

# 伪代码与流程图

从入门到工作：算法入门

# 版权声明

本内容版权属杭州饥人谷教育科技有限公司（简称饥人谷）所有。

任何媒体、网站或个人未经本网协议授权不得转载、链接、转贴，或以其他方式复制、发布和发表。

已获得饥人谷授权的媒体、网站或个人在使用时须注明「资料来源：饥人谷」。

对于违反者，饥人谷将依法追究法律责任。

# 联系方式

如果你想要购买本课程

请微信联系 [xiedaimala02](#) 或 [xiedaimala03](#)

如果你发现有人盗用本课程

请微信联系 [xiedaimala02](#) 或 [xiedaimala03](#)

# 大家觉得 JS 语法难不难

有没有发现语法不难，难的是抽象概念

你经常发现  
每句都懂，加在一起不懂

举个栗子

# 例子

```
function t(fn, delay){  
  let fn2 = ()=>{  
    if(fn2.timer){return}  
    fn()  
    fn2.timer = setTimeout(()=>{  
      fn2.timer = null  
    },delay)  
  }  
  return fn2  
}
```

```
let f = t(()=>{console.log('火球术')},5000)
```

这代码在搞什么？

# 这就是你看不懂代码的原因

你没有学习抽象知识

# 抽象能力

越高级的程序员，抽象能力越强



# 编程语言也能抽象

JS、Python、Java、PHP 根本就没区别

# 这些语言都有

- 声明
- if...else...
- while 循环、for 循环
- 函数
- 对象

# 这意味着

- ✓ 你没有必要过分关注语言的细节
- ✓ 你应该首先理解这些语言的不变之处
- ✓ 然后以不变应万变

# 有什么东西是不变的

从第一位程序员诞生至今

# 一、对编程的热爱

怎么会有人爱这么枯燥的玩意

Most good programmers do programming not because they expect to get paid or get adulation by the public, but because it is fun to program.

大多数程序员愿意写代码，不是因为他们获得高额的薪水或者他人的赞誉，而是因为写代码很有意思。

Linus Torvalds  
Linux 之父、Git 之父

I'm not a great programmer;  
I'm just a good programmer with great habits.

我不是一个牛X的程序员；  
我只是一个拥有牛X的习惯的还凑活的程序员。

Kent Beck  
极限编程创始人

我想让你知道

不是所有人都只为工资编程

虽然一开始是为了工资



# 简洁之美

人们喜欢编程的原因

## 二、逻辑

所有的编程，都在使用逻辑

# 逻辑是什么

- 三段论逻辑

命题一：JS 有 7 种数据类型：string、number、bool、symbol、undefined、null、object

命题二：JS 的函数不属于前六种类型

如果命题一、二为真，那么我们可以得到

推论：JS 的函数属于 object 类型

# 逻辑是什么

- 三段论逻辑

命题一：JS 中所有函数都是由 Function 构造的

命题二：Function、Object、Array 是函数

推论：Function、Object、Array 是由 Function 构造的

# 逻辑 V.S. 直觉

- 直觉的好处

- ✓ 直觉能让你快速学会一些东西
- ✓ 比如数字、字符串、数组、对象都可以靠直觉学会

- 直觉的坏处

- ✓ 直觉却又能阻止你学会另一些东西
- ✓ 这时你就只能依靠逻辑来说服自己
- ✓ 比如虽然不知道 Function 是如何构造 Function 的，但是逻辑告诉我们，确实如此

# 如何用代码表示逻辑

- 栗子

- ✓ 输出两个数中较大的一个数
- ✓ 如果第 1 个数大于第 2 个数，就输出第 1 个数
- ✓ 否则输出第 2 个数

- 代码

- ✓ `a = [12, 32]`
- ✓ `if a.get(0) > a.get(1)`
- ✓ `print a.get(0)`
- ✓ `else`
- ✓ `print a.get(1)`

# 用 if...else... 可以表示逻辑

发现规律

# 如何用代码表示逻辑

## • 栗子2

- ✓ 输出 N 个数中最大的一个数
- ✓ 找到第 1、2 个数字中较大的那一个，存入 max
- ✓ 找到 max 和第 3 个数字中较大的那个，存入 max
- ✓ 找到 max 和第 ... 个，存入 max
- ✓ max 就是最大的数

## • 代码

```
a = [12, 32, 9, 1 ,23, 6]
```

```
max = a[0]
```

```
for i from 1 to a.length() - 1
```

```
    if a.get(i) > max then max = a.get(i)
```

```
print max
```



