

基本语法

数据类型

基本包装类型

- 普通变量不能直接调用属性或者方法
- 对象可以直接调用属性和方法
- 本身是基本类型,但是在执行代码的过程中,如果这种类型的变量调用了属性或者是方法,那么这种类型就不再是基本类型了,而是基本包装类型,这个变量也不是普通的变量了,而是基本包装类型对象
- 如果是一个对象&&true,那么结果是true
- 如果是一个true&&对象,那么结果是对象

转整数

- parseInt
 - console.log(parseInt("10"));//10
 - console.log(parseInt("10afrswfdsf"));//10
 - console.log(parseInt("g10"));//NaN
 - console.log(parseInt("1fds0"));//1
 - console.log(parseInt("10.98"));//10
 - console.log(parseInt("10.98fdsfd"));//10

转小数

- parseFloat
 - console.log(parseFloat("10"));//10
 - console.log(parseFloat("10afrswfdsf"));//10
 - console.log(parseFloat("g10"));//NaN
 - console.log(parseFloat("1fds0"));//1
 - console.log(parseFloat("10.98"));//10.98
 - console.log(parseFloat("10.98fdsfd"));//10.98

转数字

- Number
 - console.log(Number("10"));//10
 - console.log(Number("10afrswfdsf"));//NaN
 - console.log(Number("g10"));//NaN
 - console.log(Number("1fds0"));//NaN
 - console.log(Number("10.98"));//10.98
 - console.log(Number("10.98fdsfd"));//NaN

总结

- 想要转整数用parseInt(),想要转小数用parseFloat()
- 想要转数字:Number(),要比上面的两种方式严格

其他类型转字符串类型

- .toString()
 - 如果变量有意义调用.toString()使用转换
 - 如果变量没有意义使用String()转换

其他类型转布尔类型

- Boolean
 - console.log(Boolean(1));//true
 - console.log(Boolean(0));//false
 - console.log(Boolean(11));//true
 - console.log(Boolean(-10));//true
 - console.log(Boolean("哈哈"));//true
 - console.log(Boolean(""));//false
 - console.log(Boolean(null));//false
 - console.log(Boolean(undefined));//false

运算符

- 算数运算符: + - * / %
- 一元运算符: 这个操作符只需要一个操作数就可以运算的符号 ++ --
- 二元运算符: 这个操作符需要两个操作数就可以运算,
- 三元运算符: ? :
- 复合运算符 += -= *= /= %=
- 关系运算符: > < >= <= != 不严格的 === 严格的 !== 不严格的不等 !== 严格的的不等
- 逻辑运算符:
 - &&---逻辑与---并且
 - 表达式1&&表达式2
 - 如果有一个为false,整个的结果就是false
 - ||---逻辑或---或者
 - 表达式1||表达式2
 - 如果有一个为true,整个的结果为true
 - !---逻辑非---取反--取非
 - !表达式1
 - 表达式1的结果是true,整个结果为false
 - 表达式1的结果是false,整个结果为true
- 赋值运算符: =

if

- if-else
- if-else if-else if-else
- switch case

内置对象

String

- console.log(dt.toDateString());//英文的---日期
- console.log(dt.toLocaleDateString());//数字格式--日期
- console.log(dt.toTimeString());//小时分钟秒
- console.log(dt.toLocaleTimeString());//小时分钟秒
- console.log(dt.valueOf());//毫秒值
- console.log(getDate(new Date())): ____ 格式化日期
- 不可变性,字符串的值是不能改变
- 字符串可以通过索引访问字符串中的某个值
- 引用类型
- 实例方法 ____ 必须要通过new的方式创建的对象(实例对象)来调用的方法
- 静态方法 ____ 直接通过大写的构造函数的名字调用的方法(直接通过大写的对象名字调用的)
- .length----->字符串的长度
- .charAt(索引),返回值是指定索引位置的字符串,超出索引,结果是空字符串
- .fromCharCode(数字值,可以是多个参数),返回的是ASCII码对应的值
- .concat(字符串1,字符串2,...);返回的是拼接之后的新的字符串
- .indexOf(要找的字符串,从某个位置开始的索引),返回的是这个字符串的索引值,没找到则返回-1
- .lastIndexOf(要找的字符串);从后向前找,但是索引仍然是从左向右的方式,找不到则返回-1
- .replace("原来的字符串","新的字符串"),用来替换字符串的
- .slice(开始的索引,结束的索引);从索引5的位置开始提取,到索引为10的前一个结束,没有10, 并返回这个提取后的字符串
- .split("要干掉的字符串",切割后留下的个数);切割字符串
- .substr(开始的位置,个数);返回的是截取后的新的字符串
- .substring(开始的索引,结束的索引),返回截取后的字符串,不包含结束的索引的字符串
- .toLocaleLowerCase();转小写
- .toLowerCase();转小写
- .toLocaleUpperCase();转大写
- .toUpperCase();转大写
- .trim();干掉字符串两端的空格

