JS复习

1.免费课基础知识

- 1.页面组成部分
- 2.js的组成部分(BOM,DOM,ECMAScript)
- 3.js引入方式
- 4.js输出方式
    - alert(字符串,默认调用toString转为字符串)
    - console.log/dir
    - confirm(确定true,取消false)
    - prompt(有输入框,返回值是你输入的内容)

- 5.变量
    - var/function
    - let/const
    - 命名规范(驼峰命名法;数字大小写字母下划线$但是数字不能作为开头)

数据类型

1.基本数据类型和引用数据类型的区别

number

- Number():返回值==>数字/NaN

```
Number("");//0
Number(null);//0
Number(undefined);//NaN
Number("1px");//NaN
```

- isNaN():返回值==>true/false,默认使用的是Number方法转换的
- parseInt()

```
parseInt("");//NaN
parseInt(null);//NaN
parseInt("1px");//1
```

- parseFloat()

boolean

- Boolean()

```
0,"",NaN,null,undefined是false
```

- ! 先转布尔再取反

```
![],!NaN
```

- !! 转布尔

string

- 单双引号的问题

```
var str="I'm a person"
```

- 转义\

```
var str='I\'m a person';
```

- 有length(只读的,不可以修改),有索引
- split(指定字符/正则)
- replace(指定字符/正则)
- slice(n,m)
- subStr(n,m)
- subString(n,m)

- indexOf(字符,index开始查找的索引)
- lastIndexOf(字符,index开始查找的索引)
- includes(字符,index开始查找的索引)
- toUpperCase()
- toLowerCase()
- match(正则)
- search(正则)
- concat()拼接
- repeat(num)
- startsWith()
- endsWith()
- padStart(length,str)
- padEnd(length,str);

```
解析url问题:
var url="https://www.baidu.com/s?wd=马克飞象&rsv_spt=1&rsv
_iqid=0xa1571c970005724e&issp=1&f=8&rsv_bp=1&rsv_idx=2";
var s=url.split("?")[1];
s=s.replace(/=/g,":'").replace(/&/g,"',");
eval("({"+s+"'})");
回去看一下之前写的几个方法:
WEEK2  DAY4  1.正则复习题.html
```

null,undefined

- 区别
- 什么情况会出现 undefined
- 这俩没有toString方法
- null==undefined,跟其他的都不相等
- var timer=null;在定义变量的时候暂时不赋值 ,一般会给他一个null
- 清空一个对象,obj=null
- 移除DOM0级事件.box.onclick=null;

array

- 1.length(可以修改的)
- 2.join(“”)

```
[1,2,3].join("");//"1,2,3"
[1,2,3].toString();//"1,2,3"
```

- 3.slice()
- 4.splice()
- 5.pop()
- 6.shift()
- 7.push()
- 8.unshift()
- 9.concat()
- 10.indexOf()/lastIndexOf()
- 11.includes()
- 12.find()/findIndex()
- 13.filter()
- 14.forEach()/map()
- 15.sort()
- 16.fill()
- 17.reverse()
- 18.copyWithIn()
- 扩展运算符 …

```
[...ary1,...ary2];
var ary=[1,2,3,4]
法一
eval("Math.max("+ary+")");
法二
Math.Max.apply(null,ary);
法三
Math.Max(...ary)
```

- 将类数组转为数组

```
法一
function toArray(likeAry){
   var ary=[];
   for(var i=0;i<likeAry.length;i++){
      ary.push(likeAry[i])
   }
   return ary
}
法二
function toArray(likeAry){
   return [].slice.call(likeAry);
 }
法三
function toArray(likeAry){
   return [...likeAry];
}
法四
function toArray(likeAry){
   return Array.from(likeAry);
 }
```

- 数组去重
- 数组方法重写,有回调函数的方法(注意第二个参数是用来改变函数的this的)

对象

- 键值对组合(属性名只能是字符串,属性名不可以重复)
- 获取属性值:obj.属性名,obj[“属性名”]
- 属性名可以实现变量的拼接,用[]包起来
- object.tostring方法 “[object Object]”
- for in
    - 1.先遍历数字按照从小到大的顺序遍历
    - 2.既可以遍历私有属性也可以遍历公有的属性(你加上的,自带遍历不出来)

- Object.is
- Object.assign()

函数function

- 1.定义+执行
- 2.参数问题
    - 函数的length
    - 默认值问题
    - 实参:arguments
- 3.arguments
    - 类数组
    - length 实参的个数
    - callee 函数本身
- 4.return
    - 返回值
    - return后面的代码不执行
- 5.函数的三种身份
    - 普通函数
    - new 类 ,prototype
    - Function类的实例,**proto**
- 6.this问题
- 7.箭头函数

date时间

- 1.new Date()
- getTime()
- getFullYear()
- getMonth() 0-11
- getDay() 0-6
- getDate()
- ……

倒计时
当前时间(钟表)

正则

检测数据类型的方法

- typeOf
- instanceof
- construct
- Object.toString.call()

Math函数

- 求最值
- 绝对值
- 算数平方根
- 向上/下取整
- 随机数
- 四舍五入

验证码

定时器

- 1.有个返回值,是个数字表示他是第几个定时器
- 2.定时器是异步的
- 3.setInterval(function(){},时间,给前面的函数传参数用的)

逻辑运算符

- 1.||和&&
- 2.i++,++i,i–,,–i

循环

- for()
- for in
- for of
- while()
- continue

- break

判断

- if else
- 三元运算符
- switch(),三个=判断

数据类型之间的转换

- 1.==比较
- 2.===绝对相等

2.正式课内容

1.变量提声(预解释)

