

欢迎来到Java世界，正式开干



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌

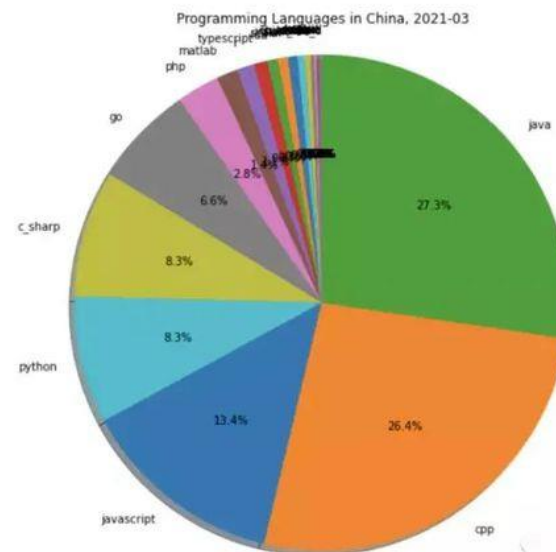
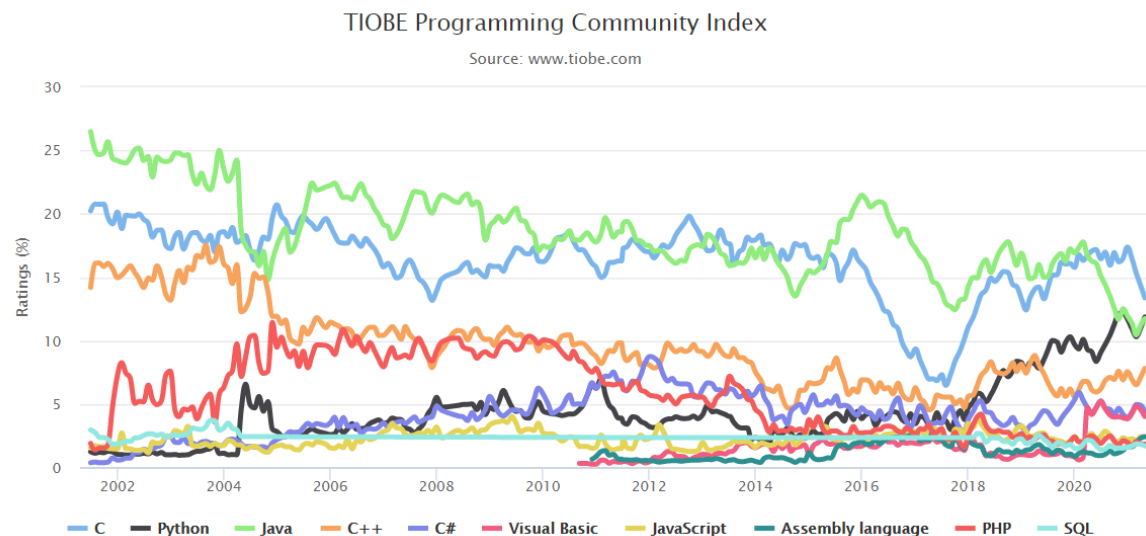
Java是什么

- Java是美国 Sun 公司 (Stanford University Network) 在1995年推出的计算机编程语言，后被Oracle公司收购。
- Java 早期称为Oak(橡树)，后期改名为Java。
- Java 之父：詹姆斯·高斯林 (James Gosling) 。

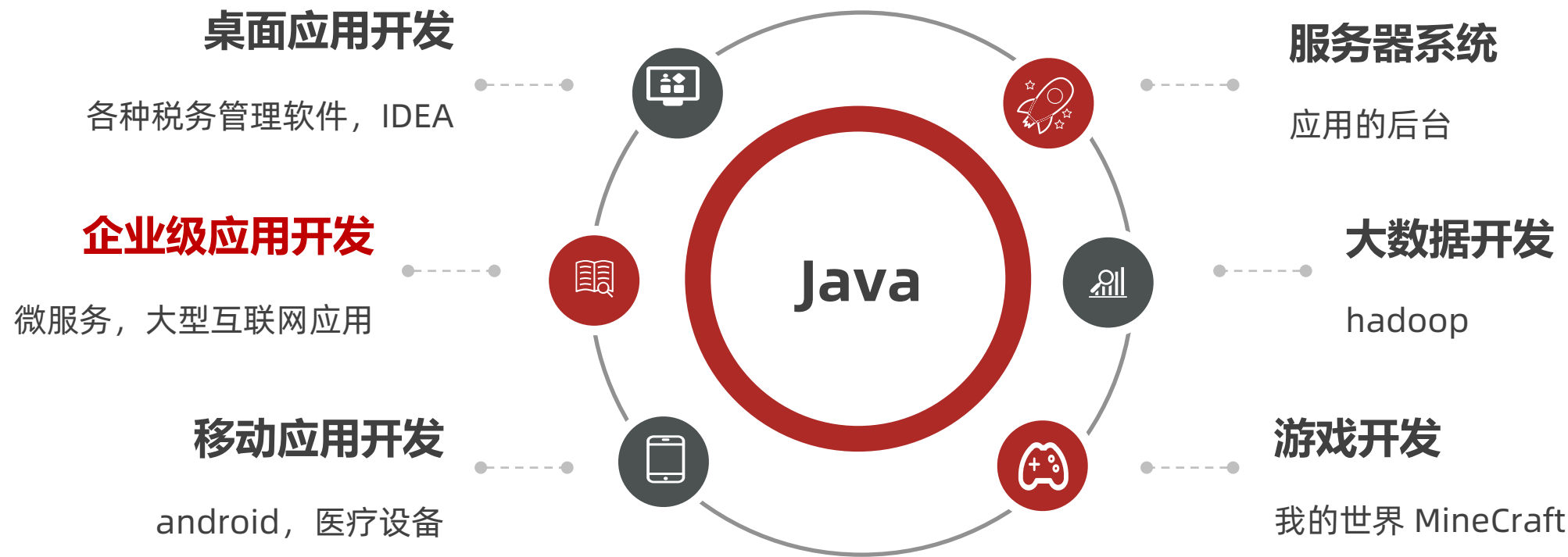


为什么用Java

- 世界上最流行的编程语言之一，在国内使用最为广泛的编程语言。
- 可移植性、安全可靠、性能较好。
- 开发社区最完善，功能最为丰富。



Java能做什么



Java能做什么





扎扎实实做好教学
以工匠精神投入课研

传智教育十五周年

Java技术体系平台

技术平台	说明
Java SE(Java Standard Edition) 标准版	Java技术的核心和基础，是学习Java EE，JavaME的基础，也是我们近期重点学习的技术。支持开发桌面级应用（如Windows下的应用程序）的Java平台
Java EE(Java Enterprise Edition)企业版	为开发企业环境下的应用程序提供的一套解决方案，主要针对于Web应用程序开发，多用于大型网站开发，是我们后期重点学习的技术。
Java ME(Java Micro Edition)小型版	是为机顶盒、移动电话和PDA之类嵌入式消费电子设备提供的Java语言平台，现在移动终端基本上都是使用Android和IOS的技术平台了。
Java Card	支持一些Java小程序（Applets）运行在小内存设备（如智能卡）上的平台，此技术也被广泛运用在SIM卡、提款卡上。



目录

Contents

➤ Java快速入门

- ◆ Java开发环境的准备：JDK简介、安装、常用命令

- ◆ Java入门程序-HelloWorld

- ◆ HelloWorld程序常见问题

- ◆ 补充知识：Java程序的执行原理

- ◆ 补充知识：JDK、JRE、跨平台原理

- ◆ 补充知识：Path环境变量的注意事项

➤ IDEA开发工具的使用

➤ Java基础语法

如何使用Java

- Java语言的产品是 **JDK (Java Development Kit : Java开发者工具包)**，必须安装JDK才能使用Java语言。

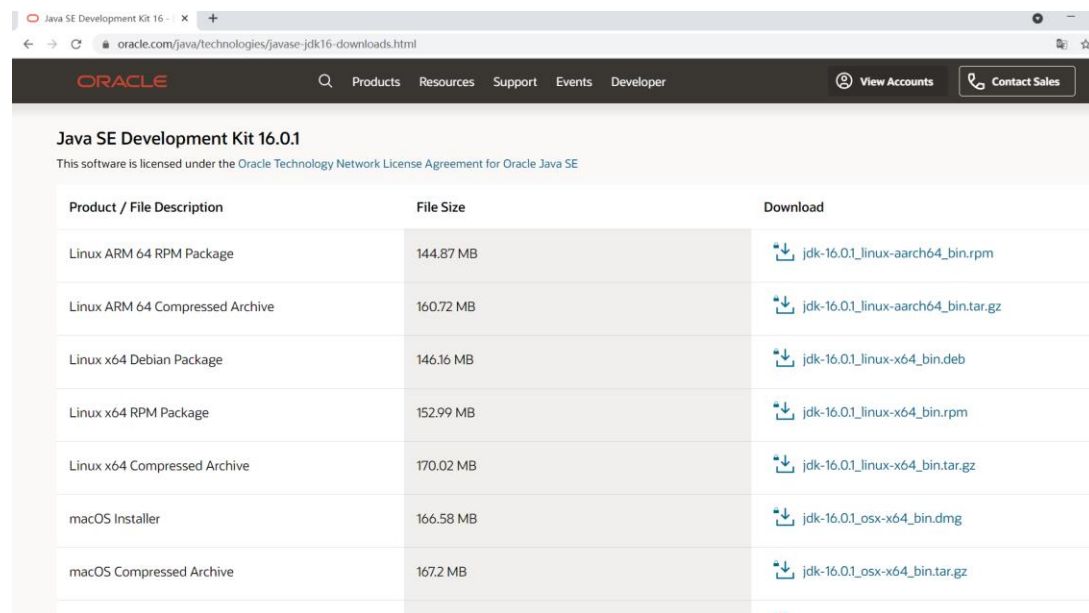
JDK产品的发展史

- | | | | | |
|----------------------------|---------------------|------------------|------------------|--|
| ● 1995年Java语言诞生 (Sun公司) | 1996年JDK (1.0) | | | |
| ● 1997年JDK(1.1) | 1998年JDK(1.2) | 2000年JDK(1.3) | 2002年(1.4) | |
| ● 2004年JDK(5.0) | 2006年JDK(6.0) | | | |
| ● 2009年JDK(Oracle甲骨文收购sun) | 2011年JDK(7.0) | | | |
| ● 2014年JDK(8.0) | 2017年9月JDK(9.0) | 2018年3月JDK(10.0) | 2018年9月JDK(11.0) | |
| ● 2019年3月JDK(12.0) | 2019年9月JDK(13.0) | | | |
| ● 2020年3月JDK(14.0) | 2020年9月15 JDK(15.0) | | | |
| ● 2021年3月16日JDK(16.0) | 2021年9月15日JDK(17.0) | | | |

如何获取JDK

- 去Oracle官网下载JDK。

<https://www.oracle.com/java/technologies/javase-jdk16-downloads.html>



The screenshot shows the Oracle website's download page for Java SE Development Kit 16.0.1. The page includes a navigation bar with the Oracle logo and links to Products, Resources, Support, Events, and Developer. Below the navigation bar, the title 'Java SE Development Kit 16.0.1' is displayed, followed by a license notice. A table lists the available download packages for different operating systems and architectures, including Linux ARM 64, Linux x64, and macOS, with their respective file sizes and download links.