数组

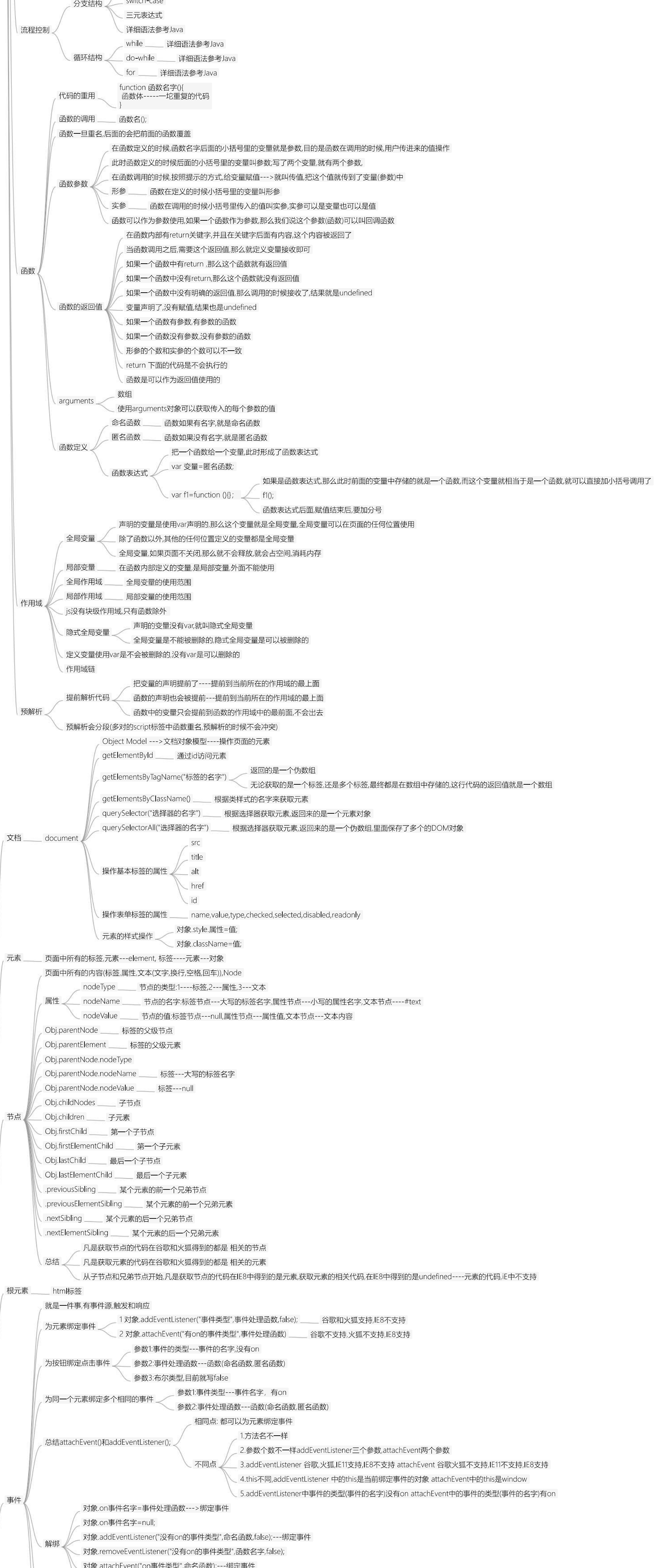
- 一组有序的数据
- 可以一次性存储多个数据
- 数组元素 ____ 数组中存储的每个数据,都可以叫数组的元素,比如:存储了3个数据,数组中3个元素
- 数组长度 ____ 就是数组的元素的个数,比如有3个元素,就说,这个数组的长度是3
- 数组索引(下标) ____ 用来存储或者访问数组中的数据的,索引从0开始,到长度减1结束
- 数组的索引和数组的长度的关系 ____ 长度减1就是最大的索引值
- 创建数组
 - 构造函数
 - var 数组名=new Array(); ____ 数组的名字如果直接输出,那么直接就可以把数组中的数据显示出来,如果没有数据,就看不到数据
 - var 数组名=new Array(长度);
 - 如果数组中没有数据,但是有长度---,数组中的每个值就是undefined
 - 构造函数的方式创建数组的时候,如果在Array(一个数字)---->数组的长度(数组元素的个数)
 - 如果在Array(多个值),这个数组中就有数据了,数组的长度就是这些数据的个数
 - 通过字面量的方式创建数组 ____ var 数组名=[];//空数组
 - 无论是构造函数的方式还是字面量的方式,定义的数组,如果有长度,那么默认是undefined

