# 基础语法

## 流程控制

### if

1. if（条件）{如果满足条件}else{否则进入}或者是 if（条件）{}else if（条件）{否则如果} else{否则进入}

if(age>=18)//如果年纪大于等于18

consolo.log("年纪18");

else//否则输出年纪+age

consolo.log("年纪"+age);

PS：

1.if后面如果是只有一个语法/一行代码 可以忽略{}。

2.分号";"也是一条语句，空语句。

3.if else 是一个整体，else匹配它同级的上一个if。

4.对于非布尔类型的会先转换成boolean后再判断。

5.比较时推荐将常量写再变量前

6.if else 可以嵌套使用

猜拳游戏

### Switch

1. Switch是一种等值判断,必须要与case以及break配套使用.它会根据预置的进入条件直接进入相关代码块.与IF相比适用范围较窄.
2. 关键字介绍:
   1. case 条件: 符合该条件的进入
   2. defult:如果没有符合case的情况下进入
   3. break:匹配到条件了就退出.(必须搭配使用,否则会在符合case的情况下再次进入defult.)
3. 示列:
   1. Switch(num){
      1. Case 1:
         1. Console.log(“这是1”)
         2. Break;
      2. Case 2:
         1. Console.log(“这是2”);
         2. Break;
      3. Defult:
         1. Console.log(“这是其他数字”);
         2. Break;
   2. }
4. 示例:录入分数给出评级(看代码)

### 循环

循环流程

1. 循环条件:根据条件(true/false)来判断是否需要进入/退出循环.
2. 执行代码块:也就是大括号”{}”的内容
3. 重复第一条
4. 关键字
   1. Break: 退出循环
   2. Continue: 终止本次循环继续下次循环.

#### While循环

1. 循环结构: while(条件){执行的代码块}
2. 循环特点: 不知道循环次数直到不符合条件.

#### Do-while

1. 循环结构: do{执行的代码块}while(条件)
2. 循环特点: 不知道循环次数直到不符合条件,但是会先执行一次代码块

#### For

1. 循环结构: for(条件){执行代码块}
2. 循环特点: 知道循环次数.

## 对象

### Function:函数

#### 定义

#### 作用

1. 将重复代码块进行封装,减少代码冗余

#### 基本使用

* 1. 声明
     1. Function name(){
        1. Console.log(“基本使用”);
     2. }
  2. 使用
     1. Name();

#### 常见的声明方式

1. 标准声明: function getSum(num) {},加载优先级高于表达式声明
2. 表达式声明: var getSum=function(num){}. 使用必须在声明之后
3. 不常见的方式: var func=new Function("console.log('不常用函数声明方式');");

#### 函数的参数

1. 形参:用于调用时给需要传递的参数所声明的参数,在声明函数时使用
2. 实参:在调用函数时传递需要进行实际使用的参数,在调用时使用.
3. 注意:
   1. 在调用函数时,可以在()中指定实参,实参将会赋值给函数中对应的形参.
   2. 调用函数时解析器不会检查参数类型,所以必要时需要在函数中进行参数类型校验
   3. 函数中的实参可以是任意数据类型
   4. 调用函数时,解析器不会检查参数数量,多余实参不会杯复制,如果实参数量少于形参数量,则没有对应实参的形参将是undefined.

#### Arguments对象

1. 函数中实参的伪数组,它会将调用函数传递的实参全部放在它里面.

#### Length属性

1. 用于获取当前对象的长度,如果是数组返回个数,如果是字符串则返回字符串的长度.

#### 函数返回值

1. 关键字:return
2. 在方法/函数中返回相关信息,或者用于结束当前函数
3. 如果没有具体返回信息则返回 undefined

#### 匿名函数

1. 声明全局函数
2. 定时器:setInterval
3. 立即执行函数(function(num,num2){console.log(num1,num2)})(1,2)

#### 函数是一种数据类型

#### 回调函数

#### 变量的作用域

### 对象

#### JS中为什么要用对象

1. 过多的参数使用对象进行封装,可以简化代码,提高重用性和可维护性.

#### 什么是对象

1. 类: 类型/类别.比如人类和动物.在JavaScript中默认类是Object
2. 对象: 有自己的特征和行为

#### JS中的对象

* + - 1. 是一种无序属性的集合,其属性可以是基本值,对象,函数
      2. 对象是一组没有顺序的值,由一组组键值对构成

#### JS中对象的分类

#### 对象字面量

#### 常见创建对象的方式

#### 属性和方法

#### New关键字

#### This关键字

#### 对象应用

#### 基础类型和引用类型在堆栈中的表示