

# 计算机基础（一）

---

## 前端和后端的概念

---

两者是相对的概念。

**前端，也称为“客户端开发”**，最常见的为Web前端。其实前端的概念还算是比较广泛的，比如网页，APP，客户端。依赖于后端数据和服务的，都可以称之为前端。

前端注重于**用户界面和交互**相关，你访问的网站的内容设计、图像、段落和线条之间的间距，按钮，图像，通知提醒，你所见的都是前端。此外点击按钮页面响应功能，或者是移动APP响应你所做的所有手势时，需要进行缩放放大/缩小、双击、滑动等操作。

前端开发人员构建的内容，创造了你在屏幕上看到的外观和触摸到的体验。

**后端，也称为“服务器端开发”**，在后端服务器和浏览器或应用程序之间存储网站、应用数据和中间媒介的服务器都属于后端。后端的核心是**效率，稳定与性能**。

比如你在短视频平台上对视频进行了点赞，系统就会记录这个数据，并且依照后台记录的数据再给你推送同类的视频。

网站的后端更侧重于**逻辑设计**，需要设置服务器，存储和检索数据，以及写好这些服务器与前端进行连接的接口，同时需要关注网站的速度、性能和响应能力。

## 文件系统

---

### 文件

大部分操作系统支持两部分的文件名。它们用 . 分隔开，比如文件名 `test.c`。

点前面的部分是文件的名称。

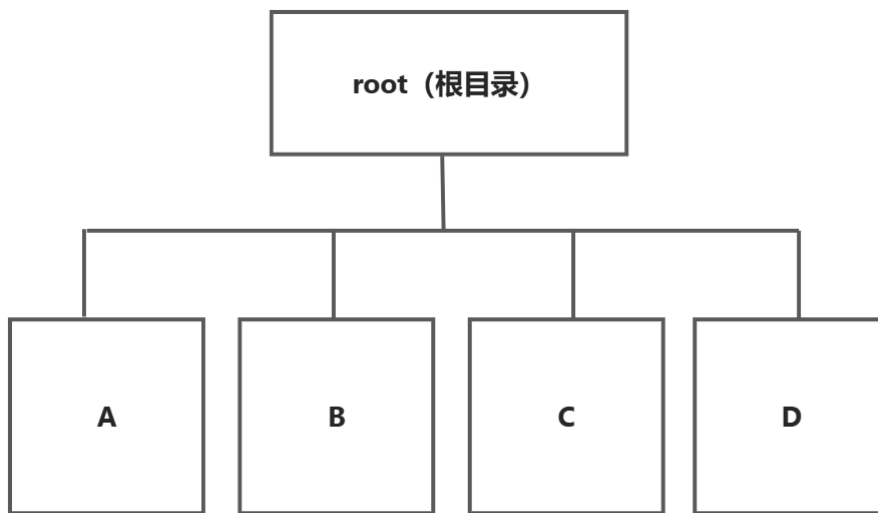
点后面的部分称为文件扩展名，文件扩展名通常表示文件的一些信息。例如 `test.html`，`test.zip` 等，html 表示一个 web 网页，.zip 表示压缩文件。

右键单击“属性”可以查看文件的详细信息。

### 目录

文件系统通常提供目录或者文件夹用于记录文件的位置。

目录系统最简单的形式是有一个能够包含所有文件的目录。这种目录被称为根目录(root directory)。在最早期的个人计算机中，这种系统很常见，部分原因是因为只有一个用户。这种目录组织形式现在一般用于简单的嵌入式设备（如数码相机和某些便携式音乐播放器）上使用。

