正课:

1. 解构

2. for of

3. class

4. \*\*\*Promise

1. 解构:

什么是: 将一个对象或数组中的成员，分别赋值给多个单独的变量——简化批量赋值

何时: 只要将一个对象或数组中的成员，分别赋值给多个单独变量时

如何: 2种:

1. 数组解构:

var arr=[1,2,3];

var [a,b,c]=arr;

console.log(a,b,c);//1 2 3

强调: 数组解构靠下标对应

2. 对象解构:

var date={y:2017,m:9,d:21};

var {y:year,m:month,d:date}=date;

console.log(year,month,date);

强调: 对象解构靠属性名对应

函数调用时结构:

定义函数时: function fun({y:y,m:m,d:d}){...}

调用时: var date={y:2017,m:9,d:21};

fun(date)

2. for of: 简化版的for循环

如何: for(var elem of arr)

elem可直接获得arr中每个元素值

何时: 代替for循环，遍历下标为数字的数组或类数组对象

强调: 不能遍历关联数组和对象，因为下标不是数字

缺点:

1. 无法获得下标——要用下标时，不能用for of

2. 只能遍历所有，不能有选择的遍历

3. 只能从前向后

3. class类型:

什么是: 封装一类对象统一结构和API的 程序结构

——简化js中的面向对象

为什么: 传统的js类型定义方式不符合封装的特点

如何:

简化类型定义——封装:

1. 用class 类型名{}包括之前的构造函数和原型对象方法

2. 构造函数名升级为class名, 构造函数更名为constructor关键词

3. 原型对象方法不用写类型.prototype和function

直接写在class中的函数，自动添加prototype中

简化继承:

1. class 子类型 extends 父类型

类似于:

Object.setPrototypeOf(

子类型.prototype, 父类型.prototype

);

2. 借用父类型构造函数: super(参数...)

不用call，不用this！

其中: super特指父类型构造函数,且自动用当前正确this替换父类型构造函数中的this

super.父类型原型方法()

静态方法:

什么是: 不需要实例化对象，就可直接访问的成员

何时: 只要一个方法，不需要实例化也能使用时

原理: 相当于直接定义在构造函数对象上的方法

如何: static 方法(){...}