Product / File Description	File Size	Download
Linux ARM 64 RPM Package	144.87 MB	jdk-16.0.1_linux-aarch64_bin.rpm
Linux ARM 64 Compressed Archive	160.72 MB	jdk-16.0.1_linux-aarch64_bin.tar.gz
Linux x64 Debian Package	146.16 MB	jdk-16.0.1_linux-x64_bin.deb
Linux x64 RPM Package	152.99 MB	jdk-16.0.1_linux-x64_bin.rpm
Linux x64 Compressed Archive	170.02 MB	jdk-16.0.1_linux-x64_bin.tar.gz
macOS Installer	166.58 MB	jdk-16.0.1_osx-x64_bin.dmg
macOS Compressed Archive	167.2 MB	jdk-16.0.1_osx-x64_bin.tar.gz

JDK提供了编译和执行Java程序的工具

此电脑 > 本地磁盘 (D:) > soft > Java > jdk-16.0.1 > bin		
名称		类型
jaccesswalker.exe		应用程序
jar.exe		应用程序
jarsigner.exe		应用程序
java.dll		应用程序扩展
java.exe	2、执行工具	应用程序
javaaccessbridge.dll		应用程序扩展
javac.exe	1、编译工具	应用程序
javadoc.exe		应用程序
javajpeg.dll		应用程序扩展
java.exe		应用程序

命令行窗口

按下 **Win + R**，在运行输入框中，输入 **cmd**，敲回车。



如何验证能用javac和java

- 在命令行 窗口中直接输入javac -version 及 java -version分别看版本提示，如果版本提示都是自己安装
的版本号，则代表JDK环境搭建成功。

```
D:\code>javac -version
javac 16.0.1

D:\code>java -version
java version "16.0.1" 2021-04-20
Java(TM) SE Runtime Environment (build 16.0.1+9-24)
Java HotSpot(TM) 64-Bit Server VM (build 16.0.1+9-24, mixed mode, sharing)
```

其他常用命令行命令

常用命令	作用
D: C:	切换到某个盘下
dir	查看当前路径下的全部内容
cd 目录	进入单级目录: cd itheima
cd 盘符:\目录1\目录2\...	进入多级目录: cd C:\itheima\JavaSE\第一天
cd ..	回退到上一级目录
cd \	回退到盘符目录
cls	清屏
exit	退出命令提示符窗口



总结

1. 搭建Java的开发环境需要安装什么,去哪儿下载啊?

- **JDK (Java Development Kit) 开发工具包; Oracle官网。**

2. Java开发环境最重要的2个命令是什么啊?

- **javac 编译命令、java 执行命令。**

3. 想直接访问这2个命令, 得在哪个窗口下访问?

- **命令行窗口。**

4. 命令行窗口中常用的命令有哪些?

- **切换盘符、dir、cd、cd.. 、cd 路径 、cls**



目录

Contents

➤ Java快速入门

- ◆ JDK介绍、安装

- ◆ Java入门程序-HelloWorld

- ◆ HelloWorld程序常见问题

- ◆ 补充知识：Java程序的执行原理

- ◆ 补充知识：JDK、JRE、跨平台原理

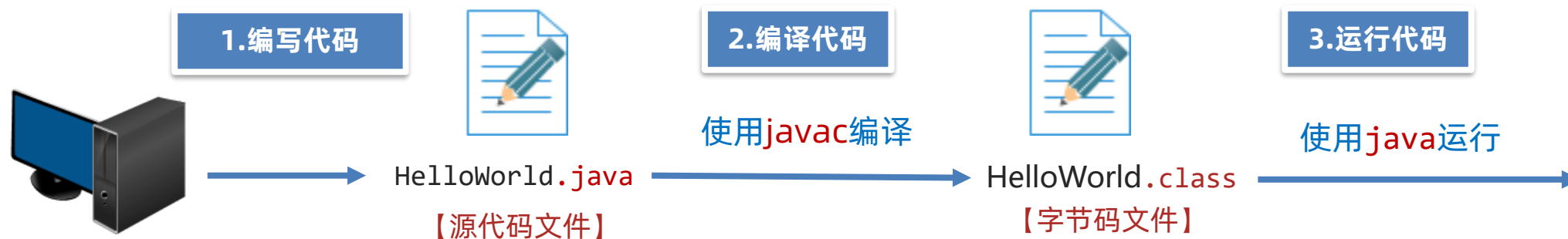
- ◆ 补充知识：Path环境变量的注意事项

➤ IDEA开发工具的使用

➤ Java基础语法

Java 程序开发的三个步骤

- 开发 Java 程序，需要三个步骤：编写代码，编译代码，运行代码



注意事项：

- 第一个 Java 程序建议使用记事本书写。
- 建议代码文件名全英文，首字母大写，满足驼峰模式，源代码文件的后缀必须是 `.java`。

编写代码

- 第一个程序的代码如下：



```
public class HelloWorld {  
    public static void main(String[] args){  
        System.out.println("Hello World");  
    }  
}
```

The screenshot shows a Notepad window titled "HelloWorld.java - 记事本". The menu bar includes "文件(E)", "编辑(E)", "格式(O)", "查看(V)", and "帮助(H)". The code is written in a standard font. The status bar at the bottom indicates "第 5 行, 第 2 列", "100%", "Windows (CRLF)", and "UTF-8".

注意：文件名称必须与代码中的类名称一致。

保存文件：ctrl + s

编译代码、运行代码

- ① 编译: javac 文件名.java

范例: **javac HelloWorld.java**

- ② 运行: java 类名

范例: **java HelloWorld**

```
D:\code>javac HelloWorld.java  
D:\code>java HelloWorld  
Hello World
```

HellWorld程序介绍

class: 定义一个类，后面跟上的HelloWorld是**类名称**

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("HelloWorld");  
    }  
}
```

程序执行时的**入口点**，main方法也称之为**主方法**

打印语句，使程序在**控制台打印双引号所包裹的内容**



总结

1、开发一个Java程序要经历哪些步骤?

- 编写、编译(javac)、运行(java)

2、Java代码编写有什么基本要求?

- 文件名称的后缀必须是java结尾。
- 文件名称必须与代码的类名称一致。
- 必须使用英文模式下的符号。



Java入门程序练习

🕒 10 分钟

需求：完成入门程序的编写、编译、运行。

```
D:\code>javac HelloWorld.java  
D:\code>java HelloWorld  
Hello World
```



目录

Contents

➤ Java快速入门

- ◆ JDK介绍、安装

- ◆ Java入门程序-HelloWorld

- ◆ HelloWorld程序常见问题

- ◆ 补充知识：Java程序的执行原理

- ◆ 补充知识：JDK、JRE、跨平台原理

- ◆ 补充知识：Path环境变量的注意事项

➤ IDEA开发工具的使用

➤ Java基础语法

HelloWorld案例常见错误

1. 扩展名被隐藏。
2. 文件名和类名不一致。
3. 大小写错误，单词拼写错误，存在中文符号，找不到main方法。
4. 括号不配对。
5. 编译或者执行命令使用不当。

HelloWorld案例常见错误

- 扩展名被隐藏



HelloWorld案例常见错误

- 文件名和类名不一致



```
D:\code>javac HelloWorld.java  
HelloWorld.java:1: 错误: 类 HelloWorld 是公共的, 应在名为 HelloWorld.java 的文件中声明  
public class HelloWorld {  
1 个错误  
D:\code>
```


HelloWorld案例常见错误

- 大小写错误，单词拼写错误，存在中文符号，找不到main方法



```
Public class HelloWorld {  
    public static void mian(String[] args){  
        System.out.println("Hello World");  
    }  
}
```

```
D:\code>javac HelloWorld.java  
HelloWorld.java:3: 错误: 编码 GBK 的不可映射字符 (0x9B)  
        System.out.println("Hello World")  
HelloWorld.java:1: 错误: 需要 class、interface、enum 或 record  
Public class HelloWorld {  
HelloWorld.java:3: 错误: 需要';'  
        System.out.println("Hello World")
```

HelloWorld案例常见错误

- 括号不配对



```
HelloWorld.java - 记事本
文件(E) 编辑(E) 格式(O) 查看(V) 帮助(H)
public class HelloWorld {
    public static void main(String[] args){
        System.out.println("Hello World");
}

```

第 4 行, 第 2 列 100% Windows (CRLF) UTF-8

```
D:\code>javac HelloWorld.java
HelloWorld.java:5: 错误: 进行语法分析时已到达文件结尾
}
1 个错误
D:\code>_
```

HelloWorld案例常见错误

- 编译、执行使用不当。

```
D:\code>javac HelloWorld.java
```

```
D:\code>java HelloWorld.class
```

```
错误: 找不到或无法加载主类 HelloWorld.class
```

```
原因: java.lang.ClassNotFoundException: HelloWorld.class
```

```
D:\code>java HelloWorld.java  
Hello World
```



**JDK 11开始支持java直接运行
源代码文件**

```
D:\code>_
```



目录

Contents

➤ Java快速入门

- ◆ JDK介绍、安装

- ◆ Java入门程序-HelloWorld

- ◆ HelloWorld程序常见问题

- ◆ 补充知识：Java程序的执行原理

- ◆ 补充知识：JDK、JRE、跨平台原理

- ◆ 补充知识：Path环境变量的注意事项

➤ IDEA开发工具的使用

➤ Java基础语法

Java是什么？

- Java是一种高级编程语言。

什么是高级编程语言呢？

- 就是编写的一些接近人类语言的指令，告诉机器该怎么做。

```
public class HelloWorld {  
    public static void main(String[] args){  
        System.out.println("Hello World!");  
    }  
}
```



计算机怎么理解Java写好的这些指令的?

