac344358cc6c8c95a26 |

通过引入JavaScript，我们可以使得网页可以根据用户的操作动态变化，并且可以通过网页交互，这个是HTML与CSS不同的

|  |
| --- |
|  |

所以可以说这个是前端设计的灵魂！

二、下面介绍两种导入方式，一种是内联样式，一种是外联样式，可以在head，也可以在body。

|  |
| --- |
|  |

且是有打印顺序的，注意图中的外联样式的console.log是在myscript.js文件中的。

三、JavaScript里的定义函数和初始化的方式一共有三种：

|  |  |
| --- | --- |
|  | 这里注意区分NULL和undefined，NULL是明确的空值，而后者是指已经定义但没有初始化。  并且尽量用let，var可能不太稳定。 |

四、下面我们看一下JavaScript的一个重要部分——事件。

|  |  |
| --- | --- |
| 653382cc1d009a6c5e989df52e4bdd48 | |
|  |  |

事件有很多种，并且从图上可以看出来，可以同时发生多个事件。

11.12

今天学习了JavaScript里的DOM，以及对于基础的增删改查的功能做了一个小练习。

|  |  |
| --- | --- |
|  |  |

上面两张图片很清楚的展示了DOM的框架，但其实DOM具体的概念不是很遥远，我们使用的全部标签啥的都和DOM有关系。

一、第一个部分关于这个getElement 后面可以添加不同的文本内容比如id，tag，class，值得注意的是，id一般是一对一的，这因为id是独一无二的，但是其他的基本是返回的是数组，所以这种时候我们利用下标[]，来返回元素。

|  |
| --- |
|  |

修改选择器标签文本内容也有很多方法，比如.innerHTML .innerText .innerCell，不同的方法适合于不同的情况，Cell一般是针对table的，Text一般只针对于文本。接着还有DOM绑定事件比如是addEventListener这个方法，它可以提供两个参数，事件和触发事件之后的函数。

1. 接下来我们对于增删改查这个基础操作做了一个练习。我认为通过这个可以培养对于一个页面基础操作的熟练程度，具体的代码如下。

|  |
| --- |
|  |
|  |
|  |

具体的效果就是这样。

我认为特别需要注意的就是this的用法，this是指当前触发事件的元素。

|  |  |
| --- | --- |
|  | 通过明确this的节点位置，我们就可以接着利用parentnode，childnode这样的方法，去在这个DOM树上去继续索引，去对于其他的的节点进行操作。  例如：  var row = button.parentNode.parentNode;  就是从button开始索引到tr |

|  |
| --- |
|  |

11.13

今天我继续学习了响应式，即移动端的布局，这个主要解决的是你写的页面在不同的设备有可能显示的情况不同。

1. 主要了解了现阶段常见的解决的响应式布局的方法：

|  |
| --- |
| eb4d6a0e634ac493aae5ca58f3a265f4 |

我们详细的来介绍一下，首先我们需要了解基础知识——Viewport

|  |
| --- |
| e8c932453e25d2780f753ca5225112d6 |

通常我们只会使用第一行，这也是默认值，这个就管理了我们用户在与我们页面交互的时候，看到我们页面的情况。但如果我们在不同的设备上浏览呢，情况一般会是不同的，所以我们为了使在不同设备上都具有良好的浏览体验，我们就需要考虑移动端布局方法。

也就是@media和rem单位的方法。

|  |  |
| --- | --- |
|  | 其实rem，vw/vh等单位不能够与@media完全割裂的实现响应式布局，通常都是相互协作，共同完成的，而其中的media是宏观层面分类不同的设备，解决例如：单列→多列、显示→隐藏、垂直→水平的问题；而rem等解决如何在保持布局结构的前提下平滑缩放尺寸比如：字体大小、间距、容器宽高的渐进变化。 |

二、接下来我们再看看实际解决布局的Flex盒子，他主要解决的是复杂的一维布局。调用方式是display：flex，可以控制项目在主轴和交叉轴上的排列、对齐和分布。

|  |  |
| --- | --- |
| 77aac46ef075d0ad4c2df02e58f209fe | |
| 1ae54161ced323db17e886a107b7692d | 298c71aba3de971041e2dc1248cc34a1 |
| c91d440f172a3c2387729b1a3591de4f | 1e999eeb226b9f5848bed6b3b81103db |
| b84cfa8dc5ccd062754190a45c01eda4 | 81253cc012c2003bb52f46e383bdf16e |

具体的可以通过尝试来自行体会。