

JavaScript基础第01天

1 - 编程语言

1.1 编程

- 编程：就是让计算机为解决某个问题而使用某种程序设计语言编写程序代码，并最终得到结果的过程。
- 计算机程序：就是计算机所执行的一系列指令集合，而程序全部都是用我们所掌握的语言来编写的，所以人们要控制计算机一定要通过计算机语言向计算机发出命令。
- 从事编程的人员，就是程序员。但是一般程序员都比较幽默，为了形容自己的辛苦工作，也成为“码农”，或者“程序猿”/“程序媛”
- 注意：上面所定义的计算机指的是任何能够执行代码的设备，可能是智能手机、ATM机、黑莓PI、服务器等等。

1.2 计算机语言

- 计算机语言指用于人与计算机之间通讯的语言，它是人与计算机之间传递信息的媒介。
- 计算机语言的种类非常的多，总的来说可以分成机器语言，汇编语言和高级语言三大类。
- 实际上计算机最终所执行的都是机器语言，它是由“0”和“1”组成的二进制数，二进制是计算机语言的基础。

0=00000000	1=00000001	2=00000010	3=00000011	4=00000100
5=00000101	6=00000110	7=00000111	8=00001000	9=00001001
10=00001010				

1.3 编程语言

- 1 **编程语言：**可以通过类似于人类语言的“语言”来控制计算机，让计算机为我们做事情，这样的语言就叫做编程语言（Programming Language）。
- 2
- 3 编程语言是用来控制计算机的一系列指令，它有固定的格式和词汇（不同编程语言的格式和词汇不一样），必须遵守。
- 4
- 5 如今通用的编程语言有两种形式：**汇编语言和高级语言**。

语言类型	说明
汇编语言	汇编语言和机器语言实质是相同的，都是直接对硬件操作，只不过指令采用了英文缩写的标识符，容易识别和记忆。
高级语言	高级语言主要是相对于低级语言而言，它并不是特指某一种具体的语言，而是包括了很多编程语言，常用的有C语言、C++、Java、C#、Python、PHP、JavaScript、Go语言、Objective-C、Swift等。

```
C语言: puts("你好");
PHP: echo "你好";
Java: System.out.println("你好");
JavaScript: alert("你好");
```

