# HTML/CSS

## 居中

1. margin : 0 auto; 水平
2. text-align: center; 水平
3. line-height: xxpx; 垂直
4. 尺寸固定，50%，定位，margin负距 中心
5. 尺寸固定，四方位0，定位，margin:auto; 中心
6. 尺寸不固定，padding中心
7. 尺寸不固定，利用inline-block、vertical-align配合伪元素中心（张鑫旭）
8. 尺寸不固定，transform:translate(-50%,-50%) 中心
9. flex布局 水平、垂直、中心

## 盒模型

1. box-sizing: content-box（标准W3C模式）

宽高不包括padding和border，就是content的宽高

1. box-sizing: border-box（IE盒模型:）

宽高是border（包括border）以内

## 浮动

1. 清除浮动的方法：

使用after伪元素清除浮动（推荐使用）

额外标签法（在最后一个浮动标签后，新加一个标签，给其设置clear：both；）（不推荐）

1. 不清除浮动产生的问题：

当父元素不给高度的时候，内部元素不浮动时会撑开, 而浮动的时候，父元素变成一条线, 造成塌陷。

## 纯css实现三角形

.box {

width:0px;

height:0px;

border-top:50px solid rgba(0,0,0,0);

border-right:50px solid rgba(0,0,0,0);

border-bottom:50px solid green;

border-left:50px solid rgba(0,0,0,0);

}

## Flex布局

解决响应式问题、解决浮动产生的问题、实现垂直居中

1. flex-wrap控制换行(默认不换行) wrap换行
2. justify-content主轴对齐方式
3. align-items交叉轴对齐方式

## css优先级

选择器的特殊性值表述为4个部分，用0,0,0,0表示。

ID选择器的特殊性值，加0,1,0,0。

类选择器、属性选择器或伪类，加0,0,1,0。(伪类:last-child)

元素和伪元素，加0,0,0,1。(伪元素：：after)

通配选择器\*对特殊性没有贡献，即0,0,0,0。

最后比较特殊的一个标志!important（权重），它没有特殊性值，但它的优先级是最高的，为了方便记忆，可以认为它的特殊性值为1,0,0,0,0。

## 常用块级、内联元素、空元素

1. 块级

div dl ol ul form p table

1. 内联

a em i img span input label select strong textarea

1. 空元素

<br/> <hr> <input> <img> <link>

## Div

div默认宽度为auto

# JavaScript/ES6

## 闭包

一个函数引用着另外一个函数的变量，因为变量被引用着，不会被回收，因此可以用来封装一个私有变量，这是优点也是缺点，不必要

的闭包，只会徒增内存消耗。

## typeOf值类型

"number"、"string"、"boolean"、"object"、"function" 和 "undefined"。

"symbol" —— ES6引入的一种新的原始数据类型Symbol，表示独一无二的值

## JS单线程，异步等

## 变量名

第一个字符必须是一个ASCII字母（大小写），或一个下划线（\_）。

第一个字符不能是数字，后续的字符必须是字母、数字或下划线。

变量名称一定不能是保留字。

## null和undefined

1. null表示不存在的对象，转化数值时为0；undefined表示“无”原始值，转换成数值为NaN，当变量提升时，为undefined
2. 常见undefined的几种情况：

