

# AP CSA

[安装IDE](#)

[组成这个世界的基本元素](#)

[什么叫程序](#)

[运行代码](#)

[逻辑运算](#)

[数据与数据类型](#)

[什么是数据与数据类型](#)

[Primitive type](#)

[bit to decimal to hex](#)

[variable — 变量](#)

[声明一个新的Primitive type的变量](#)

[对变量赋值并使用](#)

## 安装IDE

google搜索IntelliJ, 下载并安装 community版本

## 组成这个世界的基本元素

在物理层面上，程序的世界是由电讯号组成的。电讯号的高低位（电压高低）代表**逻辑上的是与非（1 和0）**，电子芯片通过传递运算，就展示出了你看到的这个世界。。。

这其中的步骤包含了大量的知识，而我们应该先从认识什么叫程序开始

## 什么叫程序

广义上讲，从一个我们在电脑端使用的程序（Application）到一段只能简单执行 $x+y$ 运算的代码都可以叫做程序

在Computer Science中我们有一些更加明确的称呼，（在 AP CSA中我们使用Java作为程序语言，以下称呼都与Java相关，在不同的编程语言中，我们可能使用不同称呼描述相同的概念）

以最原始的，最经典的HelloWorld为例：

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

我们先看这么一小段代码中都涉及到了那些基础称呼：

**Method**：一段由能够完成特定目标的代码，执行顺序从上到下，一个Method可以调用（Call）其他Method或者他自身，一个Method一定会结束

**Parameter**：参数，只在Method的构建过程中所使用的预期参数，并非实际使用的参数，每一个参数都要包含这个参数的数据类型（Data Type），在Java中我们使用Class的名称（Naming）作为数据类型，同时也可以使用Primitive type作为数据类型

**Return Type**: 该Method返回的值的类型

**Accessibility**: 成员访问权限，可以简单理解为谁能看到，谁能使用

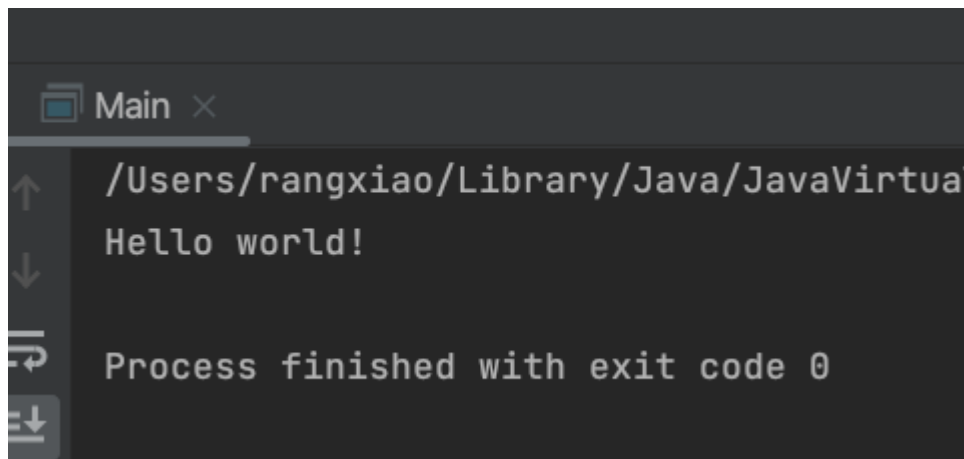
**Method Name**：这个Method的名称，同时也包括的调用（call）规则

**Binding**: 这个Method如何被使用，分为static与dynamic两种，我们目前大多是时候都是用的是dynamic binding，动态绑定，具体实现是由地址内存决定的，属于专业级别通识课

## 运行代码

我们目前写出来的代码想要运行起来，还不能像手机或者电脑上的app那么简单，首先我们的代码依存于一个环境，只有在特定的环境中才能运行起来，这就是为什么我们通过IDE来学习

在IDE中运行代码：

A screenshot of an IDE's terminal window. The window has a title bar with a folder icon and the text 'Main x'. The terminal output shows the file path '/Users/rangxiao/Library/Java/JavaVirtual...', the text 'Hello world!', and 'Process finished with exit code 0'. On the left side of the terminal, there are several icons: an upward arrow, a downward arrow, a circular arrow, and a square icon with a downward arrow.