1.4 翻译器

- 1 高级语言所编制的程序不能被计算机识别，必须经过转换才能被执行，为此，我们需要一个翻译器。

翻译器可以将我们所编写的源代码转换为机器语言，这也被称为二进制化。记住1和0。



1.5 编程语言和标记语言区别

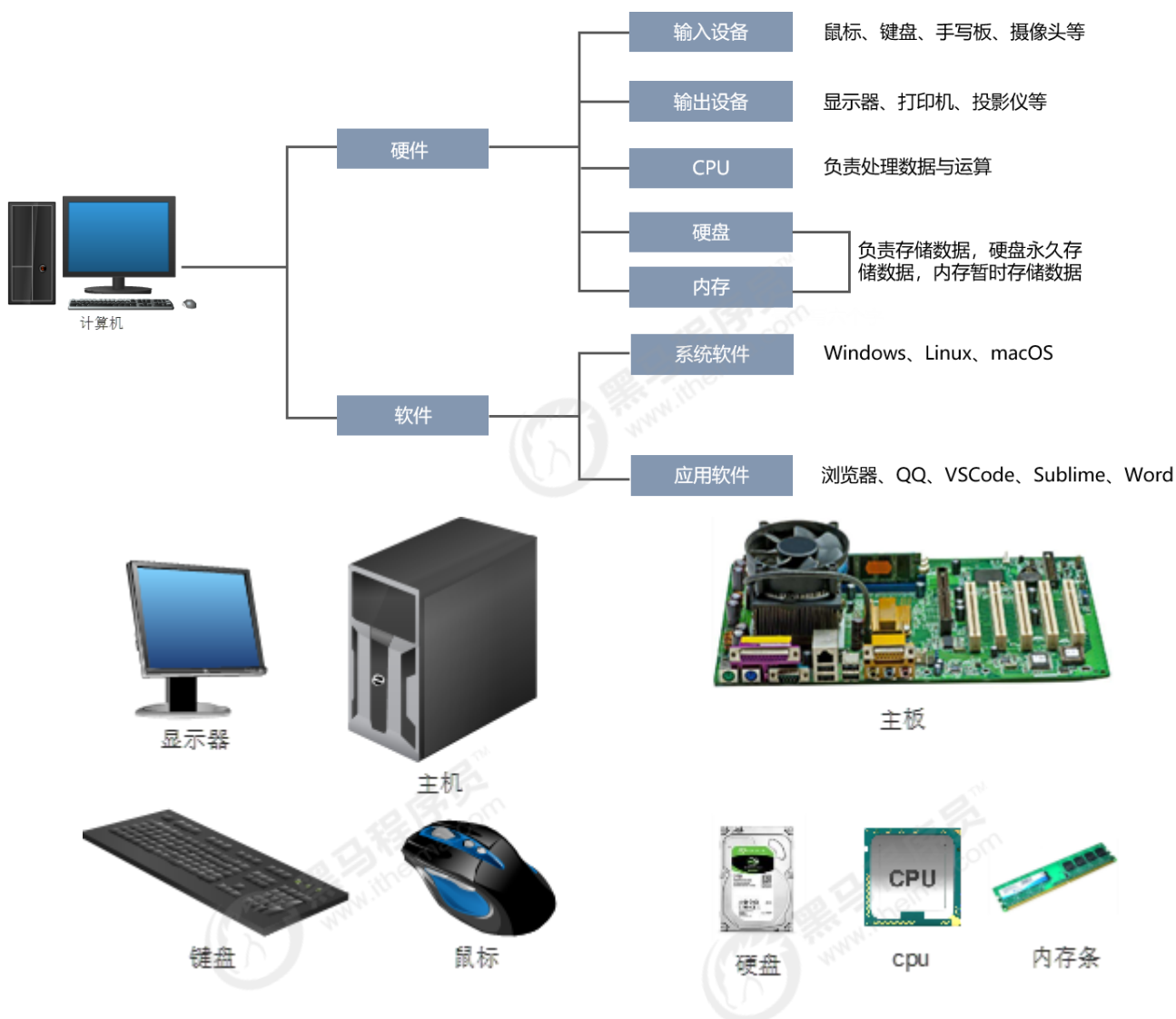
语言	说明
编程语言	编程语言有很强的逻辑和行为能力。在编程语言里, 你会看到很多 if else 、 for 、 while等具有逻辑性和行为能力的指令，这是主动的。
标记语言	标记语言（html）不用于向计算机发出指令，常用于格式化和链接。标记语言的存在是用来被读取的, 他是被动的。

总结

1. 计算机可以帮助人类解决某些问题
2. 程序员利用编程语言编写程序发出指令控制计算机来实现这些任务
3. 编程语言有机器语言、汇编语言、高级语言
4. 高级语言需要一个翻译器转换为计算机识别的机器语言
5. 编程语言是主动的有很强的逻辑性

2 - 计算机基础

2.1 计算机组成



2.2 数据存储

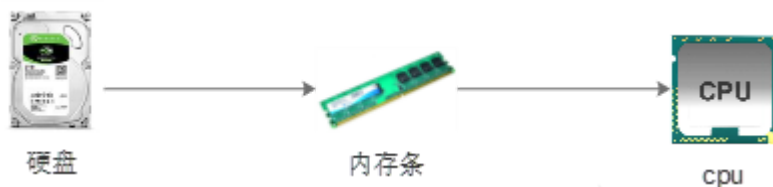
1. 计算机内部使用二进制 0 和 1 来表示数据。
2. 所有数据, 包括文件、图片等最终都是以二进制数据 (0 和 1) 的形式存放在硬盘中的。
3. 所有程序, 包括操作系统, 本质都是各种数据, 也以二进制数据的形式存放在硬盘中。平时我们所说的安装软件, 其实就是把程序文件复制到硬盘中。
4. 硬盘、内存都是保存的二进制数据。

2.3 数据存储单位

1 | 大小关系: bit < byte < kb < GB < TB<.....

