

0XHEXADECIMAL

# HEXAGON 编程指南

内部文档



# 概览

Hexagon 是一门全新的编程语言。它具有全新的完美类库分离语法、完全函数编程支持、强面向集合性等诸多令人喜爱的特性以及一些经过改善的语法。一睹为快吧。

## 0.1 *Hello, world!*

将以下代码输入一个文件中，把它重命名为 HelloWorld.Hexagon。

清单 1: 你好，世界

```
1 Hexagon::Result() Main = (){
2     var c = import Hexagon::System::Console;
3     c << "Hello, World!" << Hexagon::System::SpecChar::EndLine;
4     c.Command << "PAUSE";
5     return Hexagon::Result.Fine;
6 };
```

打开控制台，运行

```
1 $ hgc HelloWorld.Hexagon
2 <编译提示>
3 $ hgx HelloWorld.Hexec
```

产生输出：

```
1 Hello, World!
```



# 目录

概览      iii

0.1 *Hello, world!*      iii

插图目录      vii

表格目录      ix

清单目录      xi

**第一部分 入门篇      1**

**1 学校成绩数据库      3**

1.1 程序起点      3

1.2 命令行输入/输出      4

1.3 语句      4

**2 表达式与语句      5**

2.1 变量      5

2.2 运算 5

3 函数 7

4 面向对象 9

**第二部分 进阶篇 11**

5 泛型 13

6 面向集合 15

7 数据结构 17

**第三部分 提高篇 19**

8 算法 21

9 自订模块 23

**附录 25**

A 索引 27

# 插图目录





# 表格目录



# 清单目录

1	你好，世界	iii
1.1	main 函数的标准形式	3
1.2	输入/输出流	4



# 第一部分

## 入门篇



# 第 1 章

## 学校成绩数据库

在本章中，我们将为您展示一个十分基本的程序供您参阅和研究。  
这个程序将会具有以下功能：

- 运行时按行读取输入
- 录入学生的成绩
- 查询学生的成绩

接下来，让我们一起来实现它吧。

相关代码可在 < 待填充 > 下载。

### 1.1 程序起点

我们的程序的起点在一个叫做 *Main* 的函数里。

清单 1.1: *main* 函数的标准形式

```
1 Hexagon::Result() Main;           // 声明
2 Main = () {
3     // 要执行的语句
4     return Hexagon::Result.Fine;    // 返回程序的运行结果
5 };
```

我们可以把一条至多条语句放到要执行的语句的位置。如果您是第一次接触这种风格，或是第一次接触编程语言，您可以暂时不理解它们的具体含义。您只需将这段代码反复利用即可。要执行的语句就是程序的起点。

## 1.2 命令行输入/输出

您可以用 `import Hexagon::Console` 来导入所谓控制台输入/输出流。我们将这样使用它：

清单 1.2: 输入/输出流

```
1 import Hexagon::Console;
2 Console << "这是我们提供的输出流。";
3 Console << "请输入一个整数：";
4 Hexagon::Integer i;
5 Console >> i; // 这是输入的方式
```

有关于 `import` 命令和输入/输出流，我们将在以后具体描述，目前您只需了解如何基本地使用它们即可满足需要。

## 1.3 语句

我们在执行一个动作后，用分号“;”来结束表述，放弃现有的运算结果。

### 1.3.1 空语句

如果我们直接使用一个分号，而前面没有任何表达式，这时这个语句成为了一个空语句。语法上来讲，这种做法是允许的；但是为了保持我们的程序尽可能的简洁，而且容易被别人读懂，请直接删除空语句，因为空语句和什么都没有其实是等效的<sup>1</sup>。

<sup>1</sup> Hexagon 中没有任何获取上下文语句的情况，所有的控制流都是直接传入函数对象，这一点与 C/C++ 等语言不大相同。

## 1.4 基本运算

我们提供几个十分基础的算术类型，基本情况如下表所示。

1	2	3	4
1	2	3	4



# 第 2 章

## 表达式与语句

在本章里我们将研究两个极为重要的概念：能够描述一段运算的表达式和一个动作的语句。事实上，语句是表达式的一个超集。

### 2.1 变量

### 2.2 运算

表达一定的数值上的计算一般被叫做运算。例如

$1+1$

$5+3$

$(25-3)/2+7$

运算常常是字面量和运算符所进行的操作。



# 第 3 章

## 函数



# 第 4 章

## 面向对象



# 第二部分

## 进阶篇





# 第 5 章

## 泛型

有的时候我们想要制作一种基于一个类的类，例如，一个整数数组。



# 第 6 章

## 面向集合



# 第 7 章

## 数据结构



# 第三部分

## 提高篇





# 第 8 章

## 算法



# 第 9 章

## 自订模块



# 附录



# 附录 A

## 索引

import 命令, 4

Main, 3

函数, 3

字面量, 5

控制台输入/输出流, 4

程序的起点, 3

空语句, 4

表达式, 5

语句, 3, 5

输入/输出流, 4

运算, 5

运算符, 5