javaScript基础

- **介绍**
  - 编程语言，弱数据类型
  - 组成
    - EcmaScript()基础语法
    - webApi
      - Dom页面文档对象模型
      - BOM浏览器对象模型
- **总体概念**
  - 书写位置
    - 行内
    - 外部
      - 引用语法：<script src=""><script>
    - 内部
  - 输出
    - document.wirte
      - 在界面输出
    - console.log
      - 在控制台输出
    - alert
      - 弹窗
  - 输入
    - prompt
      - 弹出弹窗输入
  - 字面量
    - 在计算机中描述事物
- **变量**
  - 命名规范
    - 数字，字母，下划线，$组成，不能以数字开头
  - 声明
    - 用let声明
- **常量**
  - 命名规范
    - 同变量
  - 声明
    - 用const声明
  - 特性
    - 不能改变
    - 声明时必须要赋值
- **数据类型**
  - 基本数据类型
    - 数字类型number
      - 只要是数字都是number
    - 字符串类型string
      - 模版字符串
        - 必须用反引号``来规定
        - 用来输出变量和汉字一起
      - 拼接字符串
        - 直接用加号进行拼接
      - 字符串定义
        - 可以用'',"",``来定义，都是一样的
    - 未定义类型undefined
      - 比如let num没有赋值，这个时候就是undefined
      - undefined+任何值=nan
    - 空类型null
      - let赋值了，但是里面是空值，利用typeof检查是对象
    - 布尔类型boolean
      - 只有两个值，true和flase
    - 基本数据类型存放在栈中
  - 引用数据类型
    - 就是对象，数组也是，存储方式是将地址存在栈里面，而实际量存在堆中
  - 检测数据类型
    - typeof 变量或者是typeof(变量)
- **数据类型转化**
  - 隐式转化
    - 加号左右两边只要有一个是字符串，就把另一个自动转化为字符串
    - 减号，乘号，除号两边有一个数字，就把另一个自动转化为数字
  - 显式转化
    - number（)能够转化为数字类型
    - parseint（）能够转化为整数，不过只取开头的整数部分
    - parsefloat（）能够也取小数部分，也是开头的
    - 加号放在前面能够转化为整数部分，用的最多的就是+prompt()
  - 注意
    - prompt（）取到的都是String类型
- **运算符**
  - 赋值运算符
    - +，+=，-，-=....和java相同
  - 一元运算符
    - 一元运算符的意思就是操作数只有一个
    - 在数据类型前面加'+'，这是转化为数字类型
    - 前置++和--
  - 比较运算符
    - >,<...和java相同
    - =是赋值的意思
    - ==是比较两个数的值是否相等，不考虑数据类型
    - ===是比较两个数是否完全相等，开发中用的多
      - 特殊情况：nan===nan是false
      - 两个字符串比较，首先比较前面的，大了以后就不看后面的，否则往后继续
      - 两个字符串比较，如果前面aa，后面aac，后面大
  - 逻辑运算符
    - && ||一个逻辑与一个逻辑或
    - 最主要是的短路原则，&&前面是false，后面不看，||前面是true，后面不看
  - 运算符优先级
    - 括号>一元>算术>关系>相等>逻辑>赋值>,
    - 逻辑运算符：&&>||
  - 特别注意：这里面的/号用于整数之间，不会取整，除以多少就是多少
- **语句**
  - 语句分类
    - 顺序结构
    - 分支结构
    - 循环结构
  - 分支结构
    - if
      - if
      - if,else
      - if,else if,else
    - switch
      - 结构要记牢，代码中若有多行语句，不用加()
  - 循环语句
    - while
    - do while
    - for
    - continue和break
- **数组**
  - 定义数组
    - 初始化的语句是let num=[1,2,3],注意不是大括号，是中括号
    - 还可以用let num=new array(1,2,3,4),这是小括号
  - 数组的增删改查
    - 增
      - 数组名.push（a,b..）
        - 往数组后面插入一个或多个元素，这个函数有返回值，返回的是此时插入过数组的长度
      - 数组名.unshift（a,b...）
        - 往数组前面插入一个或多个元素，这个函数有返回值，返回的是此时插入过数组的长度
    - 删
      - 数组名.pop()
        - 往数组后面删除一个元素，返回值是删除的元素
      - 数组名.shift()
        - 往数组前面删除一个元素，返回值也是删除的元素
      - 数组名.splice(a,b)
        - 往数组指定位置删除元素，a是数组下标，b是删除元素的个数
    - 改
      - a[0]=count
    - 查
      - a[0]
  - 数组的排序
    - 冒泡排序等排序算法
    - 数组名.sort()是升序
    - 数组名.sort(function(a,b){return b-a})是降序
- **函数**
  - 为什么需要函数
    - 实现代码复用
  - 函数的声明
    - function(){}
    - 如果函数没有传参，就会变成undefined
    - 如果传入的是数组，形参和实参他们都只要写数组名即可，不用[]
  - 匿名函数
    - 函数表达式
      - 例如：let fun=function(){},如果要调用，fun(),参数写在括号里面
    - 立即执行函数
      - 第一种写法(function(){})();参数写在小括号里面，后面的是实参，前面的是形参
      - 第二种写法(function(){}());参数同上
      - 如果在里面用变量未声明，那么久转化为全局变量
      - 上面的两种方法都要加分号，写在最前面也行，最后面也行
- **补充**
  - 逻辑中断
    - &&,前面如果是假，后面就不看
    - ||前面如果是真，后面就不看
  - 转化为boolean
    - 对于数字类型来说，非0代表真，对于字符串来说，非空字符串代表真，对于null，nan，undefined都是假
    - 转为数字类型时，null=0,undefined=nan
- **对象**
  - 对象整体说明
    - 因为之前的数据类型无法表示真正的含义，所以引入对象
    - 对象和属组的区别就是，一个无序一个有序
    - 对象的声明
      - let obj={方法，属性}，之间要用逗号分离
    - 类似于强类型语言的类，有属性，有方法
  - 对象的增删改查
    - 增
      - 对象.新属性=值
    - 删
      - delete 对象.属性
    - 改
      - 对象.属性=值
    - 查
      - 方法一：对象.属性
      - 方法二：对象['属性']
  - 对象的遍历
    - 因为是无序的，所以不能用for
    - 用for in遍历，for(let i in obj),类似于C#的foeach
      - 这个i是字符串类型，所以调用的时候结合查的第二种方法，是obj[i]
  - 内置对象Math
    - 常见的方法
      - math.random()表示0-1的随机数，（0,1]
      - math.floor(a)表示向下取整
      - math.ceil(a)表示向上取整
      - math.round(a)表示四舍五入
      - math.abs(a)表示取绝对值
      - math.max(a,b......)表示取最大值
      - math.min(a,b,,,,,,,,,,,,,)表示取最小值
    - 随机数方法
      - 取0-10的随机数
        - math.floor(math.random()*11),因为这里的右边取不到
      - 取5-10的随机数
        - math.floor(math.random()*6+5)
      - 取n-m的随机数
        - math.floor(math.random()*(m-n+1)+n)

Presented with xmind