- 什么是变量提声
- var /function
- let const
- 块级作用域下 {}
- 特殊情况

```
console.log(a);
console.log(A);
if(![]==false){ //[]==false和![]==false 都是 true
  console.log(a);
  console.log(A);
  var a=1;
  function A(){};
}

function fn(){
   console.log(f);
   return f;
   function f(){};
}
var f=fn();
```

2.作用域

- 全局作用域
- 私有作用域
- 内存(堆内存:存储东西的比如因数据类型的存储地址,栈内存:作用域)
- 私有变量
    - 形参
    - 在私有作用域下声明过的变量
- 上一级作用域
- 作用域链

```
var a=1;
function fn(a){
   function f(){//f=xxxfff000
      console.log(a)
   }
   return f;//xxxfff000
}
var f=fn(++a);//f=xxxfff000
f();
```

- 作用域销毁机制
  - 全局作用域关闭浏览器才销毁
  - 私有作用域立即销毁:没有返回值,或者返回值没有被占用
  - 不销毁:返回一个引用数据类型的地址被外界占用

```
function fn(n){
    n++;//n=2
    return function(m){
        m++;
        console.log(m+n)
    }
}
var f=fn(1);
f(2);// 5
fn(1)(2);//5
```

this问题

- 看函数执行的时候前面有没有点

```
FF.prototype.fn();
fn中的this是FF.prototype
```

- 给元素绑定事件

```
box.addEventListener("click",obj.fn,false);
obj.fn里面的this就是box
```

- 自执行函数 window
- 函数当做参数的时候 window

```
window.setInterval(obj.fn,10000);
obj.fn中的this仍然是window
```

- 构造函数中this是当前实例

- 箭头函数this问题
- jQ对象.each()/$.each(),this==item
- JQ原型扩展,jQuary.fn.extend({});this是当前JQ的实例
- 修改this指向
    - call
    - applay
    - bind
    - 数组的方法(传回调函数的,第二个参数可以修改this)

面向对象(面向类的封装,继承和多态)

- 单例模式(高级单例模式)

```
var public=(function (){
    return {}
})()
```

- 工厂模式(了解)
- 构造函数模式
    - 自定义的类
    - new
    - 默认return this
    - 通过this给实例增加私有属性
    - 创建数据类型的方式
        - 字面量创建
        - 构造函数创建
- 原型模式
    - prototype增加公有属性
    - _ *proto* _
    - 原型是一个对象,浏览器默认开辟的推内存中有constructor
    - 原型图
    - 原型链
    - 数组原型上方法的实现

call,apply

- call(null/undefined/ 不传) this->window
- call方法实现的原理

```
Function.prototype.call=function(){
        var ary=[...arguments];
        ary.shift();
        if(arguments[0]==undefined){
            this(...ary);
            return;
        }
        var obj=Object(arguments[0]);
        obj.__proto__.fn=this;
        obj.fn(...ary);
    };
```

- call.call.call…..(fn,obj,x,y,…..)->fn.call(obj,x,y….)
- call用途

```
[].slice.call()
Object.toString.call()
```

- apply用途

```
Math.Max.apply(null,ary)
```

ajax

- new XMLHTTPRequest()
- xhr.open(“GET”,url,false)
- xhr.onreadyStateChange()
- xhr.send()

JSON

- JSON.parse
- JSON.stringify
- 括号表达式

    - 保证语法正确
    - 是一个表达式,返回值最后的值(1,2,3)->3

正则(很重要)

- 创建方式(注意构造函数创建方式可以实现变量的拼接,遇到'\'写成 '\\')
- 元字符
- 特殊意义的元字符
- 量词元字符
- []
- |
- 修饰符g
- test true/false
- exec null/[]
    - 返回值是个数组[捕获的内容,内容首字符的索引,元字符串]
    - 一次只能捕获一个,正则没有修饰符g,捕获永远是第一个
    - 加上修饰符g,可以全都捕获出来,但是需要多次捕获,直到结果为null,自己可以写一个while循环

- str.match()
- 懒惰性 ->g
- 贪婪性 ->在量词元字符后面加一个?
- replace(正则,function(){})
- 分组捕获,不捕获可以在(?:)
- 正向预查
- 负向预查

案例(务必要写一遍)

H5表单验证,提交到另一个页面展示内容

JS盒子模型

- 十三个属性,值是个数字没有单位,浏览默认四舍五入了
- clientWidth/clientHeight,clientLeft/clientTop
- offsetWidth/offsetHeight offsetLeft/offsetTop offsetParent
- scrollWidth/scrollHeight scrollLeft/scroolTop(只有这俩是可修改的)

- 浏览器的宽高
- 整个网页的宽高

DOM库

- toArray()
- toJSONObj()
- win()
- offset()
- prev()
- prevAll()
- next()
- nextAll()
- sibiling()
- sibilings()
- index()
- children()
- getCss()
- setCss()
- setGroupCss()
- css()
- addClass()
- removeClass()
- hasClass()
- toggleClass()

延迟加载(很重要)

- 原理
- try{} catch(e){}
- DOM映射

继承(非常重要)

- 原型继承
- call继承

- 组合继承

动画库(自己再写一下)

JQ

事件(非常重要的)

- 什么事件
- 事件行为
- 事件绑定
    - DOM0级事件绑定
    - DOM2级事件绑定

- 事件对象
- 事件传播
- 阻止事件冒泡
- 阻止浏览器的默认行为
- on,off(很重要)