```
public class HelloWorld {  
    public static void main(String[] args){  
        System.out.println("Hello World!");  
    }  
}
```

javac编译器

HelloWorld.class

Java执行工具（解释）

计算机可以理解的指令(机器指令)

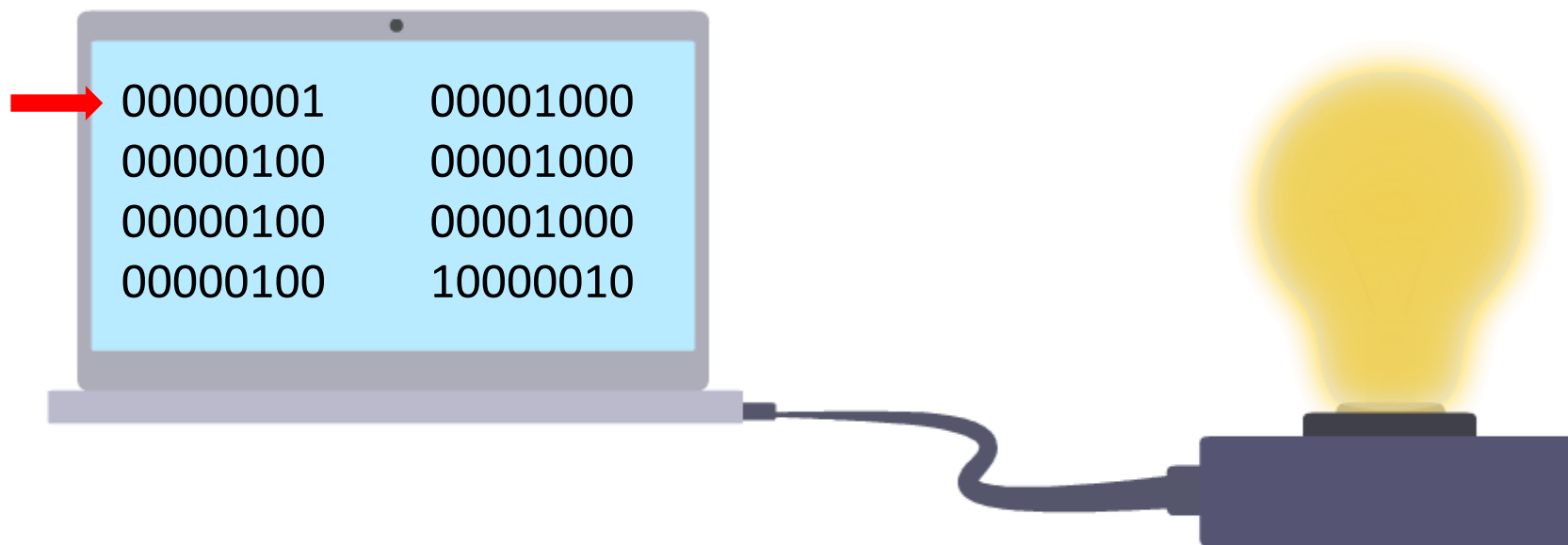
驱动机器干活

计算机能识别的指令是什么样的？

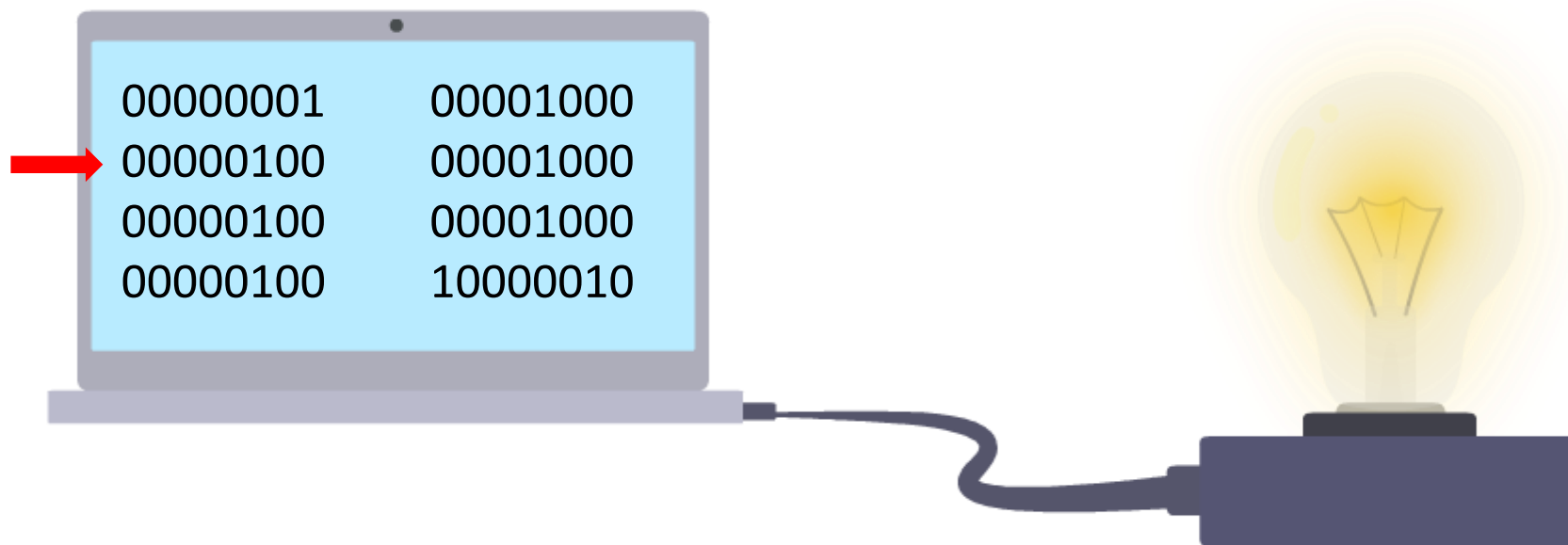
- 由于计算机底层都是硬件电路，可以理解成开关，开表示0，关表示1，机器指令就是一个一个的二进制串。
- 计算机中会内置一些机器指令，代表具备的功能。

机器指令	机器操作
00000000	停止程序
00000001	完全打开灯泡
00000010	完全关闭灯泡
00000100	把灯泡调暗 20%
00001000	把灯泡调亮 20%
10000001	跳转到位置1执行
10000010	跳转到位置2执行
...	...

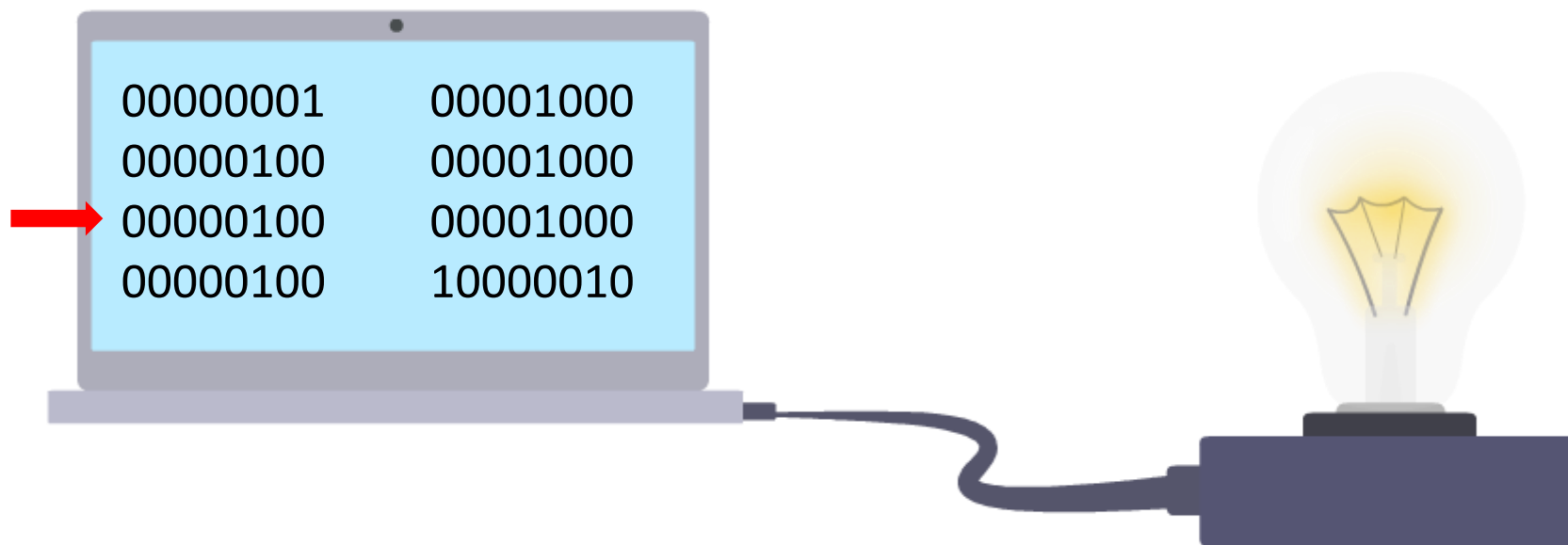
编程控制呼吸灯



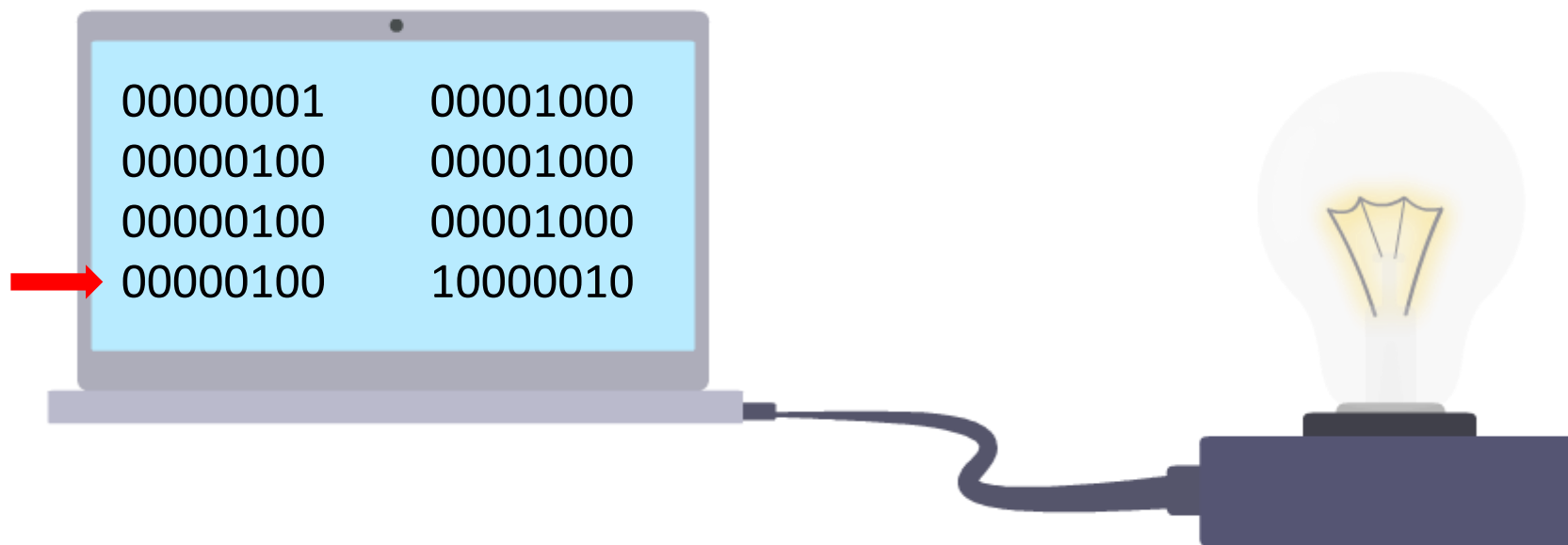
编程控制呼吸灯



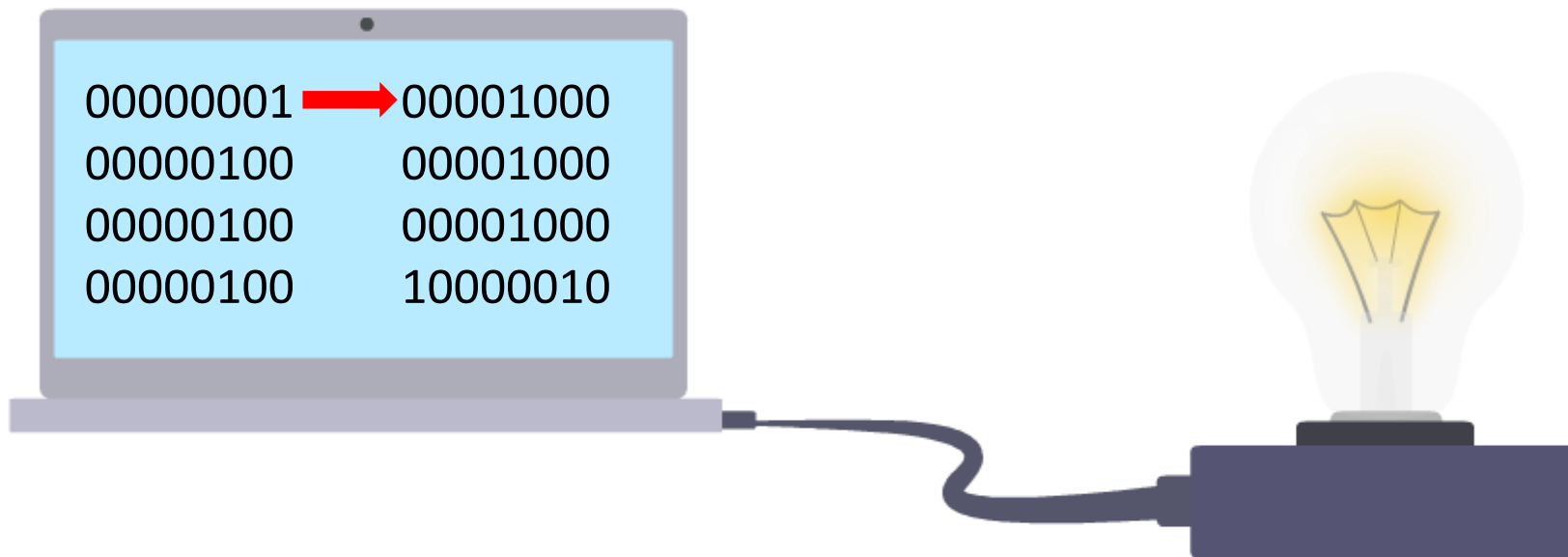
编程控制呼吸灯



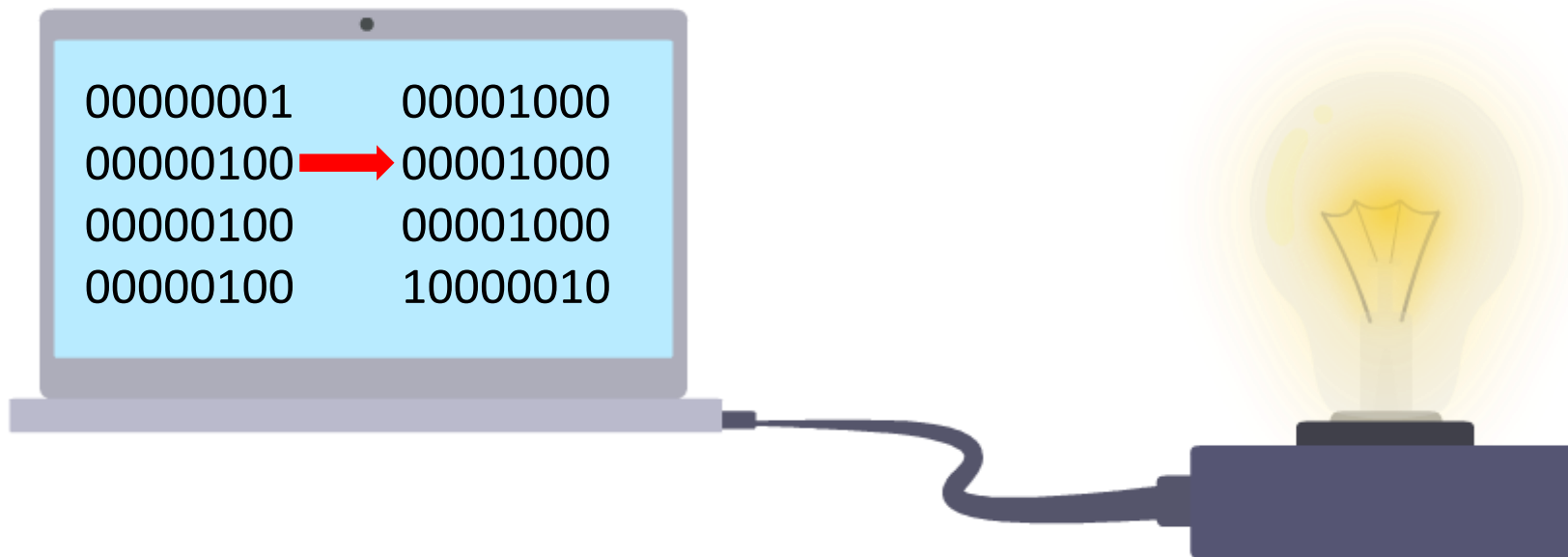
编程控制呼吸灯



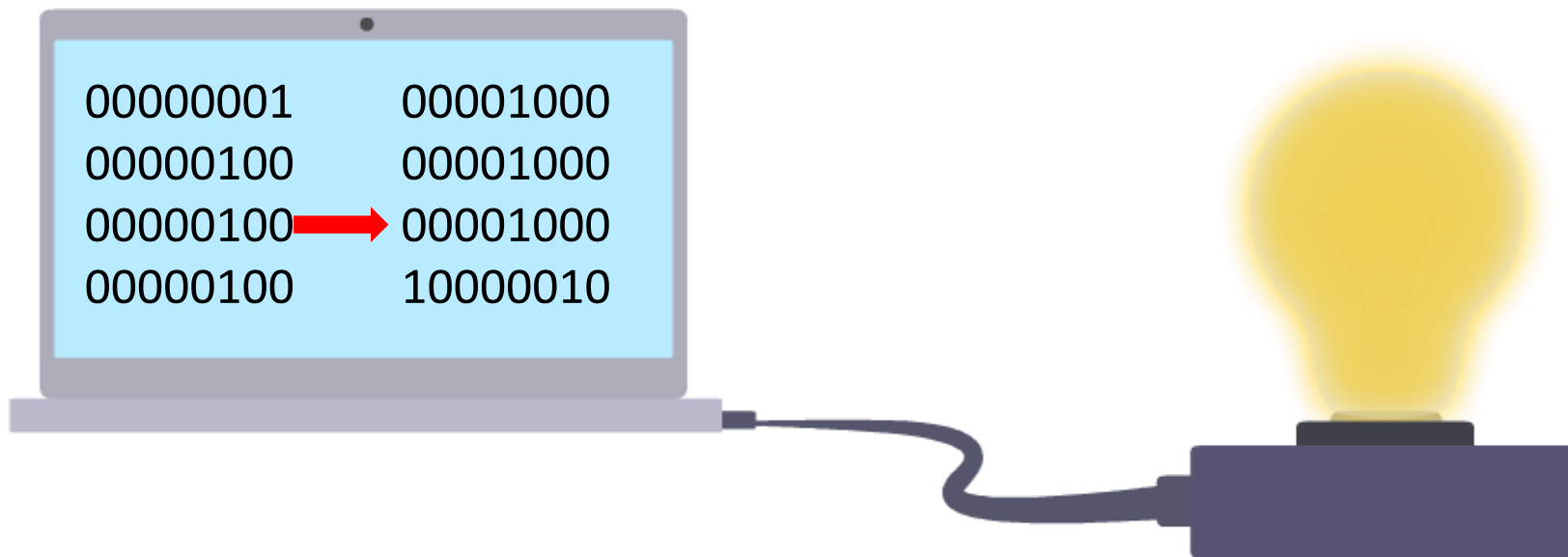
编程控制呼吸灯