- 位(bit): 1bit 可以保存一个 0 或者 1 (最小的存储单位)
- 字节(Byte): 1B = 8b
- 千字节(KB): 1KB = 1024B
- 兆字节(MB): 1MB = 1024KB
- 吉字节(GB): 1GB = 1024MB
- 太字节(TB): 1TB = 1024GB

2.4 程序运行



1 计算机运行软件的过程：

1. 打开某个程序时，先从硬盘中把程序的代码加载到内存中
2. CPU执行内存中的代码

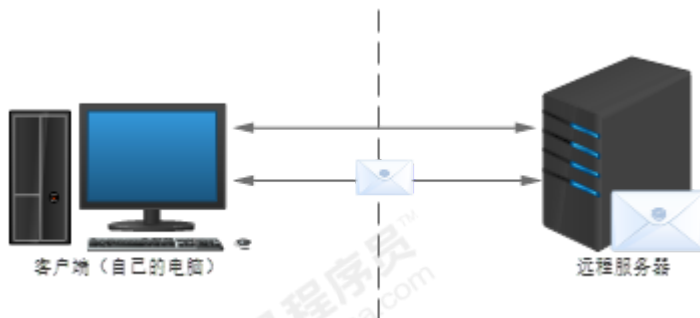
注意：之所以要内存的一个重要原因，是因为 cpu运行太快了，如果只从硬盘中读数据，会浪费cpu性能，所以才使用存取速度更快的内存来保存运行时的数据。（内存是电，硬盘是机械）

3 - 初始JavaScript

3.1 JavaScript 是什么

布兰登·艾奇（Brendan Eich，1961年~）。神奇的大哥用10天完成 JavaScript 设计。最初命名为 LiveScript，后来在与 Sun 合作之后将其改名为 JavaScript。

- JavaScript 是世界上最流行的语言之一，是一种运行在客户端的脚本语言（Script 是脚本的意思）
- 脚本语言：不需要编译，运行过程中由 js 解释器(js 引擎)逐行来进行解释并执行
- 现在也可以基于 Node.js 技术进行服务器端编程



3.2 JavaScript的作用

- 表单动态校验（密码强度检测）（JS 产生最初的目的）
- 网页特效
- 服务端开发(Node.js)
- 桌面程序(Electron)
- App(Cordova)
- 控制硬件-物联网(Ruff)
- 游戏开发(cocos2d-js)