# 用 for 循环可以表示逻辑

又发现规律

# 表示逻辑 需要多少种语句

掌握了这些语句就能表达所有逻辑

# 只需要三种语句

「结构化编程」理论\*

\*但结构化编程在某些情况下会变得麻烦，比如错误处理

# 三种语句搞定所有逻辑

- 顺序执行语句

- ✓ 语句1
- ✓ 语句2

- 条件判断语句

- ✓ if ... then ... else ...
- ✓ if ... else if ... else

- 循环语句

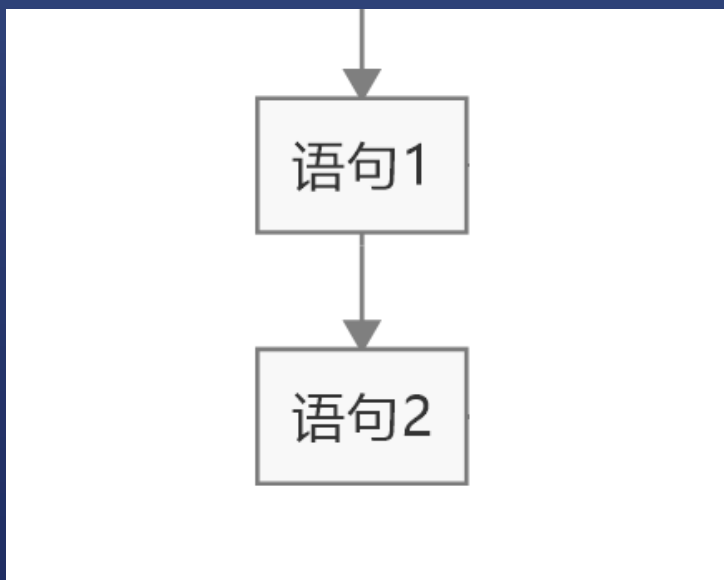
- ✓ while ... do ...
- ✓ for i from 1 to n ...