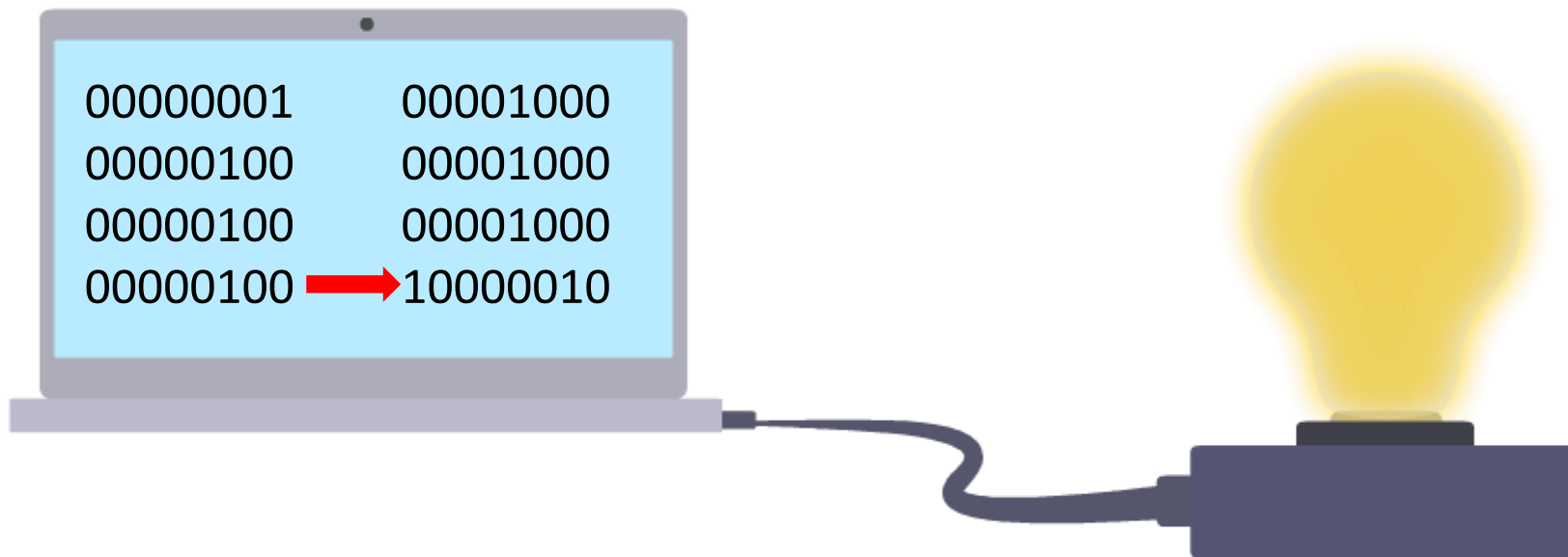
编程控制呼吸灯



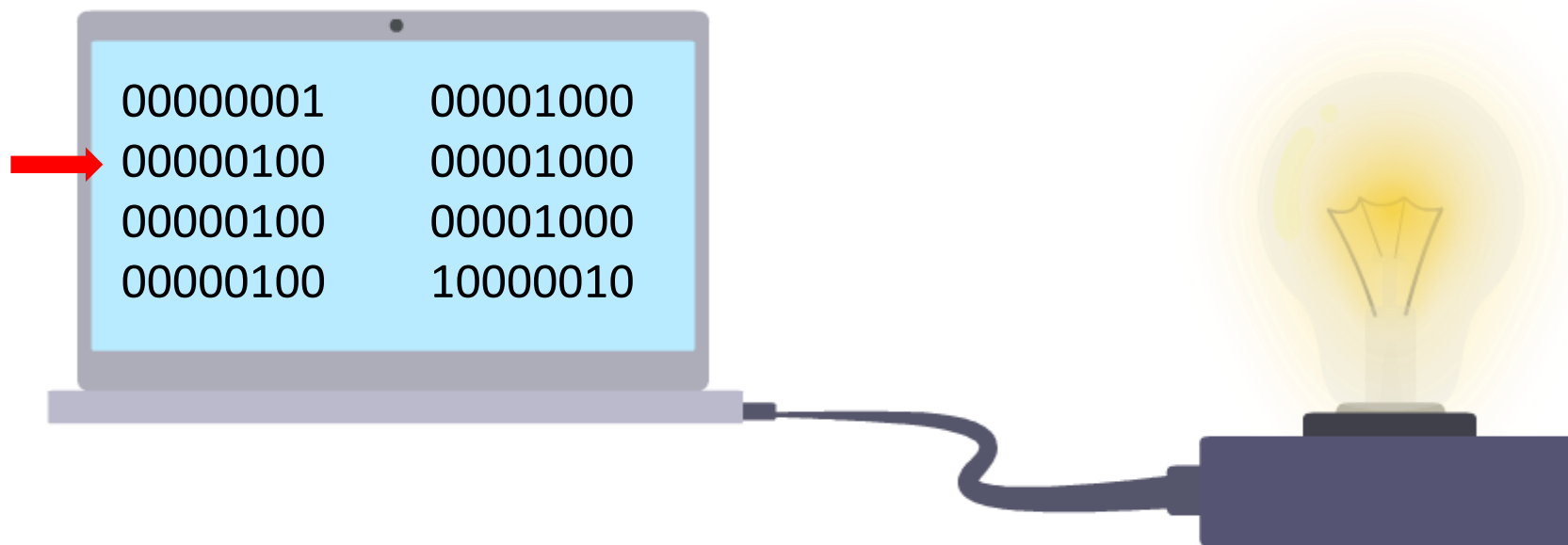
编程控制呼吸灯



编程控制呼吸灯



编程控制呼吸灯



早期的编程形式是什么样的?



编程语言发展历程

- 机器语言
- 汇编语言
- 高级语言

为什么学习高级编程语言?

- 更简单：使用人类自己的语言书写，翻译器再将其翻译成计算机能理解的指令。



总结

1. Java语言最终是如何去控制计算机执行的?

- Java语言存在编译器和执行器
- 最终可以编译、解释成计算机可以识别的机器指令。



目录

Contents

➤ Java快速入门

- ◆ JDK介绍、安装

- ◆ Java入门程序-HelloWorld

- ◆ HelloWorld程序常见问题

- ◆ 补充知识：Java程序的执行原理

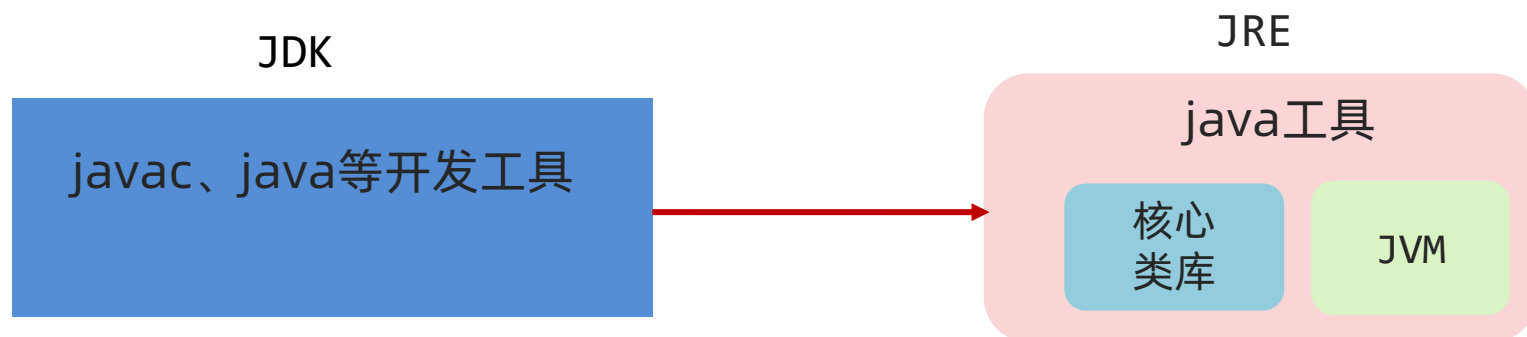
- ◆ 补充知识：JDK、JRE、跨平台原理

- ◆ 补充知识：Path环境变量的注意事项

➤ IDEA开发工具的使用

➤ Java基础语法

JDK、JRE的关系



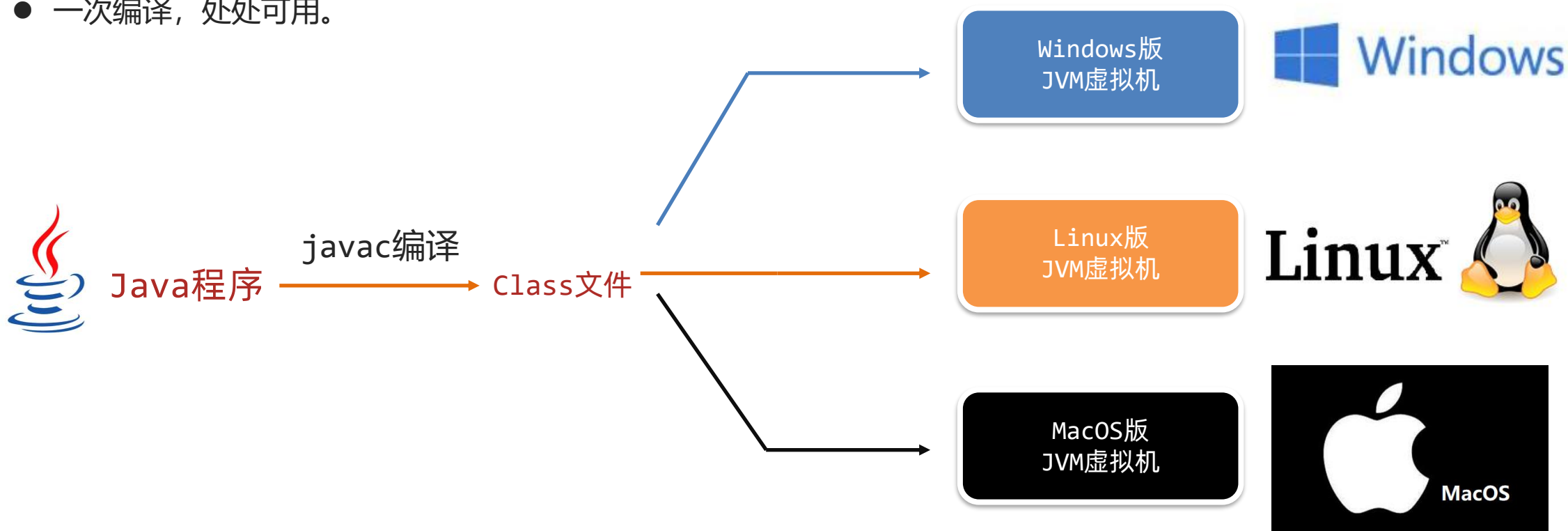
- JVM (Java Virtual Machine) : Java虚拟机, 真正运行Java程序的地方。
- JDK (Java Development Kit) : Java开发工具, 包含java编译器、java虚拟机、以及java帮我们写好的程序...等所有需要的东西。
- JRE (Java Runtime Environment) : Java运行时环境, 只包含java虚拟机、以及java帮我们写好的程序等java运行所需的東西。

