* 元素标签分类（块元素 行内元素 行内块元素）
  + 块元素：
    - div p ol ui li h1-h6 （from，table,tfoot,th,tr,td,thead，hr）
  + 行内元素：
    - a span textarea （i,em,br）
  + 行内块元素： input img video
  + 可变元素：可变元素为根据上下文语境决定该元素为块元素或者内联元素
    - button del iframe ins map object script

html5新增语义标签：footer、header，、video、nav、audio、canvas

css3新增特性：transition、animation、box-shadow、border-image、 color: rgba、**弹性布局**  [F](https://links.jianshu.com/go?to=https%3A%2F%2Flink.juejin.im%2F%3Ftarget%3Dhttp%253A%252F%252Fwww.ruanyifeng.com%252Fblog%252F2015%252F07%252Fflex-grammar.html)lex、**栅格布局** grid

* 盒模型分类
  + - 基准盒模型，怪异盒模型两种盒模型的区别
      * 宽度和高度的计算方式不同
      * 标准盒模型：
      * width = content-width
      * height = content-height
      * 怪异盒模型:
      * width = content-width + padding-width + border-width
      * height = content-height + padding-height + border-height
      * 如何在CSS 设置这两个模型
      * 标准盒模型：
      * box-sizing: content-box
      * 怪异盒模型：
      * box-sizing: border-box
* 定位类型
  + - 相对定位：position：relative
    - 绝对定位：position：absolute
    - 固定定位：position：fixed
    - 静态定位：position：static
* 如何解决浮动造成的内容塌陷 清除浮动
  + - overflow：hidden
    - after伪元素
    - float: none;
    - clear：both
* 页面常见布局
  + 静态布局，流式布局，弹性盒子，响应式布局
* 100vw与100%有啥区别
  + 答案：100vw是根据视口（可视区域）的100%，100%是根据父级元素的宽度设置的
* div水平垂直居中
* position：absolute；
  + - left：0；right：0；top：0；bottom：0；margin：0 auto；
* position: absolute;
  + - left:50%;
    - top:50%;
    - margin-left:-300px;
    - margin-top:-300px;
* position: absolute;
  + - left:50%;
    - top:50%;
    - transform: translate(-50%, -50%);
* background: red;
  + - position:absolute;
    - top:50%;
    - left:50%;
    - margin:-100px 0 0 -100px;
    - width:200px;
    - height:200px;
    - border:1px solid red;

弹性盒子实现：

父元素{

display: grid;

}

子元素 {

margin: auto;

}

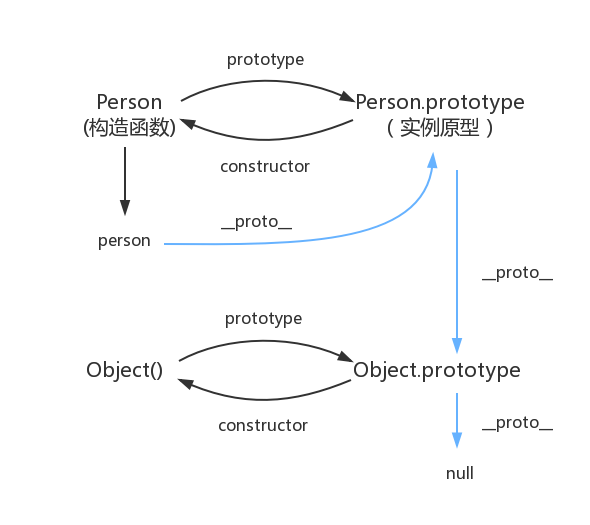
父元素 {

display: flex;

align-items: center;

justify-content: center;

