

21.4.9学习记录

```
//修改box样式
// box.style.width = "200px";
//通过style属性来修改元素的样式，
//每修改一个样式，浏览器就需要重新渲染一次页面，性能较差
//通过js控制class，可以更好使行为和样式分离
box.className += " b2";
```

一、JSON

--js中的对象只有js自己认识，其他语言都互不认识

--json就是一个特殊格式的字符串，这个字符串可以被任意的语言所识别，并且可以转换为任意语言中的对象，JSON在开发中主要用来数据的传递交互

JavaScript Object Notation JS对象表示法

JSON和JS对象的格式一样，**只不过JSON字符串中属性名必须加双引号**

--分类：

--对象：{ }

--数组：[]

--JSON允许的值：字符串、数值、布尔值、空值、普通对象、数组

--工具类JSON，这个对象可以将一个JSON和JS对象之间互相转换

--JSON.parse ()：JSON字符串转JS对象

--JSON.stringify ()：JS对象转JSON字符串

--但是JSON对象在IE7以下不支持

--eval () 执行一段字符串形式的JS代码，并将执行结果返回

使用此函数执行的字符串中含有大括号，它会将这个大括号当成是代码块；如果不希望这样，则需要在字符串前后各加一个小括号；

注意：这个函数功能很强大，可以直接执行一个字符串中的js代码，但是开发中尽量不要使用，首先它的执行性能比较差，还有安全隐患

**兼容IE7以下的JSON操作，可以引入一个外部js文件来处理
json2.js**

二、ES6 --ECMA

--TC39标准委员会：每年六月份发布新版本

--stage0（展示阶段）、1（征求意见阶段）、2（草案阶段）、3（候选阶段）、4（定案阶段）

--ES6环境：webpack3.x traceur 谷歌支持高

1、关于声明变量

--之前：var a：变量声明提前；if和for循环里声明的当全局用；

作用域：全局和函数作用域

用var定义的全局变量是属于window的，let、const不同

--现在：块级作用域就是大括号

--**let**：相当于之前的var

--不存在声明提前即预解析，必须先定义再使用

--同一个作用域中不能重复定义，但是for循环里面是父级作用域

--**const**：常量，定义时必须赋值，且不能改变

--const特性和let一样

--定义数组后，数组方法可以实现

--使用Object.freeze方法数组也没法修改了

--立即执行函数（function（）{}）（）

2、解构赋值：非常重要

```
--let [a, b, c] = [12, 5, 6];
```

--左右两边，解构和格式要保持一致

```
let { name, age, gender } = json;
```

```
let { name, age, gender: a } = json; 可以改名字用冒号
```

```
let [a, b, c = '暂无数据'] = ['aa', 'bb']; 又可以设置默认值，传null过来会有值
```

3、字符串模板 ``

--可以随意换行

--字符串拼接用`\${}`，很方便扔进去就行了，而且添加元素也是直接扔变量

--字符串查找

--indexOf (要找的东西) 返回索引，未找到返回-1

--includes (要找的东西) 返回true

--字符串以谁开头/结尾

--str.startsWith(要检测的东西);

--str.endsWith(要检测的东西);

--重复字符串

str.repeat();

--填充字符串

str.padStart(整个字符串的长度, 填充的东西);

str.padEnd(整个字符串的长度, 填充的东西);

4、函数变化

--函数默认参数

--函数参数默认已经定义了，不能再使用get, const声明

--扩展运算符或者叫rest运算符：（三个点）...对数组可开可收

--箭头函数

-- () => return 东西

-- () => {语句 return}

--注意:

①this问题, 定义函数所在的对象, 不再是运行时所在的对象

②箭头函数里面没有arguments, 用三个点...

③箭头函数不能当构造函数

三、数组

ES5新增的一些东西

--arr.forEach() 代替普通for循环

--arr.map() 很有用, 做数据交互即映射

--正常情况下需要配合return, 返回一个新的数组

--若是没有return就相当于forEach ();

--重新整理数据结构: [{title:'aaa'}] -> [{t:'aaaa'}]

--arr.filter()

--过滤一些不符合条件的, 返回新数组

--arr.some()

--类似查找, 某一个符合就返回true

--arr.every()

--类似查找, 全部符合就返回true

其实它们可以接收两个参数 (回调函数, this指向谁);

--arr.reduce((prev,cur,index,arr) => {})

--从左往右

--例如求数组的和或者阶乘

--幂运算符 $2^{**3} = 8$

--arr.reduceRight()

--从右往左

--for。。 of循环

--arr.keys ()

--arr.entries ()