结论:

- 我们现阶段只要安装了JDK即可拥有java程序开发和运行需要的全部环境。
- 如果是在企业的部署环境下, 其实只需要安装JRE就可以运行编译好的Java程序了。

跨平台工作原理

- 一次编译，处处可用。





总结

1. JDK和JRE的关系是什么样的?

- JDK中会关联JRE。
- 安装JDK时包含了: java、javac、jvm和核心类库等全部java开发、运行的环境。
- JRE中只包含了: java、jvm和核心类库等Java的运行环境。

2. Java的跨平台是什么含义，如何实现的?

- 一次编译、处处可用
- 不同平台有对应的JRE环境。



目录

Contents

➤ Java快速入门

- ◆ JDK介绍、安装
- ◆ Java入门程序-HelloWorld
- ◆ HelloWorld程序常见问题
- ◆ 补充知识：Java程序的执行原理
- ◆ 补充知识：JDK、JRE、跨平台原理
- ◆ 补充知识：JDK安装时Path环境变量的注意

➤ IDEA开发工具的使用

➤ Java基础语法

Path环境变量的作用

- Path环境变量是用来配置命令的路径的，这样可以在命令行窗口中的任何位置直接使用这些命令。

JDK安装时Path需要注意的事项：

- 目前较新的JDK安装时会自动配置javac、java命令的路径到Path环境变量中去，所以javac、java可以直接使用。
- 但是以前下载的老版本的JDK是没有自动配置的，此时必需要自己配置Path环境变量。

环境变量的位置在：我的电脑 -> 属性 -> 高级系统设置 -> 高级 -> 环境变量。



建议还是自己配置环境变量

- ① **JAVA_HOME**: 告诉操作系统JDK安装在了哪个位置 (未来其他技术要通过这个找JDK)