}

* 左边固定右边自适应布局
  + (1).left{float:left} .right{overflow:hidden;zoom:1//兼容ie6}
  + (2).left{float:left} .right{magrin-left}
  + (3).left{position:flxed} .right{magrin-left}
  + (4).left{width:100px} .right{width:calc(100% - 100px)}
* promise
  + Promise 是异步编程的一种解决方案，其实是一个构造函数，自己身上有all、reject、resolve这几个方法，原型上有then、catch等方法。
  + 解决了回调地狱
* js动态添加类名有哪几种方式
  + - 元素.classList.add()
    - 元素.className()
* div水平居中有哪几种方式
  + - text-align:center; magrin:0 auto;
* js获取DOM的几种方式
  + - document.getElementsById()//返回指定id元素
    - document.getElementsByClassName()//返回指定class元素
    - document.getElementsByName()//返回指定name 属性元素
    - document.getElementsByTagName()//返回指定标签元素
    - document.document.querySelector()//返回文档中匹配指定 CSS 选择器的一个元素
* js数据类型（基本7种）
  + - String，字符串类型
    - number，数字类型
    - null，空
    - undefined，未定义
    - boolean，布尔类型（值为true，false）
    - object，对象类型
    - symbol，表示独一无二的值
* 原型 原型链
  + - Prototype：在JavaScript中，每个函数都有一个prototype属性，函数的prototype指向了一个对象，而这个对象正是调用构造函数时创建的实例的原型
    - 原型链示意图
    - 
    - \_proto\_：
      * function Person() {}
      * var person = new Person();
      * console.log(person.\_\_proto\_\_ === Person.prototype); // true
* 堆和栈的区别
  + 栈由系统自动分配，而堆是人为申请开辟；
  + 栈获得的空间较小，而堆获得的空间较大；
  + 栈由系统自动分配，速度较快，而堆一般速度比较慢；
  + 栈是连续的空间，而堆是不连续的空间。
    - 栈堆：存放数据的地方  
      基本数据类型：number,string,[boolean](https://so.csdn.net/so/search?q=boolean&spm=1001.2101.3001.7020),null,undefined.  
      引用数据类型(Object类)：有常规名值对的无序对象{a:1}，数组[1,2,3]，以及函数等。
* js深浅拷贝
  + 如何区分深拷贝与浅拷贝，简单点来说，就是假设B复制了A，当修改A时，如果B也跟着变了，说明这是浅拷贝，如果B没变，那就是深拷贝。
  + 赋值其实是将原始对象存储在栈中的指向堆内存的地址赋值给新对象，而不是堆中的数据。所以新对象的任何数据类型的属性值发生改变，都会影响到原始数据。
  + 浅拷贝：创建一个新对象，这个对象对原始对象的属性值有着一份精确的拷贝。如果属性是基本类型数据，拷贝的就是基本类型的值，改变原始数据属性，新对象属性不会发生改变；如果属性是引用类型，那么就是拷贝的这个属性的内存地址，如果属性值发生改变，就会影响到原始对象。
  + 简单来说：浅拷贝，拷贝基本类型属性，那么拷贝的是他的值，不会影响原数据，如果拷贝的属性是引用类型，那么就是拷贝的就是这个属性的地址，就会影响到原始对象。
  + 深拷贝：创建一个新对象，将原始对象从内存中完整的拷贝出来，从堆内存中开辟出一个新的区域存放新对象，且修改新对象，不会影响原始对象
    - 总结：**JSON 对象的 parse 和 stringify都是深拷贝**
      * 1：深拷贝只是从源数据中拷贝一份出来进行操作，而不是改变源数据；改变源数据的那是浅拷贝；
      * 2：原生js方法slice、concat都不是真正意义上的深拷贝，都仅只适用于一维数组，拷贝的属性不够彻底；
      * 3：实现js深拷贝我们可以通过JSON.parse(JSON.stringify(object)) //JSON.parse是转换obj为字符串，JSON.stringify是转换字符串为obj、递归以及JQuery库的extend方法来实现；
* 数组常用方法：
  + - push() 方法可向数组的末尾添加一个或多个元素，并返回新的长度。
    - pop() 方法用于删除并返回数组的最后一个元素。
    - shift() 方法用于把数组的第一个元素从其中删除，并返回第一个元素的值。
    - unshift() 方法可向数组的开头添加一个或更多元素，并返回新的长度。
    - sort() 方法用于对数组的元素进行排序。
    - reverse() 方法用于颠倒数组中元素的顺序。
    - slice（）
    - split（）
    - splice（）
* 字符串常用方法：
  + - indexOf（）返回指定字符创第一次出现的位置
    - lastindexOf（）返回指定字符串最后一次出现的位置
    - charAt（）返回指定索引的字符
    - substring（）提取字符串中两个指定索引号之间的字符（两个索引不能为负值）
    - slice（）提取字符串中两个指定索引号之间的字符（索引可以为负值，-1就是倒数第二位）
    - split（）把字符串分割为子字符串数组
    - toString（）转换为字符串格式
    - concat（）链接两个字符串
* 闭包
  + 概念：一个函数引用另一个函数的声明。一个函数和对其周围状态（**lexical environment，词法环境**）的引用捆绑在一起（或者说函数被引用包围），这样的组合就是**闭包**（**closure**）。
  + 也就是说，闭包让你可以在一个内层函数中访问到其外层函数的作用域。在 JavaScript 中，每当创建一个函数，闭包就会在函数创建的同时被创建出来。
  + 在一个外函数（函数a）中定义了一个内函数（函数b），内函数里运用了外函数（a）的临时变量（TV），并且外函数的返回值（return b）是内函数的引用。这样就构成了一个闭包。

function makeFunc() {

var name = "Mozilla";

function displayName() {

alert(name);

