# 变量、数据类型、对象

## 0.1 + 0.2 === 0.3 嘛？为什么？怎么解决？

精度丢失可能出现在进制转换和对阶运算中

JavaScript 使用 Number 类型来表示数字（整数或浮点数），遵循 IEEE 754 标准，通过 64 位来表示一个数字（1 + 11 + 52）

1 符号位，0 表示正数，1 表示负数 s 11 指数位（e） 52 尾数，小数部分（即有效数字）

最大安全数字：Number.MAX\_SAFE\_INTEGER = Math.pow(2, 53) - 1，转换成整数就是 16 位，所以 0.1 === 0.1，是因为通过 toPrecision(16) 去有效位之后，两者是相等的。 在两数相加时，会先转换成二进制，0.1 和 0.2 转换成二进制的时候尾数会发生无限循环，然后进行对阶运算，JS 引擎对二进制进行截断，所以造成精度丢失。

## JS 整数是怎么表示的？

通过 Number 类型来表示，遵循 IEEE754 标准，通过 64 位来表示一个数字，（1 + 11 + 52），最大安全数字是 Math.pow(2, 53) - 1，对于 16 位十进制。（符号位 + 指数位 + 小数部分有效位）

## Number() 的存储空间是多大？如果后台发送了一个超过最大限制的数字怎么办?

Math.pow(2, 53) ，53 为有效数字，会发生截断，等于 JS 能支持的最大数字。

## 判断数据类型的方式有哪些

1. typeof  
2. instanceof  
3. constructor  
4. Object.prototype.toString.call()

## new 一个函数发生了什么

创造一个全新的对象 这个对象会被执行 [[Prototype]] 连接，将这个新对象的 [[Prototype]] 链接到这个构造函数.prototype 所指向的对象 这个新对象会绑定到函数调用的 this 如果函数没有返回其他对象，那么 new 表达式中的函数调用会自动返回这个新对象

## new 一个构造函数，如果函数返回 return {} 、 return null ， return 1 ， return true 会发生什么情况？

如果函数返回一个对象，那么new 这个函数调用返回这个函数的返回对象，否则返回 new 创建的新对象

## symbol 有什么用处？

可以用来表示一个独一无二的变量防止命名冲突。但是面试官问还有吗？我没想出其他的用处就直接答我不知道了，还可以利用 symbol 不会被常规的方法（除了 Object.getOwnPropertySymbols 外）遍历到，所以可以用来模拟私有变量。 主要用来提供遍历接口，布置了 symbol.iterator 的对象才可以使用 for···of 循环，可以统一处理数据结构。调用之后回返回一个遍历器对象，包含有一个 next 方法，使用 next 方法后有两个返回值 value 和 done 分别表示函数当前执行位置的值和是否遍历完毕。 Symbol.for() 可以在全局访问 symbol。

## NaN 是什么，用 typeof 会输出什么？

Not a Number，表示非数字，typeof NaN === 'number'

## JS 隐式转换，显示转换?

一般非基础类型进行转换时会先调用 valueOf，如果 valueOf 无法返回基本类型值，就会调用 toString

* 字符串和数字
* "+" 操作符，如果有一个为字符串，那么都转化到字符串然后执行字符串拼接
* "-" 操作符，转换为数字，相减 (-a, a \* 1 a/1) 都能进行隐式强制类型转换
* [] + {} 和 {} + []
* 复制代码
* 布尔值到数字
* 1 + true = 2
* 1 + false = 1
* 转换为布尔值
* for 中第二个
* while
* if
* 三元表达式
* || （逻辑或） && （逻辑与）左边的操作数
* 符号
* 不能被转换为数字
* 能被转换为布尔值（都是 true）
* 可以被转换成字符串 "Symbol(cool)"
* 宽松相等和严格相等

宽松相等允许进行强制类型转换，而严格相等不允许

* 字符串与数字

转换为数字然后比较

-其他类型与布尔类型

* 先把布尔类型转换为数字，然后继续进行比较

-对象与非对象

* 执行对象的 ToPrimitive(对象）然后继续进行比较

-假值列表

* undefined
* null
* false
* +0, -0, NaN
* ""

## 如何判断一个对象是不是空对象？

Object.keys(obj).length === 0

## 如何判断数组类型？

Array.isArray

## 函数中的arguments是数组吗？类数组转数组的方法了解一下？

是类数组，是属于鸭子类型的范畴，长得像数组，

* ... 运算符
* Array.from
* Array.prototype.slice.apply(arguments)

## 什么叫变量提升？

函数在运行的时候，会首先创建执行上下文，然后将执行上下文入栈，然后当此执行上下文处于栈顶时，开始运行执行上下文。

在创建执行上下文的过程中会做三件事：创建变量对象，创建作用域链，确定 this 指向，其中创建变量对象的过程中，首先会为 arguments 创建一个属性，值为 arguments，然后会扫码 function 函数声明，创建一个同名属性，值为函数的引用，接着会扫码 var 变量声明，创建一个同名属性，值为 undefined，这就是变量提升。

## instance 如何使用？

左边可以是任意值，右边只能是函数

'hello tuture' instanceof String // false

## 如果一个构造函数，bind了一个对象，用这个构造函数创建出的实例会继承这个对象的属性吗？为什么？

不会继承，因为根据 this 绑定四大规则，new 绑定的优先级高于 bind 显示绑定，通过 new 进行构造函数调用时，会创建一个新对象，这个新对象会代替 bind 的对象绑定，作为此函数的 this，并且在此函数没有返回对象的情况下，返回这个新建的对象