所有的复杂算法  
都能用这三种语句描述

多么简洁

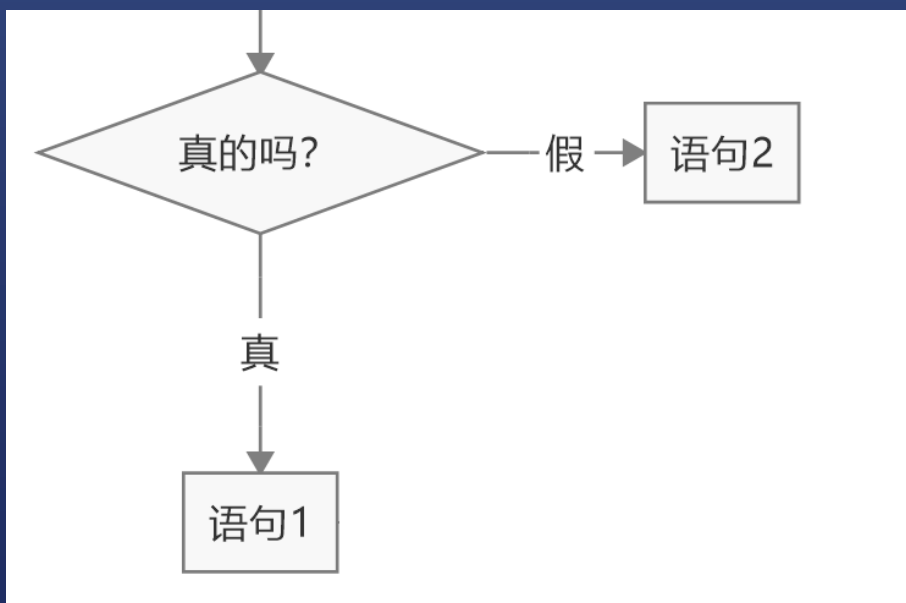
# 顺序执行语句

语句1  
语句2



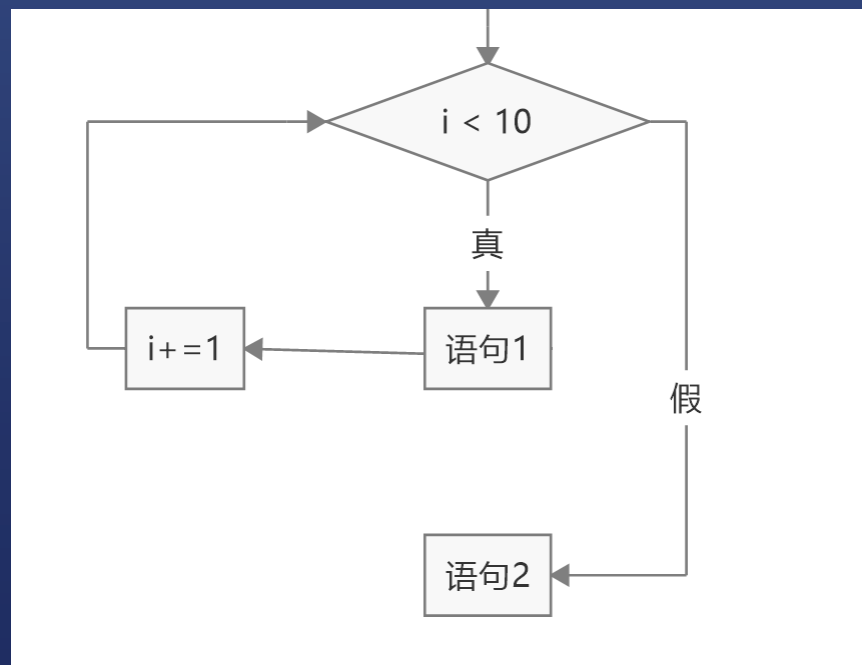
# 条件判断语句

```
if x > 10  
    语句1  
else  
    语句2
```



# 循环语句

```
when i < 10  
    语句1  
    i+=1  
语句2
```





# 推荐使用 Mindjet 画流程图

也可以用 XMind、百度脑图等工具代替

# 流程图、伪代码的好处

- 锻炼你的大脑

- ✓ 你必须自己画出来
- ✓ 不能运行在计算机里

- 整理你的思路

- ✓ 思路乱，则图乱
- ✓ 伪代码都写不好，则代码更写不好

# 用流程图找到最大数

两个数、N 个数

# 总结

- 逻辑很重要
- 用三种语句表达逻辑
- 用图和伪代码表示三种语句

# 三、数据结构

就是数据与数据之间的关系和结构

# 如何表示两个数据

- 如果顺序有意义

- ✓  $[x, y]$  表示第一个是  $x$ ，第二个是  $y$
- ✓  $[y, x]$  表示第一个是  $y$ ，第二个是  $x$
- ✓ 比如坐标就是这样的数据
- ✓ 要提供 first 和 last 操作

- 如果顺序无意义

- ✓  $(x, y)$  和  $(y, x)$  一样
- ✓ 比如血压值  $(120, 80)$  和  $(80, 120)$  没区别
- ✓ 不需要提供 first 和 last 操作

# 如何表示 N 个数据

- 如果顺序有意义

- ✓ 数组表示  $[a_1, a_2, \dots, a_N]$
- ✓ 要提供索引操作  $\text{get}(i)$
- ✓ 要提供  $\text{add}$  /  $\text{indexOf}$  /  $\text{delete}$  操作

- 如果顺序没有意义

- ✓ 集合表示  $\{a_1, a_2, \dots, a_N\}$
- ✓ 今天所有的代码都跟 JS 无关
- ✓ 要提供  $\text{add}$  /  $\text{delete}$  /  $\text{has}$  操作

# 如何表示 N 对 N 数据

- 比如学号

- ✓ 用「哈希表」表示

- ✓  $\text{hash} = \{1001 \Rightarrow \text{'小方'}, 1002 \Rightarrow \text{'小红'}\}$



# 数据结构 = 数据形式 + 操作

不同形式的数据暴露不同的操作

# 有没有觉得很抽象

数据结构都是抽象的

# 面试题

- 有一段英文对白，里面只会出现 a-z A-Z、标点符号和空格，请告诉我每个字符出现的次数

- ✓ 例如 Hi, I'm Frank
- ✓ 输出 a 出现 1 次，F 出现 1 次，H 出现 1 次……

# 数据结构的作用

- 提前记住一些结构

- ✓ 这些结构很常见
- ✓ 这些结构能让你很快理清思路
- ✓ 这些结构面试经常问

- 锻炼你的抽象能力

- ✓ 一种数据结构往往能解决很多类似的问题
- ✓ 如果你选错了数据结构，你根本就想不出思路
- ✓ 牛逼的程序员更重视数据结构，而不是算法

# 再见

下节课开始学习算法和数据结构