}

return displayName;//返回内层函数引用，

}

var myFunc = makeFunc();

myFunc();

//闭包

* Vue 路由 mode模式：
  + Mode：“Hash” hash模式灵活运用了html的瞄点功能、改变#后的路径本质上是更换了当前页面的瞄点，所以不会刷新页面。
  + Mode：“history”是使用了 H5 提供的pushState() 和 replaceState()，允许开发者直接更改前端路由，即更新浏览器 URL 地址而不重新发起请求(将url替换并且不刷新页面)。
* V-if：条件符合元素才会出现
* V-show：设置的元素都会出现或消失
* v-if是通过控制dom节点的存在与否来控制元素的显隐；
* v-show是通过设置DOM元素的display样式，block为显示，none为隐藏；
* **组件中 data 为什么是一个函数？**
  + 为什么组件中的 data 必须是一个函数，然后 return 一个对象，而 new Vue 实例里，data 可以直接是一个对象？
  + 因为组件是用来复用的，且 JS 里对象是引用关系，如果组件中 data 是一个对象，那么这样作用域没有隔离，子组件中的 data 属性值会相互影响，如果组件中 data 选项是一个函数，那么每个实例可以维护一份被返回对象的独立的拷贝，组件实例之间的 data 属性值不会互相影响；而 new Vue 的实例，是不会被复用的，因此不存在引用对象的问题。

## ****v-model 的原理？****

* + 我们在 vue 项目中主要使用 v-model 指令在表单 input、textarea、select 等元素上创建双向数据绑定，我们知道 v-model 本质上不过是语法糖，v-model 在内部为不同的输入元素使用不同的属性并抛出不同的事件：
    - text 和 textarea 元素使用 value 属性和 input 事件；
    - checkbox 和 radio 使用 checked 属性和 change 事件；
    - select 字段将 value 作为 prop 并将 change 作为事件。
* v-model的原理简单描述  
  v-model主要提供了两个功能，view层输入值影响data的属性值，data属性值发生改变会更新view层的数值变化。  
  其核心就是，一方面modal层通过defineProperty来劫持每个属性，一旦监听到变化通过相关的页面元素更新。另一方面通过编译模板文件，为控件的v-model绑定input事件，从而页面输入能实时更新相关data属性值。  
  v-model是什么  
  v-model就是vue的双向绑定的指令，能将页面上控件输入的值同步更新到相关绑定的data属性，也会在更新data绑定属性时候，更新页面上输入控件的值。
* **为什么使用v-model**v-model作为双向绑定指令也是vue两大核心功能之一，使用非常方便，提高前端开发效率。在view层，model层相互需要数据交互，即可使用v-model。webpack打包原理：是根据文件间的依赖关系对其进行静态分析，将这些模块按指定规则生成静态资源，当 webpack处理程序时，它会递归地构建一个依赖关系图，其中包含应用程序需要的每个模块，将所有这些模块打包成一个或多个bundle。
* **钩子函数：**
* 生命周期 描述
  + beforeCreate 组件实例被创建之初，组件的属性生效之前
  + created 组件实例已经完全创建，属性也绑定，但真实 dom 还没有生成，$el 还不可用
  + beforeMount 在挂载开始之前被调用：相关的 render 函数首次被调用
  + mounted el 被新创建的 vm.$el 替换，并挂载到实例上去之后调用该钩子
  + beforeUpdate 组件数据更新之前调用，发生在虚拟 DOM 打补丁之前
  + update 组件数据更新之后
  + activited keep-alive 专属，组件被激活时调用
  + deactivated keep-alive 专属，组件被销毁时调用
  + beforeDestory 组件销毁前调用
  + destoryed 组件销毁后调用

js创建添加节点的方法：

http://www.360doc.com/content/19/0410/21/5923560\_827793576.shtml