Array

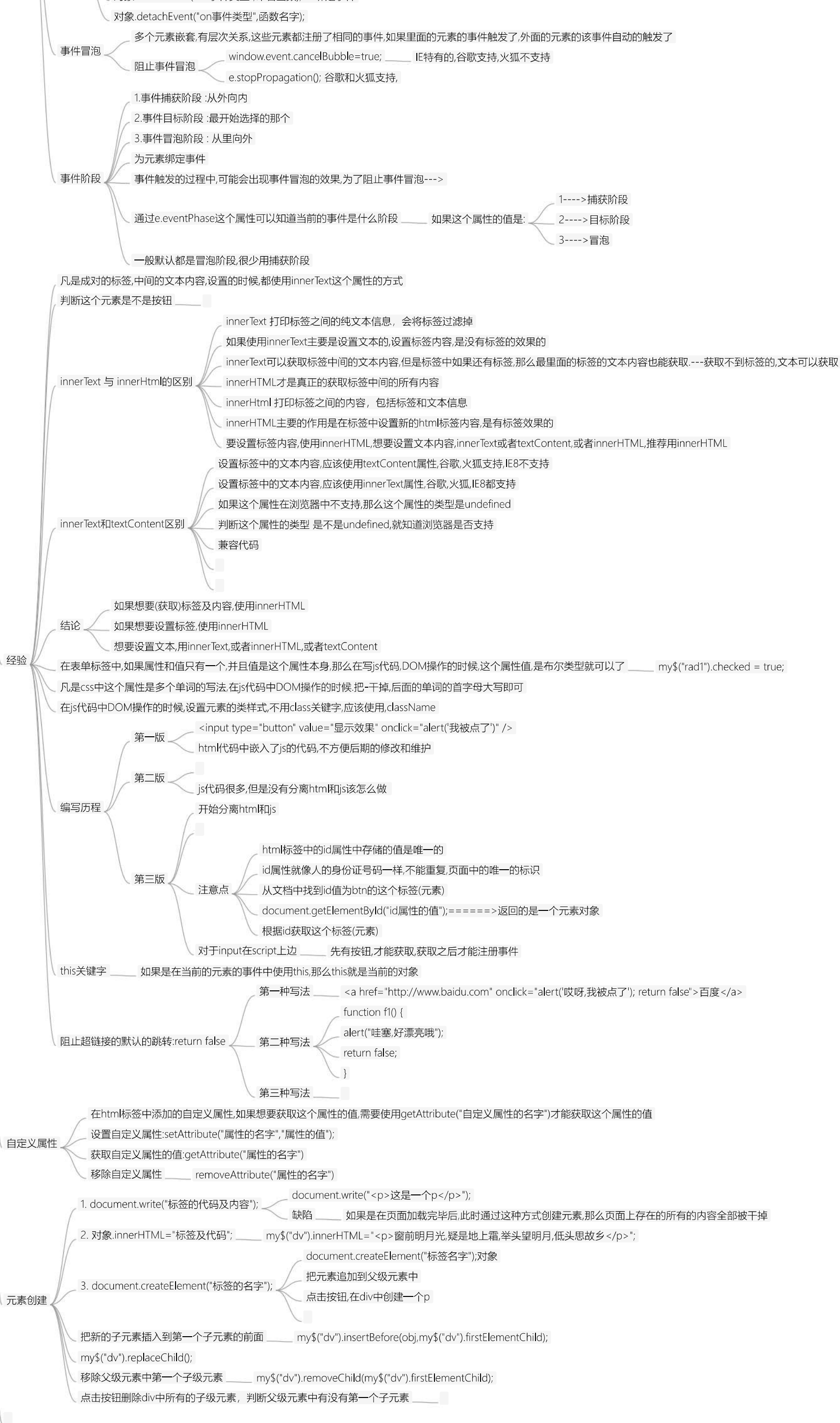
- Array.isArray(对象)---->判断这个对象是不是数组
- .concat(数组,数组,数组...) 组合一个新的数组
- .every(函数)--返回值是布尔类型,函数作为参数使用,函数中有三个参数,第一个参数是元素的值, 第二个参数是索引值,第三个参数是原来的数组(没用)
- 如果这个数组中的每个元素的值都符合条件,最后才返回的是true
- .filter(函数),返回的是数组中每一个元素都复合条件的元素,组成了一个新的数组
- .push(值);---->把值追加到数组中,加到最后了---返回值也是追加数据之后的数组长度
- .pop();---->删除数组中最后一个元素,返回值就是删除的这个值
- .shift();---->删除数组中第一个元素,返回值就是删除的这个值
- .unshift();---->向数组的第一个元素前面插入一个新的元素,----返回值是插入后的程度
- .forEach(函数)方法---遍历数组用---相当于for循环
- .indexOf(元素值);返回的是索引,没有则是-1
- .join("字符串");---->返回的是一个字符串
- .map(函数);---->数组中的每个元素都要执行这个函数,把执行后的结果重新的全部的放在一个新的数组中
- .reverse();----->反转数组
- .sort();---排序的,可能不稳定,如果不稳定,请写MDN中的那个固定的代码
- .arr.slice(开始的索引,结束的索引);把截取的数组的值放在一个新的数组中,但是不包含结束的索引对应的元素值
- .splice(开始的位置,要删除的个数,替换的元素的值);一般用于删除数组中的元素,或者是替换元素,或者是插入元素