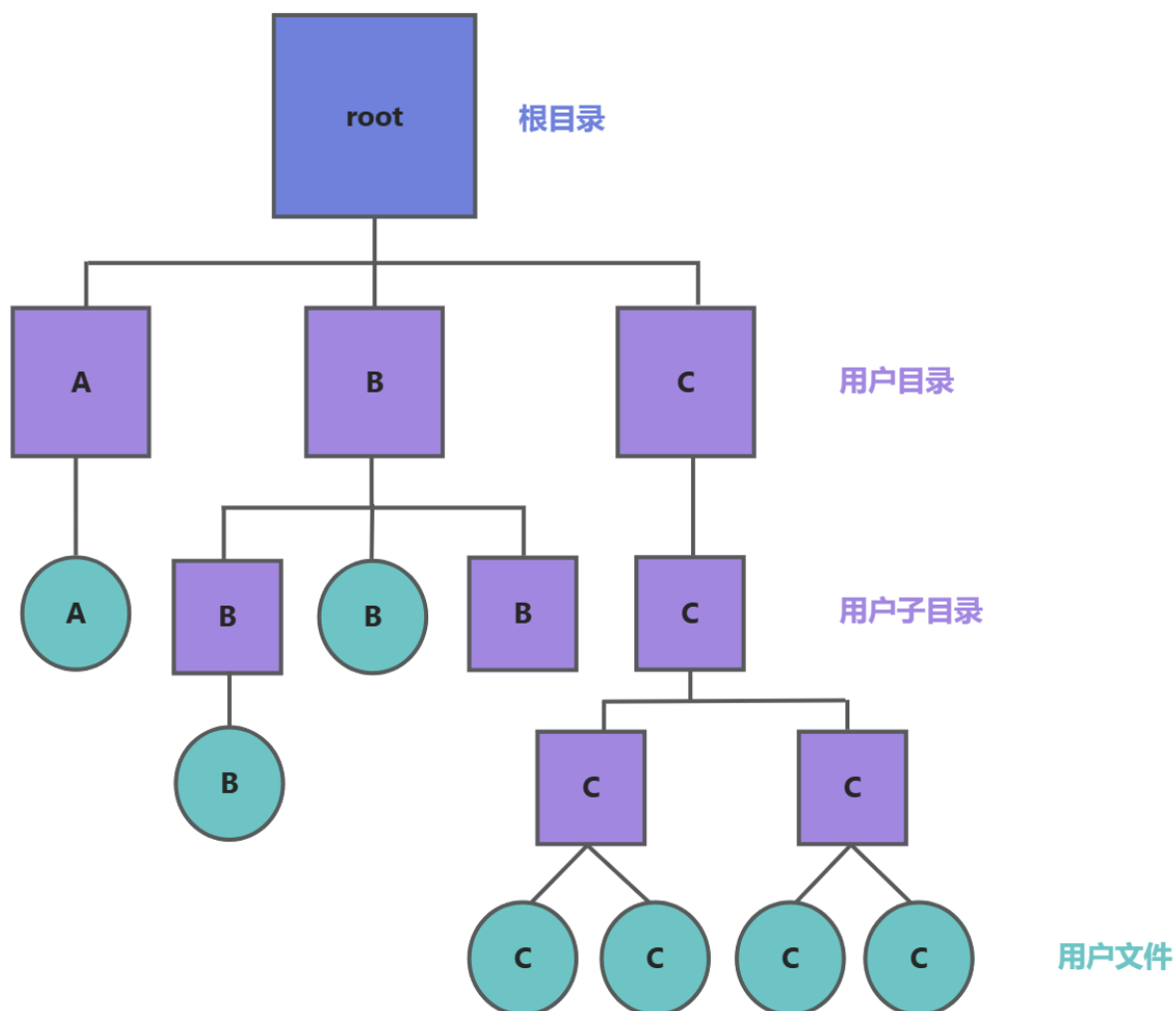


现代计算机含有成千上万个文件和文件夹。如果都放在根目录下，查找起来会非常困难。

为了解决这一问题，出现了**层次目录系统**(Hierarchical Directory Systems)，也称为**目录树**。通过这种方式，可以用很多目录把文件进行分组。如果多个用户共享同一个文件服务器，每个用户可以为自己的目录树拥有自己的私人根目录。

如下图，根目录含有目录 A、B 和 C，分别属于不同的用户，其中两个用户各自创建了子目录。

每个用户可以创建任意数量的子目录。



## 路径

当目录树组织文件系统时，需要有某种方法指明文件名。

### 绝对路径

绝对路径是文件真实的存放位置，它由根目录到文件的路径组成。

绝对路径也有唯一确定的起点，也就是“根目录”。在Windows中，路径的分隔符为\

例如 `F:\Code\hello.c`，`F:\`（F盘）就是根目录。`hello.c`文件放在F盘下面的C code文件夹里。

### 相对路径

相对路径是相对于当前位置的路径。首先需要清楚地知道当前的目录在什么位置。

`cd`：切换当前路径的命令

输入**盘符（字母+冒号）**可以切换到对应的根上去。

`"."`（一个点，即英文句号），代表当前目录

`".."`（两个点，即两个英文句号），代表上一级目录

`"/"`（斜杠），代表根目录

`F:\Code\hello.c` 这个路径的上一级目录是 `F:\Code`

`F:\Code` 的上一级目录就是 `F:`

在图片引用中经常用到相对路径

`../img.png`

引用上级文件：`../xxx.jpg`

引用同级文件：`xxx.jpg`

引用下级文件：当前文件夹/`xxx.jpg`

引用上上级文件：`../../xxx1.jpg`

有的时候，你在安装软件可能会看见下面的警告：

“程序检测到您的安装路径有中文字符 请将文件夹移到任意英文目录在运行此工具。”