```
JAVA_HOME C:\soft\java\jdk-17
```

- ② **Path**: 告诉操作系统JDK提供的javac(编译)、java(执行)命令安装到了哪个位置

```
Path C:\soft\java\jdk-17\bin
```

或者

```
Path %JAVA_HOME%\bin
```

注意：新版本的JDK只是自动配置了Path，没有自动配置JAVA_HOME。

如何检测是否配置成功

- 打开cmd，输入javac -version 及 java -version分别看版本提示。
- 如果版本提示都是自己安装版本号，则代表JDK环境搭建成功。

```
C:\Users\itheima>java -version
java version "17" 2021-09-14 LTS
Java(TM) SE Runtime Environment (build 17+35-LTS-2724)
Java HotSpot(TM) 64-Bit Server VM (build 17+35-LTS-2724, mixed mode, sharing)

C:\Users\itheima>javac -version
javac 17
```



总结

1. 什么是Path环境变量?

- 在操作系统中配置的命令路径，实现在系统中直接访问命令。

2. JDK安装时，环境变量需要注意什么?

- 较新版本的JDK会自动配置PATH环境变量。
- 建议自己配置“JAVA_HOME”、“Path”



目录

Contents

- **Java快速入门**
- **IDEA开发工具的使用**
 - ◆ IntelliJ IDEA 开发工具概述、安装
 - ◆ IDEA 编写第一个程序
 - ◆ IDEA 字体、主题、背景色设置
 - ◆ IDEA 常用快捷键介绍
 - ◆ IDEA 工程、模块常用操作
- **Java基础语法**

之前的开发工具

- 文本编辑工具：记事本 、 UltraEdit 、 EditPlus 、 TextPad 、 NotePad++、 sublime..
- 存在的问题：编写代码时没有错误提醒、没有智能提示、需要自己进行编译、执行。功能不够强大。

集成开发环境（ IDE， Integrated Development Environment ）

- 把代码编写，编译，执行等多种功能综合到一起的开发工具，可以进行代码智能提示，错误提醒，项目管理等等。
- 常见Java的IDE工具有：Eclipse、MyEclipse、IntelliJ IDEA、Jbuilder、NetBeans等。

IntelliJ IDEA开发工具

- **IntelliJ IDEA** 一般简称IDEA，是Java编程语言开发的集成环境，在业界被公认为最好的java开发工具。
- IDEA 在代码错误提醒，智能代码补全等多方面表现的都非常优秀，是Java开发企业级首选工具。



IDEA 下载

下载: <https://www.jetbrains.com/idea/>

安装: 傻瓜式安装, 建议修改安装路径 (不要安装在有空格和中文的路径下)。





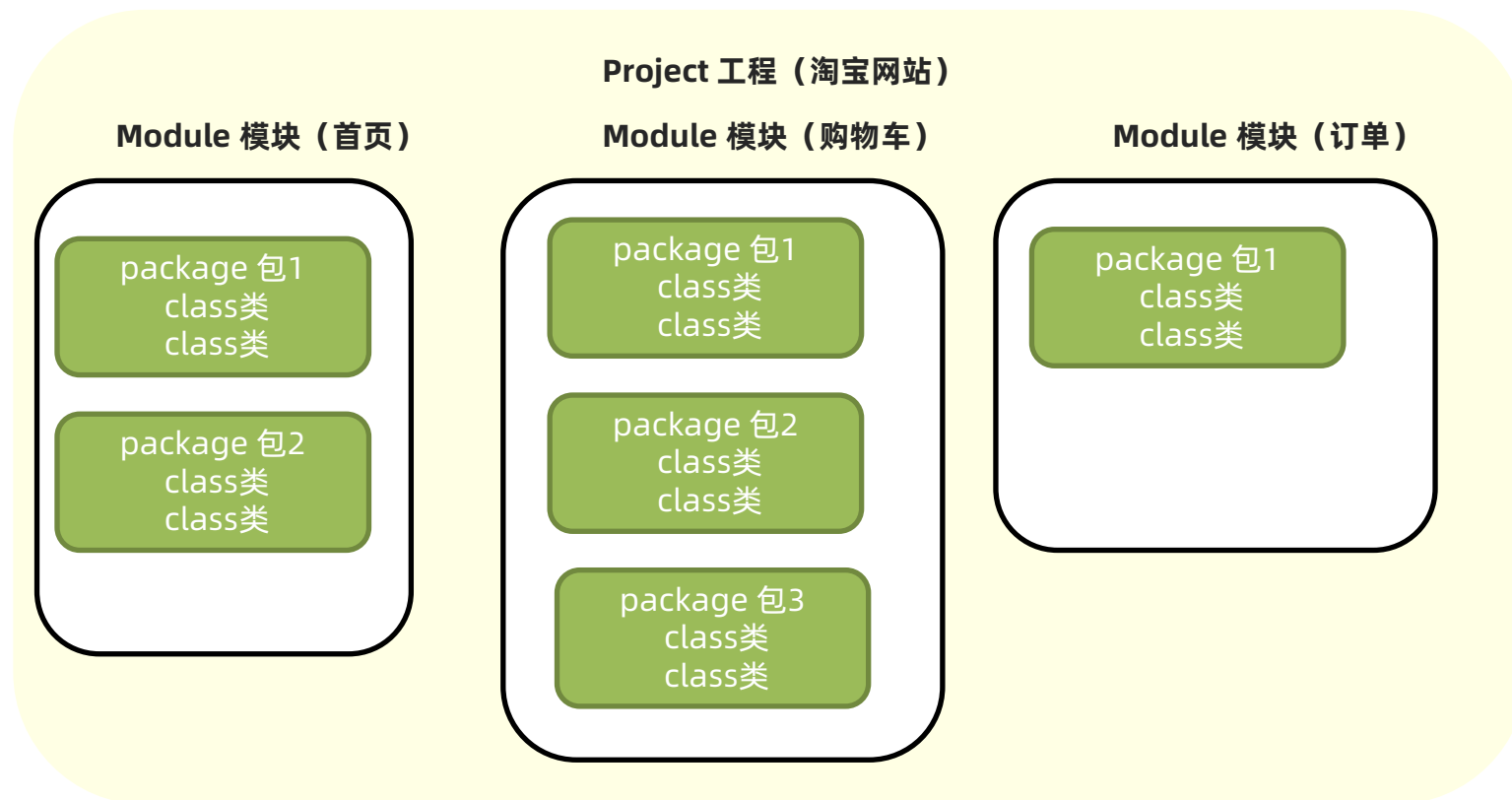
目录

Contents

- **Java快速入门**
- **IDEA开发工具的使用**
 - ◆ IntelliJ IDEA 开发工具概述、安装
 - ◆ **IDEA 编写第一个程序**
 - ◆ IDEA 字体、主题、背景色设置
 - ◆ IDEA常用快捷键介绍
 - ◆ IDEA工程、模块常用操作
- **Java基础语法**

IDEA项目结构介绍

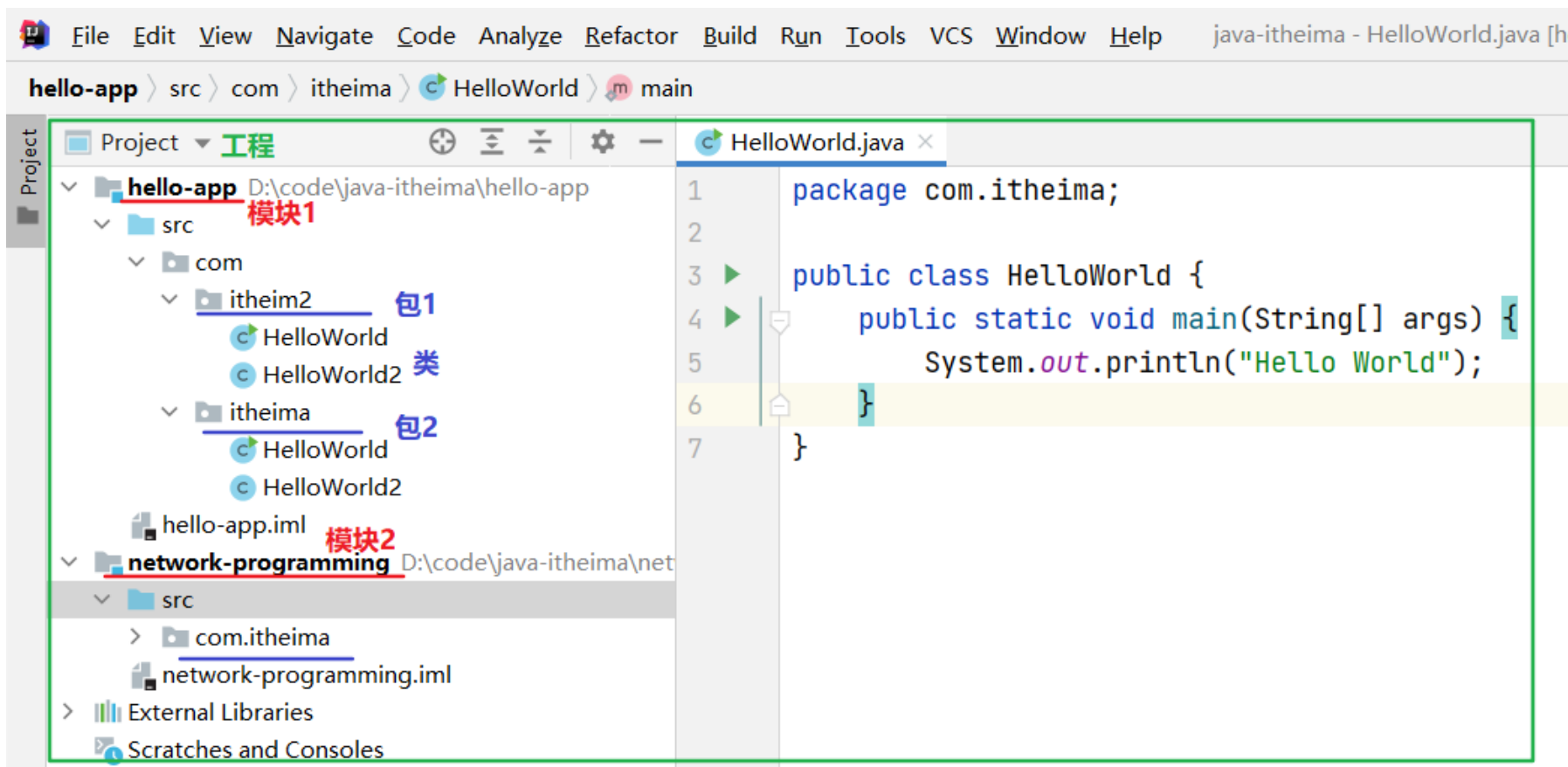
- project (项目、工程)
- module (模块)
- package (包)
- class (类)



结论：这些结构的划分，是为了方便管理项目技术和代码文件的

使用步骤:

- ① 创建Project空工程
- ② 创建Module模块
- ③ 创建Package包
- ④ 创建class 类
- ⑤ 在类中编写代码
- ⑥ 完成编译运行





总结

1. IDEA的结构都是什么样的?

- **project - module - package - class**
- **project中可以创建多个module**
- **module中可以创建多个package**
- **package中可以创建多个class**

2. 新建都是哪些操作

- **New project/module/package/class**

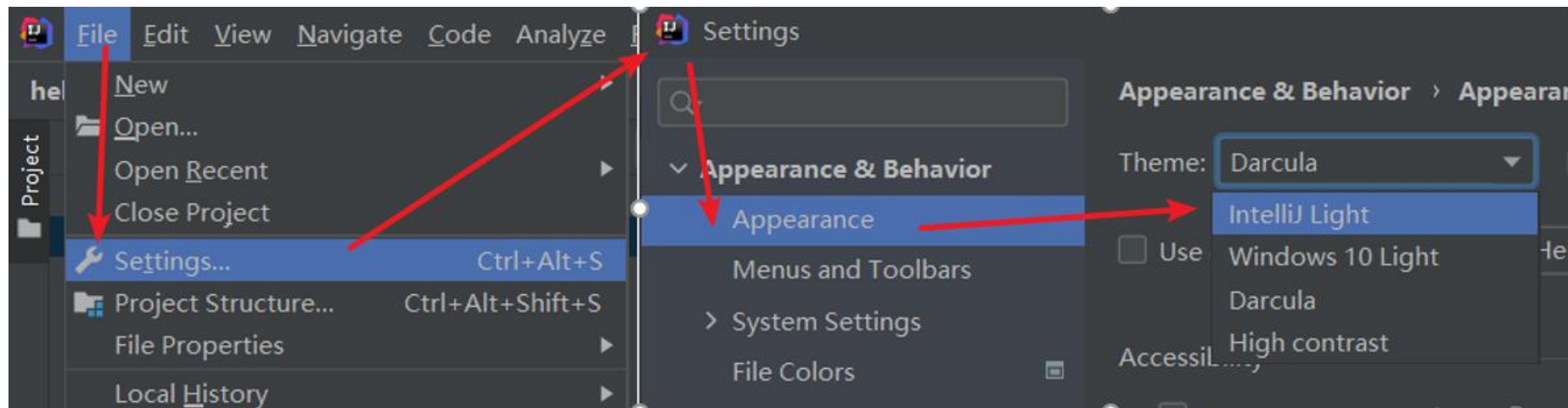


目录

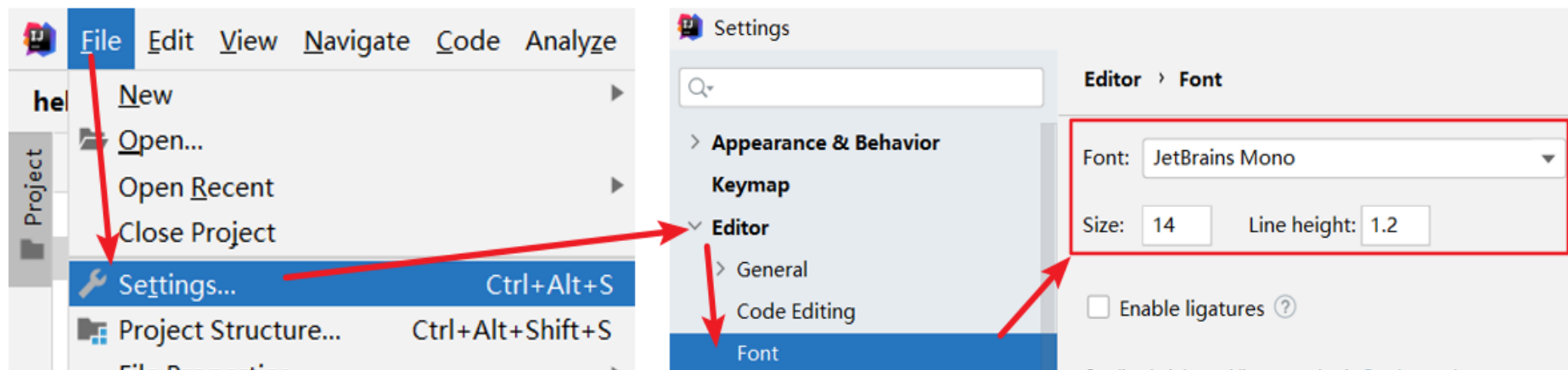
Contents

- **Java快速入门**
- **IDEA开发工具的使用**
 - ◆ IntelliJ IDEA 开发工具概述、安装
 - ◆ IDEA 编写第一个程序
 - ◆ **IDEA 字体、主题、背景色设置**
 - ◆ IDEA常用快捷键介绍
 - ◆ IDEA工程、模块常用操作
- **Java基础语法**

IDEA 中设置主题、字体



主题配置



字体配置



目录

Contents

- **Java快速入门**
- **IDEA开发工具的使用**
 - ◆ IntelliJ IDEA 开发工具概述、安装
 - ◆ IDEA 编写第一个程序
 - ◆ IDEA 字体、主题、背景色设置
 - ◆ **IDEA常用快捷键介绍**
 - ◆ IDEA类、模块、工程常用操作
- **Java基础语法**

IDEA 常用快捷键

快捷键	功能效果
main/psvm、sout、...	快速键入相关代码
Ctrl + D	复制当前行数据到下一行
Ctrl + Y	删除所在行，建议用Ctrl + X
Ctrl + ALT + L	格式化代码
ALT + SHIFT + ↑ , ALT + SHIFT + ↓	上下移动当前代码
Ctrl + / , Ctrl + Shift + /	对代码进行注释(讲注释的时候再说)



目录

Contents

- **Java快速入门**
- **IDEA开发工具的使用**
 - ◆ IntelliJ IDEA 开发工具概述、安装
 - ◆ IDEA 编写第一个程序
 - ◆ IDEA 字体、主题、背景色设置
 - ◆ IDEA常用快捷键介绍
 - ◆ **IDEA工程、模块常用操作**
- **Java基础语法**

IDEA 中其他操作

- 删除类文件
- 修改类名称
- 修改模块
- 导入模块
- 删除模块
- 打开工程
- 关闭工程



目录

Contents

- **Java快速入门**
- **IDEA开发工具的使用**
- **Java基础语法**
 - ◆ **注释**
 - ◆ 字面量
 - ◆ 变量
 - ◆ 变量使用注意事项
 - ◆ 数据的存储形式：二进制
 - ◆ 数据类型
 - ◆ ASCII编码
 - ◆ 关键字、标志符

什么是注释

- 在程序中对代码进行解释说明的文字，方便自己和其他人理解、查看，**不会影响程序的正常执行。**

注释有哪些

- 单行注释

