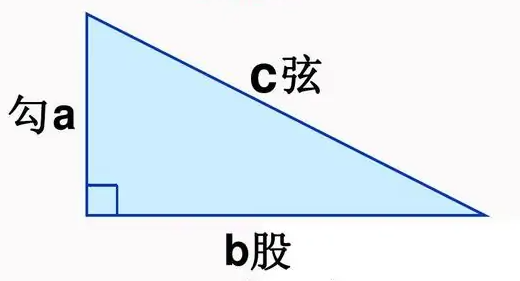
1. 创建求和函数，求两个数的加和。
2. 创建求和函数，求N个数的加和。
   1. 函数结构要求为function fn(参数1, 参数2, ...)
   2. 函数结构要求为function fn(【参数1, 参数2, ...】)
3. 创建一个计算器，要求拥有
   1. 基本加法、减法、乘法、除法和取余运算
   2. 四舍五入、向上取整和向下取整运算
   3. 求平方、求立方、开平方运算
   4. 补充要求：计算器只能存在一个输入框面板
4. 创建勾股定理函数
   1. 当给定任意两边后，能计算出第三条边的长度

任意两边可以是勾、股、弦任意两边。



1. 编写斐波那契数列求和函数
   1. 函数参数为需要计算的斐波那契数列的长度
   2. 函数执行后能够具有返回值结果，结果是参数长度斐波那契数列的加和。
2. 将前五个练习中的函数使用箭头函数改写，并保证代码正确执行。
3. 已知数据结构，要求创建不同功能的函数。
   1. var product = [

{ "title": "手机", "price": 2200, "num": 2 },

{ "title": "电饭煲", "price": 300, "num": 5 },

{ "title": "电冰箱", "price": 5000, "num": 1 }

];

* 1. 创建求电饭煲所在的索引值函数
  2. 创建能够筛选出价格大于300的数据的函数
  3. 创建能够筛选出价格大于1000的产品名称的函数

1. 使用IIFE实现钢琴块练习功能。
2. 创建冒泡排序功能函数
   1. 函数执行时传入乱序数组，函数执行完毕后能够将数组冒泡排序返回
3. 使用递归函数实现1~N数字求和
4. 创建薪资结构计算函数
   1. 已知第一年薪资是6k
   2. 每年涨薪8%
   3. 函数能够接收年份作为参数，并计算经过该年份后薪资是多少，并返回
5. 已知数据结构，要求创建实现功能函数
   1. var userlist = [

[1, "A根节点", 0],

[2, "A二级节点1", 1],

[3, "A二级节点2", 1],

[4, "A二级节点3", 1],

[5, "A三级节点1", 2],

[6, "A三级节点2", 2],

[7, "A三级节点3", 2],

[8, "A三级节点4", 3],

[9, "A三级节点5", 3],

[10, "A三级节点6", 3],

[11, "A三级节点7", 4],

[12, "A三级节点8", 4],

[13, "A三级节点9", 4],

[14, "B根节点", 0],

[15, "B二级节点", 14]

];

数组元素结构为【当前节点ID、自己名称、父节点ID】

* 1. 要求函数接收当前节点id作为参数，返回当前节点的根节点
  2. 根节点的特征为：父节点id为0

例如

findRoot(6) ==>[1, "A根节点", 0]

findRoot(15) ==>[14, "B根节点", 0]