简单来说这就是这段代码的运行结果了，但是实际上像这种能够显示出来的代码并不是我们进行程序设计是的大多数，大部分的时间，我们都是在对数据（Data）的运行设计运行规则。

想要学习如何设计规则，我们要先了解计算机世界的基本运行逻辑

## 逻辑运算

在计算机的世界中，物理层面上，使用的都是逻辑运算，通过逻辑运算得出计算结果，这也是为什么数据最终都是0和1

在二元世界中的逻辑判断逻辑分为以下几种

与（AND &）

```
1 & 1 = 1
1 & 0 = 0
0 & 0 = 0
```

或（OR |）

```
1 | 1 = 1
1 | 0 = 1
0 | 0 = 0
```

非（NOT !）

```
!1 = 0
!0 = 1
```

每一个逻辑字符的大小为 1bit，8 bit = 1 byte

## 数据与数据类型

### 什么是数据与数据类型

计算机中的一切都可以被视作数据，计算机通过对数据进行提取，运算，储存，与展示帮助我们进行计算

计算机如何将01变成我们看到的一切？

计算机通过识别不同的数据类型，对不同类型的数据进行不同的操作，比如，对于数字类型的数据就可以进行加减乘除，对于文字类的数据就可以进行增删改查，图片与视频也是通过对于颜色的变化而实现的，而颜色则是通过不同颜色的编码规则定义。

在Java中，最原始的数据类型被称作Primitive type

## Primitive type

在Java中，程序语言预先设定好的最基础的数据格式被称之为Primitive Datatype，分别是：

- **int** （整数 4byte）
- short （整数 2byte）
- long （整数 8 byte）
- string （字符串）
- float （浮点数、小数 4byte）
- **double** （浮点数、小数 8byte）
- **boolean** （逻辑字符）
- char （字符）
- byte （字节）

## bit to decimal to hex

bit表示二进制，只由0和1构成

decimal 表示十进制，由0~9构成

hex 表示16进制，由0~F构成

Decimal	Bit	Hex
0	0	0
1	1	1
2	10	2
3	11	3

4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	a
11	1011	b
12	1100	c
13	1101	d
14	1110	e
15	1111	f

## variable — 变量

如同我们在数学中使用的一样，在计算机中，我们通过声明来创建一个新的变量，同时可以对变量进行赋值以及使用

```
int a = 0;
int b = 1;
int c = a+b;
System.out.println(c);
```

在这段代码中，我们声明了三个新的variable a、b、c，并同时赋予他们各自的值。

同时他们就像是计算器一样，能够对值进行操作，通过不同的方法，可以通过命令让计算机帮助我们进行计算。计算的结果是多种多样的，并不仅限于数字上的计算，像是我们看到的视频，听到的声音，使用的功能，都是计算的结果。

## 声明一个新的Primitive type的变量

```
int a;
int b,c,d;
```

在变量声明的时候，我们需要指明变量的数据类型（datatype），就像我们之前说过的，这里也可以使用Class的名称作为数据类型的声明，在下一个章节中我们将会学习如何创建并使用一个Class。

## 对变量赋值并使用

```
a = 10;  
b = 100;  
c = 1 + a;  
d = c * b;
```

对于已经声明的变量，在使用过程中，不再需要说明他的数据类型。但是需要保证计算过程是符合规范的。在这里我们对相同的int类型进行数学操作，加法 `+` 和乘法 `*`，其他的数学逻辑也都有相对应操作符（Operation）

操作符	含义
+	加法
-	减法
*	乘法
/	除法
%	求余运算
=	赋值，将右侧的运算结果赋值给左侧的变量中

在数学运算之外，我们还可以进行使用逻辑上的比较

逻辑操作符	含义
==	相等/相同
>	大于
≥	大于或等于
<	小于
≤	小于或等于
!=	不等于

逻辑比较的结果为 `boolean` 类型，值为 `True` 或者 `False`，并通过逻辑运算相连接。在这里的逻辑运算所使用的符号与之前bit操作符之间略有不同，更多的，我们使用的是这一种

&&	And
	Or
!	Not

运算的逻辑与之前相同



### 练习

声明两个新的整数类型的变量 `int a` 和 `int b`，并分别给他们赋值12和349。比较a与b的大小，并将比较的结果储存在一个 `boolean` 类型的变量c中。