1：什么是事件委托，事件委托优点是什么?

> 利用了事件冒泡原理，把事件加到父级上，触发执行效果

优点：

```
1:减少事件注册，节省内存
2:简化了dom节点更新时，相应事件的更新
```

缺点：

```
1:事件委托基于冒泡，对于不冒泡的事件不支持。
2:层级过多，冒泡过程中，可能会被某层阻止掉。
```

2：谈谈对this对象的理解?

1. 在一般函数方法中使用 this 指代全局对象
2. 作为对象方法调用，this 指代上级对象
3. 作为构造函数调用，this 指代new 出的对象
4. apply 调用 ，apply方法作用是改变函数的调用对象，此方法的第一个参数为改变后调用这个函数的对象，this指代第一个参数
5. 在事件中，this指向触发这个事件的对象，特殊的是，IE中的attachEvent中的this总是指向全局对象Window；

3：new操作符具体干了什么呢?

1. 创建一个空对象，并且 this 变量引用该对象，同时还继承了该函数的原型。
2. 属性和方法被加入到 this 引用的对象中。

3. 新创建的对象由 this 所引用，并且最后隐式的返回 this 。

4：eval是做什么的?

> 它的功能是把对应的字符串解析成JS代码并运行； 应该避免使用eval，不安全，非常耗性能（2次，一次解析成js语句，一次执行）。 由JSON字符串转换为JSON对象的时候可以用eval，var obj =eval('('+ str +')')。

5：window.onload 和 jquery的 $(document).ready(function)区别

1. window.onload 必须等到页面内包括图片的所有元素加载完毕后才能执行

2. document.ready是DOM结构绘制完毕后就执行，不必等到加载完毕。

6:什么是作用域？有几种作用域?

> 函数和变量的作用范围

- 全局作用域
- 函数作用域
- eval作用域

7:显式转换和隐式转换

1. 显式转换

> 所谓的显式转换就是人为的方式对数据进行类型转换。

```
1.转换为数值类型:Number(mix)、parseInt(string,radix)、parseFloat(string)。
2.转换为字符串类型:toString(radix)、String(mix)。
3.Boolean(mix)。
```

2. 隐式转换

> JavaScript引擎在运算之前会悄悄的把他们进行了隐式类型转换的

```
//数字和布尔
3 + true;   // 4

//字符串和数字
var x = '5';
console.log(x + 1)   '51'

//判断语句中的判断条件需要是Boolean类型，所以条件表达式会被隐式转换为Boolean。
var obj = {};
if(obj){
    while(obj);
}
```

8：值传递和引用传递的区别?

(1)值传递：声明的是原始的类型，具体的值，传给变量后，就与它本身没有关系了
(2)引用传递：把地址传给变量，传完后变量的值会根据变量的变化而变化