Object

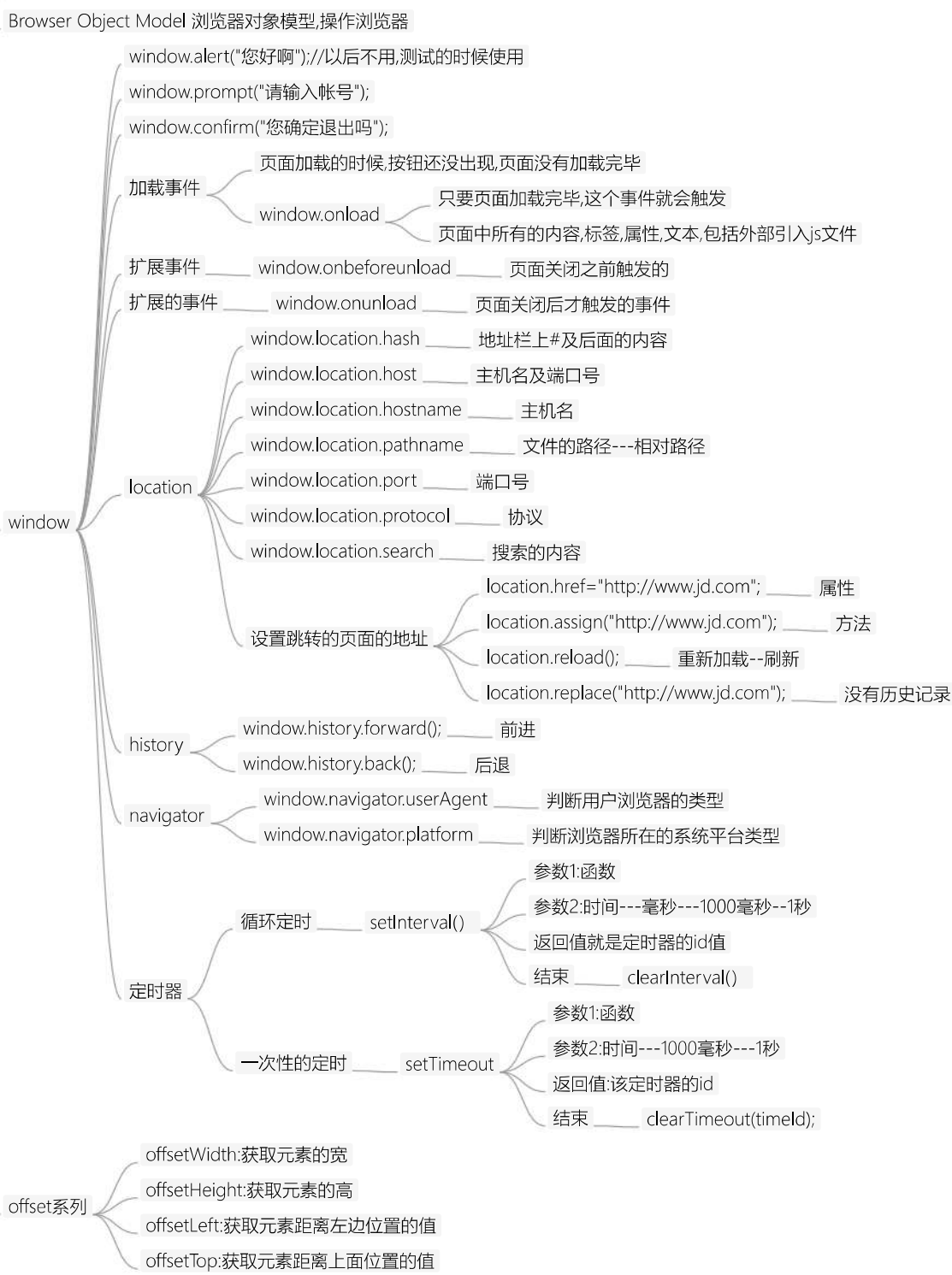
JavaScript



DOM



BOM



ECMAScript

是一门脚本语言