3.3 HTML/CSS/JS 的关系

HTML/CSS 标记语言--描述类语言

- HTML 决定网页结构和内容(决定看到什么), 相当于人的身体
- CSS 决定网页呈现给用户的模样(决定好不好看), 相当于给人穿衣服、化妆



JS 脚本语言--编程类语言

- 实现业务逻辑和页面控制(决定功能), 相当于人的各种动作



3.4 浏览器执行 JS 简介

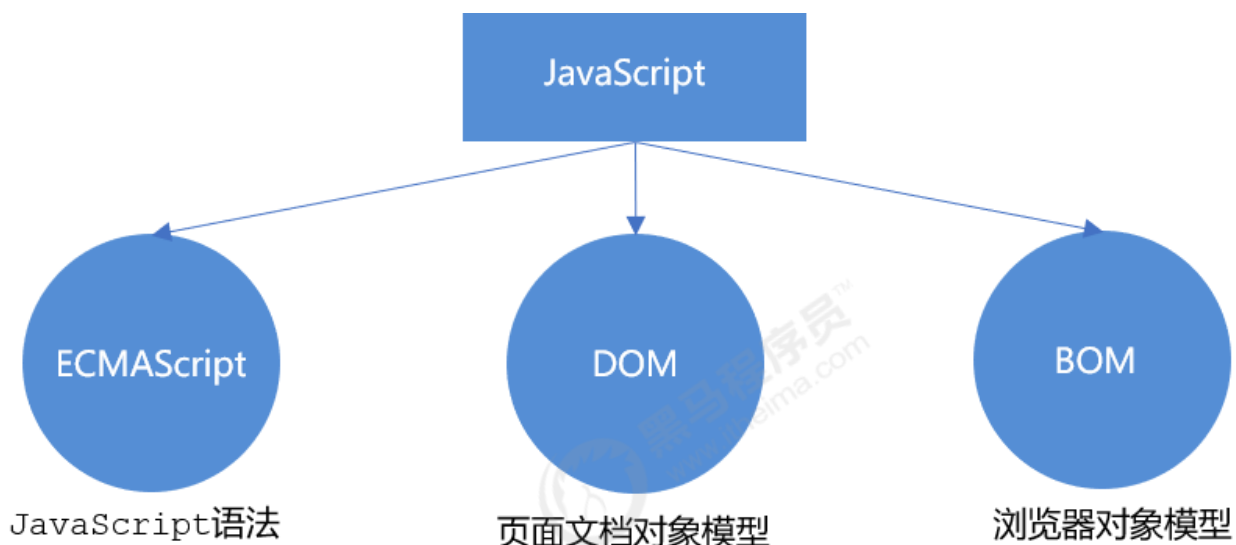
浏览器分成两部分：渲染引擎和 JS 引擎

渲染引擎	用来解析HTML与CSS，俗称内核，比如 chrome 浏览器的 blink，老版本的 webkit
JS 引擎	也称为 JS 解释器。用来读取网页中的JavaScript代码，对其处理后运行，比如 chrome 浏览器的 V8

- 1 浏览器本身并不会执行JS代码，而是通过内置 JavaScript 引擎(解释器) 来执行 JS 代码。JS 引擎执行代码时逐行解释每一句源码（转换为机器语言），然后由计算机去执行，所以 JavaScript 语言归为脚本语言，会逐行解释执行。



3.5 JS 的组成



1. ECMAScript

ECMAScript 是由ECMA 国际（原欧洲计算机制造商协会）进行标准化的一门编程语言，这种语言在万维网上应用广泛，它往往被称为 JavaScript或 JScript，但实际上后两者是 ECMAScript 语言的实现和扩展。



ECMAScript：规定了JS的编程语法和基础核心知识，是所有浏览器厂商共同遵守的一套JS语法工业标准。

更多参看MDN: [MDN手册](#)

2. DOM——文档对象模型

文档对象模型 (Document Object Model, 简称DOM)，是W3C组织推荐的处理可扩展标记语言的标准编程接口。通过 DOM 提供的接口可以对页面上的各种元素进行操作（大小、位置、颜色等）

3. BOM——浏览器对象模型

浏览器对象模型(Browser Object Model, 简称BOM) 是指浏览器对象模型，它提供了独立于内容的、可以与浏览器窗口进行互动的对象结构。通过BOM可以操作浏览器窗口，比如弹出框、控制浏览器跳转、获取分辨率等。

3.6 JS 初体验

1 JS 有3种书写位置，分别为行内、内嵌和外部。

1. 行内式

```
1 <input type="button" value="点我试试" onclick="alert('Hello World')" />
```

- 可以将单行或少量 JS 代码写在 HTML 标签的事件属性中（以 on 开头的属性），如：onclick
- 注意单双引号的使用：在 HTML 中我们推荐使用双引号，JS 中我们推荐使用单引号
- 可读性差，在 html 中编写 JS 大量代码时，不方便阅读；
- 引号易错，引号多层嵌套匹配时，非常容易弄混；
- 特殊情况下使用

2. 内嵌式

```
1 <script>
2     alert('Hello World~!');
3 </script>
```

- 可以将多行 JS 代码写到