```
// 注释内容，只能写一行
```

- 多行注释

```
/*  
    注释内容1  
    注释内容2  
*/
```

- 文档注释：

一般用在类和方法上的注释、以后会讲解区别。

```
/**  
    注释内容  
    注释内容  
*/
```

快捷键进行注释	功能效果
Ctrl + /	单行注释。
Ctrl + Shift + /	选中代码进行多行注释。

写注释是一个利人利己的好习惯



不是说
好人一生平安的吗



目录

Contents

- Java快速入门
- IDEA开发工具的使用
- Java基础语法
 - ◆ 注释
 - ◆ 字面量
 - ◆ 变量
 - ◆ 变量使用注意事项
 - ◆ 数据的存储形式：二进制
 - ◆ 数据类型
 - ◆ ASCII编码
 - ◆ 关键字、标志符

字面量

- 告诉程序员，数据在程序中的书写格式。

字面量分类

字面量分类	生活中的写法	程序中的写法	说明
整数	666, -88	666, -88	程序中的写法与生活中写法一致
小数	13.14, -5.21	13.14, -5.21	程序中的写法与生活中写法一致
字符	A, 0, 我	'A', '0', '我'	程序中必须使用单引号，有且仅能一个字符
字符串	黑马程序员	"HelloWorld", "黑马程序员"	程序中必须使用双引号，内容可有可无
布尔值	true、false	true 、 false	布尔值，表示真假，只有两个值：true, false
空值		值是：null	一个特殊的值，空值(后面会讲解作用，暂时不管)



总结

1. 字符、字符串的字面量的格式要求?

- 字符必须单引号围起来，有且仅能一个字符。
- 字符串必须用双引号围起来。
- 布尔类型只有2个值：false 、 true



目录

Contents

- Java快速入门
- IDEA开发工具的使用
- Java基础语法
 - ◆ 注释
 - ◆ 字面量
 - ◆ 变量
 - ◆ 变量使用注意事项
 - ◆ 数据的存储形式：二进制
 - ◆ 数据类型
 - ◆ ASCII编码
 - ◆ 关键字、标志符

什么是变量

- 变量就是用来存储一个数据的内存区域（可以理解成盒子），且里面存储的数据可以变化。

变量定义的格式

数据类型 变量名称 = 初始值;

存储的初始数据

取名称，首字母建议小写，有意义。

强制限制盒子中存储数据的形式。例如：int（整数类型）、double(小数类型)

```
double money = 6.0;
```

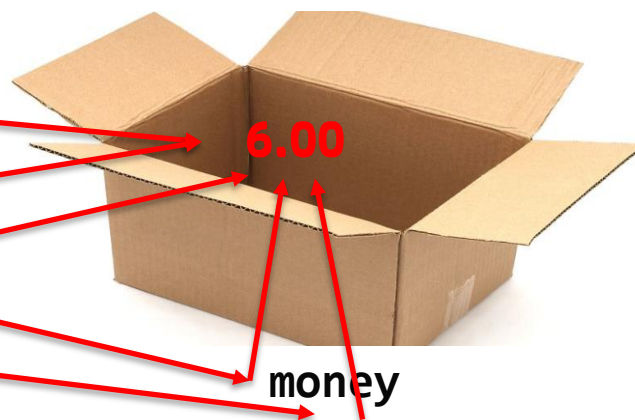


修改变量中存储的值

