正课:

ES6

块作用域

箭头函数

参数增强

模板字符串

结构

OOP

for of

promise

1. 块作用域:

let: 声明一个不会被提前的变量

为什么: 传统js中声明提前破坏了程序正常的执行顺序

何时: 所有的var，都要用let代替

如何: 代替var

原理: let a之前不允许提前使用a，也不允许重复声明a

let为js添加了块级作用域:

什么是块级作用域: 代码中任何一对{}之间的语句称为一个代码块，代码块中的变量，只能在块内使用

比如: if(){...}else{...} for(){...} while(){...}

问题: 没有块级作用域，块内的变量即使不执行，也会被提前到块外，污染块外部！

为什么: 为了避免块内的变量提前到块外，污染外部的变量

何时: 只要希望代码块内的变量不要影响外部的变量时

原理: 自动使用匿名函数自调方式，包括块内的代码

2. 箭头函数: 简化回调函数

何时: 今后所有回调函数都可用箭头函数简化

如何:

1. 去function加=>

2. 更简化: 如果只有一个参数，可省略()

3. 更简化: 如果回调函数中只有一句话，可省略{}

强调: 如果只有一句话，则结尾;不要！

更简化: 如果仅有一句return可继续省略return

特点: 箭头内外共用同一个this

特例: 如果希望内外this不通用时，就不能用箭头函数

3. 参数增强: 3种:

默认值default: ES6中可对参数指定默认值。

调用时，如果没有传入参数值，则默认使用提前指定的默认值

比如: Array.prototype.indexOf=function(elem,fromi=0){

//fromi=fromi||0;

}

强调: 有默认值的参数，必须放在参数列表的最后

剩余rest: 当传入函数的参数个数不确定时，可用...接住剩余的一切参数——代替arguments

何时: 只要参数个数不确定

为什么: arguments的3大问题:

1. 总是获得全部参数，不能有选择的获得部分

2. 是类数组对象，不是纯正的数组，不能使用数组API

3. 不能改名

如何: 在函数定义时的参数列表中: (其它固定参数, ...数组名)

优: 1. 可有选择的获得部分参数

2. 是纯正的数组

3. 可自定义名称

总结: 今后都用rest代替arguments

散播spread: 在函数调用时，用...打散数组类型的参数为单个元素，再传入函数——代替apply

何时: 如果函数本身需要多个参数单独传入，而需要传入的确是数组时

如何: 在调用时，传入参数时用"...数组"打散数组参数，单个元素分别传入

4. 模板字符串:

什么是: 支持内置表达式的字符串拼接

——简化了字符串拼接

何时: 今后，只要拼接的字符串中包含要计算的表达式，就用模板字符串简化

如何:

1. 模板字符串必须用` `包裹

2. 如果包含需要动态计算的表达式，则用${}包裹

3. 反引号内默认支持换行