变量被声明了，但没有赋值时，就等于undefined。

对象没有赋值的属性，该属性的值为undefined。

调用函数时，应该提供的参数没有提供，该参数等于undefined。

函数没有返回值时，默认返回undefined。

1. 常见null：

作为函数的参数，表示该函数的参数不是对象。

作为对象原型链的终点。

## 数组去重的几种实现方式

## 拖拽事件写法

鼠标按下，获取鼠标的坐标点、移动对象的距离值；

鼠标移动，获取移动坐标点，求移动的偏移量，移动对象设置移动偏移量+初始距离值；

鼠标松开，清空鼠标按下事件。

## 事件模型

事件冒泡/事件捕获

## this指向

js的this指向是不确定的，是可以动态改变的，this 一般情况下，都是指向函数的拥有者。

1. 在全局作用域下，this默认指向window对象。
2. 处在函数中的this，

一般定义的函数，this 指向的是 window

一般定义，用new调用执行，让this指向新建的空对象。

作为对象属性的函数，调用时，this指向调用函数的对象。

1. 通过call()和apply()来改变this的默认引用：

所有函数对象都有的call方法和apply方法，function().call(b)，this指向b。

apply通过数组的方式传递参数，call通过一个个的形参传递参数。

1. 一些函数特殊执行情况this的指向问题：

setTimeout()和setInverval():指向window

1. dom中触发事件的回调方法执行中活动对象里的this指向该dom对象。

## 数组方法

concat() 合并数组，不改变现有数组

join() 数组=>字符串

pop() 删最后一个

push() 加最后一个

shift() 删第一个

unshift() 加第一个

splice() 删/加（索引，删几个，加什么）

sort() 排序

valueOf() 原始值

toString() 数组=>字符串

reverse() 反转字符串

indexOf() 返回符合条件的索引值

reduce() 缩减求和

some() 有符合条件的

map() 遍历，不改变原有数组

forEach() 遍历

keys() 从数组创建一个包含数组键的可迭代对象

includes() 是否包含第一个指定的值

find() 符合条件的第一个数组元素

findIndex()符合条件的第一个数组元素的索引

filter() 筛选指定条件，创建新数组

every() 所有元素是否符合条件

## 面向对象

构造函数、原型模式、组合构造函数+原型模式

JS中所有的变量和对象我们都可以认为是直接或间接的new某个函数来创建的，被new的函数叫对象的‘构造函数’。函数的prototype属性叫对象的‘原型对象’，js规定原型对象所具有的属性和方法，new出来的对象也自动具有！

## Ajax相关

1. ajax是一种创建交互式网页应用的技术

流程：

创建XMLHttpRequest异步对象；

创建一个新的HTTP请求,并指定该HTTP请求的方法、URL及验证信息；

设置响应HTTP请求状态变化的函

数；

XMLHttpRequest异步对象对请求进行封装，然后发送给服务器；

服务器以流的方式把数据返回给浏览器；

XMLHttpRequest异步对象会不停的监听服务器的状态，获取异步调用返回的数据；

使用回调函数实现局部刷新。

1. GET/POST

* GET（“查询”）

使用queryString在url后面携带参数，一般有长度限制，提交到服务器的数据量小。

从服务器端获取数据。

可以复用前面的请求数据作为缓存结果返回。

* POST（“修改”）

可以传递queryString（application/x-www-form-urlencoded）、multipart/form-data的二进制报文（支持文件信息嵌入报文传输）、纯文本或二进制的body参数。提交到服务器的数据量大。

向服务器提交数据。

一般不会被缓存因素影响。

POST请求的报文不会被记录，更安全。

1. readyState

0： 请求未初始化

1： 服务器连接已建立

2： 请求已接收

3： 请求处理中

4： 请求已完成，且响应已就绪

1. status状态码

200 服务器响应正常

304 该资源在上次请求后没有任何修改（浏览器缓存机制）

400 无法找到请求的资源

401 访问资源的权限不够

403 没有权限访问资源

404 访问的资源不存在

405 需要访问的资源被禁止

407 访问的资源需要代理身份验证

414 请求的URL太长

500 服务器内部错误

1. 谈谈跨域

JSONP

CORS

## 列举ES6常用的新特性

比较常用的有let/const、解构/赋值、箭头函数、class类、Promise、async、Module模块等

## let/const

1. let 块级作用域
2. const 声明一个只读常量，一旦声明常量的值不可改变，声明的同时必须初始化赋值

const保证的是变量指向的那个内存地址不可改变，对于简单的数据（数值、字符串、布尔值），值就保存在那个内存地址，等同于常量，但对于复合数据类型（对象、数组），变量指向的内存地址，保存的是一个指针。const只能保证这个指针是固定的，至于内部数据不可控。

## 箭头函数解决了什么问题

箭头函数解决了this指向的问题 + 例子说明

## 谈谈Promise

简单说promise一个容器，存放着未来才会结束的事件，promise解决了回调地狱的问题，把嵌套的结构变成树形结构。

## Promise实现原理

## 深拷贝、浅拷贝

## 指针相关

## 谈谈webStorage以及cookie

（webStorage）sessionStorage、localStorage、cookie都是在浏览器端存储的本地数据。

* cookie：

单个域名存储量比较小（各个浏览器不同，大概4kb）

所有域名存储量有个数限制（各个浏览器不同，一般不超过50个）

cookie有潜在安全风险，可能被拦截篡改

* sessionStorage：

生命周期是在浏览器关闭之前，临时保存

存储量大

* localStorage：

生命周期是永久性的，存储在本地硬件设备

存储量比较大，大概5MB，且总量无限制

## 其他

java是强类型语言，javascript是弱类型语言。

面向对象的三大特性：封装、继承、多态。

# Vue

## 谈谈生命周期

## vue常用指令

v-for

v-if

v-show

v-bind

v-on

v-html

v-text

## v-if和v-show区别

## vue常用修饰符

事件修饰符 .stop .prevent .capture

按键修饰符 .enter .tab .delete .space .left .right .middle

.lazy .number .trim

## 双向数据绑定如何实现

## 组件消息传递方式

## MVVM思想

## 什么时候用VueX

## VueX解决了什么问题

## VueX优缺点

* vuex的优点

解决了非父子组件的消息传递（将数据存放在state中）

减少了AJAX请求次数，有些情景可以直接从内存中的state获取

* vuex的缺点

刷新浏览器，vuex中的state会重新变为初始状态

解决方案vuex-along ,vuex-persistedstate

https://github.com/robinvdvleuten/vuex-persistedstate

# 移动端

# 其他

## 谈谈http

## http/https区别