```
double money = 6.0;  
System.out.println(money); // 6.0
```

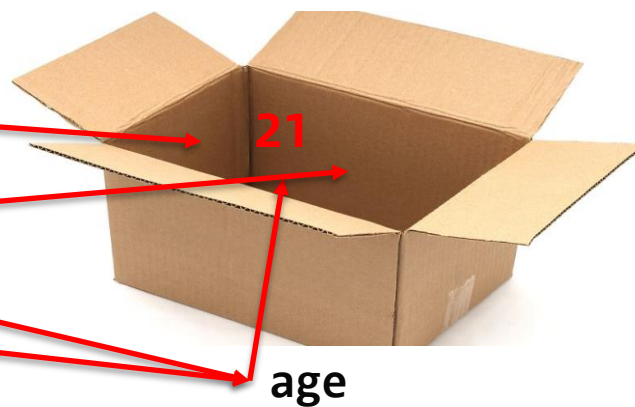
```
money = money + 4.0;  
System.out.println(money); // 10.0
```

+ 4.0 = 10.0



修改变量中存储的值

```
int age = 21;  
System.out.println(age); // 21  
  
age = 25;  
System.out.println(age); // 25
```





总结

1. 变量的作用?

- 存储一个数据的，可以改变里面存储的数据。

2. 变量的格式

- 数据类型 变量名称 = 初始值;



目录

Contents

- Java入门
- IDEA开发工具的使用
- Java语言
 - ◆ 注释
 - ◆ 字面量
 - ◆ 变量
 - ◆ 变量使用注意事项
 - ◆ 数据的存储形式：二进制
 - ◆ 数据类型
 - ◆ ASCII编码
 - ◆ 关键字、标志符

定义变量需要注意的几个问题

- 某个变量存储了其他类型的数据。

```
public static void main(String[] args) {  
    int age = 23;  
  
    double money = "有钱";  
}
```

- 变量的有效范围是从定义开始到 “}” 截止, 同一个范围内部不能定义2个同名的变量。

```
public static void main(String[] args) {  
    int age = 23;  
    System.out.println(age);  
    int age = 21;  
}
```

- 变量定义的时候可以没有初始值，但是使用的时候必须给初始值。

```
public static void main(String[] args) {  
    int age;  
    System.out.println(age);  
}
```




总结

1. 变量定义有哪些注意点?

- **什么类型变量存储什么类型的字面值。**
- 同一个范围变量名不能重复
- 变量在使用时要有初始值
- 变量存在访问范围



目录

Contents

- Java入门
- IDEA开发工具的使用
- Java语言
 - ◆ 注释
 - ◆ 字面量
 - ◆ 变量
 - ◆ 变量使用注意事项
 - ◆ 数据的存储形式：二进制
 - ◆ 数据类型
 - ◆ ASCII编码
 - ◆ 关键字、标志符

二进制

- 计算机底层都是一些数字电路(理解成开关)，用开表示0、关表示1，这些01的形式就是二进制。
- 数据在计算机底层都是采用二进制存储的。



二进制具体如何表示数据的

一个开关的状态



两个开关的状态



三个开关的状态



开关越多，能够表示的数据也越多。

二进制转换成十进制的运算方式

$$00 = 0 * 2^0 + 0 * 2^1 = 0$$

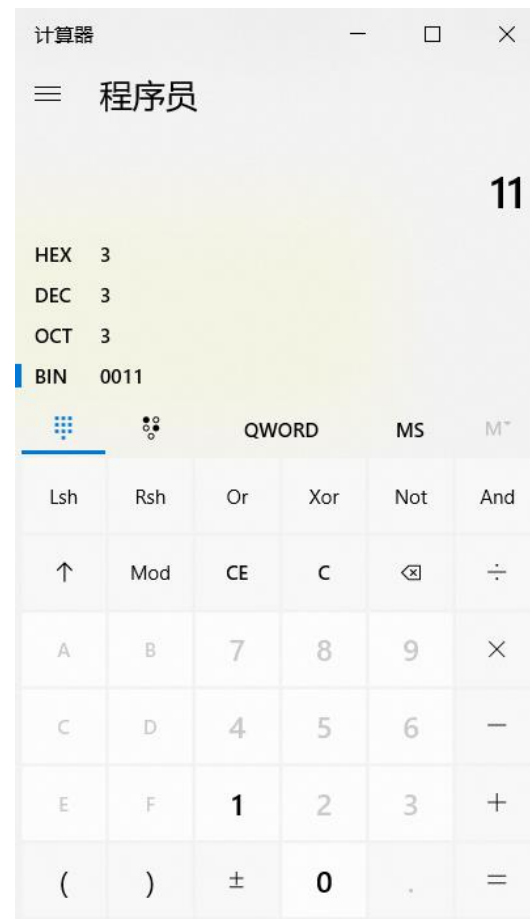
$$01 = 1 * 2^0 + 0 * 2^1 = 1$$

$$10 = 0 * 2^0 + 1 * 2^1 = 2$$

$$11 = 1 * 2^0 + 1 * 2^1 = 3$$

`int age1 = 3;` 实际上，数据在内存中存储的有效二进制位是：11

`int age2 = 19;` 实际上，数据在内存中存储的有效二进制位是：10011



计算机的数据单位

- 在计算机中认为一个开关表示的0|1称为1位（b），每8位称为一个字节（B），所以1B=8b。
- 字节是计算机中数据的最小单位。

$$1K = 1024$$

$$1KB = 1024B$$

$$1M = 1024 * 1024 = 1024K$$

$$1MB = 1024KB$$

$$1G = 1024 * 1024 * 1024 = 1024M$$

$$1GB = 1024MB$$

$$1T = 1024 * 1024 * 1024 * 1024 = 1024G$$

$$1TB = 1024GB$$



总结

1. 数据在计算机底层都是怎么存储的？

- 都是采用二进制：就是0、1表示的数据形式。

2. 数据在计算机中存储的最基本单位是什么？

- 字节

3. 数据单位换算是什么样的？

$$1\text{B} = 8\text{b}$$

$$1\text{K} = 1024$$

$$1\text{M} = 1024 * 1024 = 1024\text{K}$$

$$1\text{G} = 1024 * 1024 * 1024 = 1024\text{M}$$

$$1\text{T} = 1024 * 1024 * 1024 * 1024 = 1024\text{G}$$

$$1\text{KB} = 1024\text{B}$$

$$1\text{MB} = 1024\text{KB}$$

$$1\text{GB} = 1024\text{MB}$$

$$1\text{TB} = 1024\text{GB}$$



目录

Contents

- Java入门
- IDEA开发工具的使用
- Java语言
 - ◆ 注释
 - ◆ 字面量
 - ◆ 变量
 - ◆ 变量的注意事项
 - ◆ 数据的存储形式：二进制
 - ◆ 数据类型
 - ◆ ASCII编码表
 - ◆ 关键字、标志符

数据类型的作用

- 数据类型就是约束变量存储数据的形式。

数据类型 变量名称 = 初始值;

```
public class HelloWorld {  
    public static void main(String[] args) {  
        double money = 9.9;  
    }  
}
```

```
int age = 23;  
}
```



数据类型的分类

- 引用数据类型（除基本数据类型之外的，如String ， 其他的后面学习）
- 基本数据类型：4大类8种。

数据类型	关键字	取值范围	内存占用（字节数）
整数	byte	-128~127	1
	short	-32768~32767	2
	int(默认)	-2147483648~2147483647 （10位数）	4
	long	-9223372036854775808 ~ 9223372036854775807 （19位数）	8
浮点数	float	1.401298e-45到3.402823e+38	4
	double（默认）	4.9000000e-324 到1.797693e+308	8
字符	char	0-65535	2
布尔	boolean	true, false	1

数据类型定义变量的几个补充知识点

- 随便写一个整数字面值，默认是int类型的，如果希望随便写一个整数默认是long型的必须在数据后加L或者l表示。
- 随便写一个小数字面值，默认是double类型的，如果希望这个小数是float类型的，必须在数据后加F或者f表示。



总结

1. 数据类型分为几种?

- 引用数据类型:String
- 基本数据类型: 4大类8种。
 - ✓ byte short **int (默认)** long 整型
 - ✓ float **double (默认)** 浮点型
 - ✓ char 字符型
 - ✓ boolean 布尔型

2. 随便写的整数、小数字面值默认什么类型?

- 23 , 默认是int类型 , 加上L/l就是long类型的数据了。
- 23.8, 默认是double类型, 加上F/f就是float类型了。



目录

Contents

- Java入门
- IDEA开发工具的使用
- Java语言
 - ◆ 注释
 - ◆ 字面量
 - ◆ 变量
 - ◆ 变量使用注意事项
 - ◆ 数据的存储形式：二进制
 - ◆ 数据类型
 - ◆ ASCII编码
 - ◆ 关键字、标志符

ASCII (American Standard Code for Information Interchange)

- 美国信息交换标准编码，就是一套现代英语、数字、和其他西欧字符（不含中文）的编号规则。
- 字符在计算机底层就是存储其编号的二进制形式。

字符	十进制	二进制
A	65	0100 0001
B	66	0100 0010
a	97	0110 0001
b	98	0110 0010
0	48	0011 0000



```
char ch = 'a';
```



总结

1. 字符在计算机底层是怎么存储的?
 - 存储的是二进制的形式。
2. ASCII编码是什么?
 - 对如英文字符进行编号的规则，规定英文等字符在底层存储的二进制形式。



目录

Contents

- Java入门
- IDEA开发工具的使用
- Java语言
 - ◆ 注释
 - ◆ 字面量
 - ◆ 变量
 - ◆ 变量使用注意事项
 - ◆ 数据类型
 - ◆ 数据的存储形式：二进制
 - ◆ ASCII编码
 - ◆ 关键字、标志符

关键字

- Java自己保留的一些单词，作为特殊功能的，例如：public、class、byte、short、int、long、double...
- 我们不能用来作为类名或者是变量名称，否则报错。
- 注意：关键字很多，不用刻意去记。

abstract	assert	boolean	break	byte
case	catch	char	class	const
continue	default	do	double	else
enum	extends	final	finally	float
for	goto	if	implements	import
instanceof	int	interface	long	native
new	package	private	protected	public
return	strictfp	short	static	super
switch	synchronized	this	throw	throws
transient	try	void	volatile	while

标识符

- 标识符就是由一些字符、符号组合起来的名称，用于给类，方法，变量等起名字的规矩。

标识符的要求

- 基本要求：由数字、字母、下划线(_)和美元符(\$)等组成
- 强制要求：不能以数字开头、不能是关键字、区分大小写

bj	b2	2b	class	\$2
_2b	#itheima	ak47	Class	HelloWorld



命名指导规范

- 变量名称：满足标识符规则，建议全英文、有意义、首字母小写，满足“驼峰模式”，例如：int studyNumber = 59。
- 类名称：满足标识符规则，建议全英文、有意义、首字母大写，满足“驼峰模式”，例如：HelloWorld.java。



总结

1. 关键字的特点

- Java自己保留的，我们不能用来取名字。

2. 标识符

- 取名字的规则，数字，字母，下划线，美元符等组成。
- 数字不能开头，不能是关键字。



传智教育旗下高端IT教育品牌