1.flex-grow、flex-shrink、flex-basis详解

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| flex-grow | 用来设置当父元素的宽度大于所有子元素的宽度的和时（即父元素会有剩余空间），子元素如何分配父元素的剩余空间。 | | | | 默认值0 |
| flex-shrink | 当**..**小于..（即子元素会超出父元素），子元素如何缩小自己的宽度的。 | | | | 默认值1 |
| flex-basis | 该属性用来设置元素的宽度，其实，width也可以设置宽度。如果元素上同时设置了width和flex-basis，那么width 的值就会被flex-basis覆盖掉。 | | | | |
| flex | | 等效于 | flex-grow | flex-shrink | flex-basis |
| none | | 0 | 0 | auto |
| auto | | 1 | 1 |
| >>0(eg:2) | | 2 | 1 | 0% |
| 3 4(两个非负数) | | 3 | 4 |
| 5 (一个非负数) 12px/10% | | 5 | 1 | 12px/10% |

1. vue单页面和多页面优缺点  
   单页面应用（spa）  
   概念：只有一个html页面，所有跳转方式都是通过组件切换完成的。  
   优点：页面之间跳转流畅、组件化开发、组件可复用、开发便捷、易维护。  
   缺点：首屏加载较慢，加载整个项目中使用的css、js，SEO优化不好。  
   多页面应用（mpa）  
   概念：整个项目有多个html，所有跳转方式都是页面之间相互跳转的。  
   优点：首屏加载较快，只加载本页所使用的的css、js，SEO优化较好。  
   缺点：页面跳转较慢。
2. vue中操作DOM的方法

A.this.$refs.\* 拿到虚拟Dom,可以进行真实dom的一切操作

1）在**标签**中添加ref="\*" this.$refs.box1.style.background="black";

2）在**方法**中用**this.$refs.\***拿到这个元素，跟document.getElementById('\*')一样。

B.自定义**指令**中，el就是当前环境下的Dom元素

C.vue中有两种方法获取到当前元素

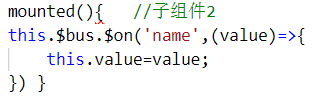
var el = event.target; //当前元素，可修改（能够用此方法获取到他的子元素，不能获取他本身的内容）

var el = event.currentTarget;//当前元素，不可修改（能够用此方法获取到他的子元素及能获取他本身内容）

4.父组件调用子组件方法？

通过ref属性，给**子组件**绑定ref属性，**父组件通过this$refs.ref**的绑定值.子组件的方法

5.vue兄弟组件之间传值的方法

1)通过事件总线bus传值 再main.js里首先声明 Vue.prototype.**$bus**=new Vue();

2)通过父组件在中间做中间件，相互传值（子组件1传值给父组件$emit，父组件传值给子组件props)

6.json数据遍历详解

1)json数据获取方式: json对象.键名 json对象[“键名”] 数组对象[索引]

2)遍历 A.双重for循环



B.JQuery遍历 $.each(index,item,callback)

7.Var,let,const的区别？ 提到这个就先说一下作用域吧（三个作用域说一遍）

(1)var.let**声明**变量，const声明常量，且不能**修改**；(2)let和const声明的数据不能再**重复声明**

(3)Var存在**预解析**，可以先使用在声明，但是let和const必须先声明再使用

(5)实现原理：

Var会直接在栈内存中预分配空间，在实际执行时再存储相应的变量，如果是引用类型，会在堆内存中开辟一个存储空间存放实际内容，而在栈内存中存放指向其堆内存的指针

let不会预分配空间，会在栈内存分配变量时，做一个检查，如果有相同变量名存在，就会报错

Const也不会与分配空间，也会做检查，且const定义的变量不能更改，对于基本类型来说，不能修改它的值，对于引用类型来说，不能修改栈内存中分配的指针，但是可以修改指针指向的对象中的属性  
8.箭头函数

9.闭包:在一个函数内部创建另一个函数，通过另一个函数访问这个函数的局部变量（参数和变量不会被垃圾回收机制所收回） [是指有权访问另外一个函数作用域中的变量的函数]

2．闭包的优缺点，什么时候使用闭包？

闭包会使得函数中的变量都被保存在内存中，内存消耗很大，所以不能滥用闭包，否则会造成网页的性能问题，在

IE中可能导致内存泄露。过度使用闭包会导致性能下降，尽量少使用闭包，只在必要时使用。

3．怎么解决闭包的变量长驻内存

解决方法：在退出函数时，将不使用的变量全部删除

4．什么使用使用闭包

闭包可以将一些不希望暴露在全局的变量封装成“私有变量”。