1. 数据类型：可分为原始类型和对象类型。数字、字符串、布尔值、null、undefined和对象共6种。也可分为可变类型和不可变类型，原始类型皆不可变，对象类型均可变。
2. JS可自由地进行数据类型转换。例如，在程序希望使用数字的地方使用了字符串，那么JS会自动地将字符串转换为数字。
3. JS变量是无类型的。变量可以被赋予任何类型的值，同样一个变量完全可以重新被赋予不同类型的值。
4. 数字。JS中数字上溢、下溢、被0整除都不会报错。  
   上溢会返回Infinity或-Infinity。  
   下溢会返回0。  
   被0整除会返回Infinity或-Infinity。  
   0除以0是没有意义的，无穷大除以无穷大、负数开平方、或者算数运算符与不是数字或者无法转换为数字的操作数一起运算时会返回NaN。  
   Infinity和NaN是全局变量，且typeof返回值是“number”
5. 字符串  
   JS中没有字符一说，只有字符串
6. 布尔值  
   任意的数据类型都可以转换为布尔值。无论什么样的对象转换为布尔值一定为true，只有原始类型中的几种情况才会被转换为false：undefined、null、0、-0、“”、NaN共6种。  
   在if判断语句中，什么样的语句最终都会被转换为布尔值检查。
7. Null和undefined  
   null是一个特殊的对象，typeof返回的是“object”，它是object.prototype的指向。  
   undefined主要是表示值没有初始化，typeof返回的是“undefined”
8. Typeof只会返回小写的字符串，共6种情况，有“number”、“string”、“boolean”、“undefined”、“object”、“function”。
9. 包装对象  
   原始类型是绝对没有方法的，存取字符串、数字和布尔值的属性时会创建一个临时对象，这个临时对象JS称为包装对象。用过即丢。
10. 值的比较  
    原始值的比较，它们的值相等就相等。对象的比较是所指向的是一个同一个对象才相等。即使这两个对象在字面上完全相等，但是他们不是一个对象，所以比较起来并不相等。
11. 类型转换  
    原始值向对象的转换默认使用String（）、Number（）、Boolean（）。
12. 布尔值向其它的转换
13. 数字、字符串之间的转换  
    字符串→数字：  
    Number（）传入的字符串只能以十进制进行转换，并且不能有非法字符。  
    全局函数parseInt（）会跳过类型0x或0X的字符，默认解析为十进制。但如果有非法字符，返回NaN。第二个参数为转换后数字的进制。  
    toFixed（）十进制，参数为指定的小数点位数。  
    toExponential（）十进制，指数类型，参数指定小数点后的数量。  
    toPrecision（）十进制，有效数字，参数指定有效位数，如果参数小于整数部分会用指数形式，如果大于整数部分会使用小数点补0.  
    数字→字符串：  
    Number对象定义了toString的方法，参数表示转换为字符串时的基数，例如  
    var n=17；   
    n.toString(2) “100001”  
    n.toString(8) “21”  
    n.toString(16) “11”这其中会有包装对象
14. 对象转换为原始值  
    对象→字符串：  
    所有的对象都从Object.prototype上继承了一个toString（）方法，返回”[object Object]”。大多数情况下，这个没什么用。但是有些类改写了这个方法（原型链），例如数组的toSting（）使用join（“”）方法，函数的toString（）返回函数的定义字符串，日期返回类似“Sat Jul 21 2018 17:00:12 GMT+0800 (中国标准时间)”的字符串，正则表达式返回所定义的字符串。  
    对象→数字  
    大多数对象会调用valueOf（）方法，默认会返回对象本身（仍然是个对象），大部分没什么用。但是有些有用，比如日期的返回到1980年0点0分的毫秒数。
15. 对象到字符串的转换过程  
    1.查看toString（）方法，如果是原始值就使用  
    2.如果没有定义toString（）方法，或者该对象的toSring（）方法不返回一个原始值，那么会调用valueOf（）方法。如果返回原始值，则将这个原始值转换为字符串。  
    3.如果上述都没有。则抛出类型错误。
16. 对象到数字的转换过程  
    1.查看valueOf（）方法，如果返回原始值，则使用  
    2.如果valueOf（）方法没有或者不返回原始值，则查看toString（）方法  
    3.如果都没有，抛出类型出错误异常
17. []转换到字符串为””,转换到数字的时候，由于数字的valueOf（）方法返回的是数组本身，不是一个原始值，所以此时查看valueOf（）方法，为“”,””用Number（）转换为数字为0。[‘a’]转换为字符串调用join（）返回’a’,转换为数字时再调用Number（）返回NaN。
18. ES5中是没有块级作用域的，只有全局作用域和函数作用域。作用域链实际上是为了查找变量的作用域，一个变量既然声明了，你总要知道它在哪申明，作用域是哪吧！  
    变量会声明提前，但是定义还是在那个位置，当你在前面使用该变量会是undefined。如果使用未声明的变量会抛出一个引用错误。
19. 定义一个全局变量，实际上就是给全局对象定义了一个属性。但是这个属性却是不可配置的，不能用delete删除。
20. 作用域链（scope chain）是一个对象的列表或来链表，定义一个函数其实就是保存一个作用域链。