这是由于英文和中文的编码问题，中文需要特殊的编码，有些海外公司并没有考虑这个问题，两种编码方式不兼容，就导致中文字符会被截断，然后被解释成其他的东西。

**不要在文件安装目录中使用中文路径**

## 字符编码

世界上各种语言里的字和标点符号都属于字符。一堆字符组成的集合叫做字符集。

计算机并不认识人类世界中的字符。计算机只认识二进制。为了让计算机能够识别字符，就需要把这些字符用 0 和 1 表示出来。把字符转换成 0 和 1 的规则叫做字符编码。

计算机存储数据的基本单位是字节，也就是bite。1 个字节由 8 个二进制位组成，也就是 8 个比特。每个比特有 0 和 1 两种状态。8 个比特一共有 2 的 8 次方，也就是 256 种不同的状态。所以如果只用一个字节长度来编码字符，也就是用 8 个 0 或者 1 来代表一个字，这个字符集就能容纳 256 个字符。

## ASCII

美国信息交换标准代码：共定义了128个字符

## GB2312

我国 1980 年公布的 GB2312-80(信息交换用汉字编码字符集基本集)，包含了 6763 个常用简体汉字，以及一些标点符号、数字、拉丁字母等。

GB2312使用两个字节长度来编码一个字符，这样理论上就可以容纳 2 的 16 次方，也就是 65536 个字符。

现在输入法里的全角标点符号就是用两个字节编码的，而半角标点符号就是用一个字节编码的。

在更换字体后，有时候会发现一些格格不入的孤儿文字，或者字与字之间出现一些□□□。

沂沃沅沅沅沅

沂沃沅沅沅沅

沂沃□ 沅沅□

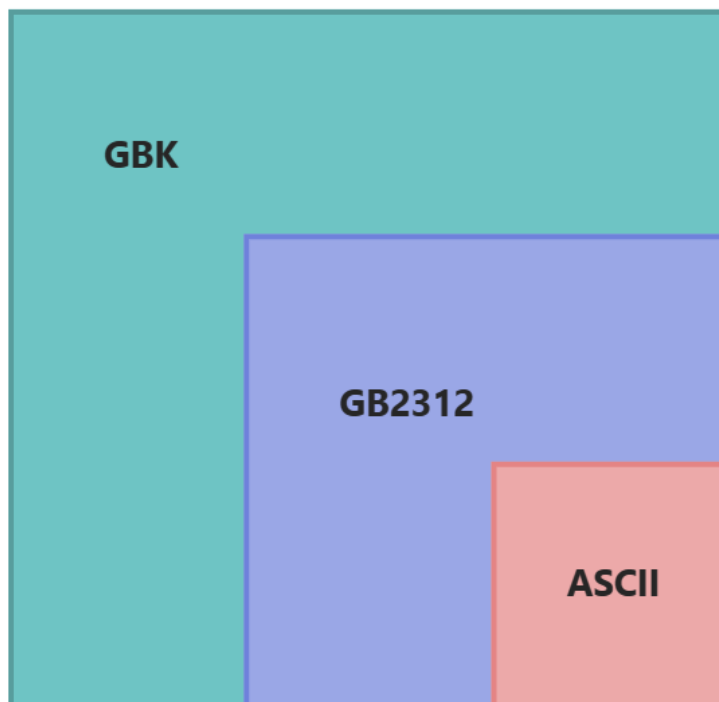
因为绝大多数简体汉字字体都还是只适配了GB2312中的 6000 多个常用汉字。如果遇到比较生僻的字，它们并不在字符集中，计算机要么调用其他字体来显示，要么直接显示成□。

## GBK

中文汉字有简体，有繁体；日本使用日本汉字；韩国用朝鲜汉字。于是出现了各种各样的字符集和编码规则，但是它们相互之间的兼容性很差。

后来微软根据各地字符集和字符编码，扩展了 GB2312 字符集，加入了一些繁体汉字。

这份扩展的字符集后来就成为了汉字内码扩展规范GBK（k 就是扩展的意思）。但GBK并不是国家标准，只是一个技术规范。



## Unicode

随着时代的发展，计算机在全球迅速普及，各种字符编码已经乱的不能再乱了。后来出现了一种字符集——Unicode，也叫万国码或者统一码。目前已经成为了计算机科学领域最常用的字符集。

不同于之前的字符集，Unicode字符集有很多种编码方式，比如 UTF-8、UTF-16、UTF-32 等。这些都是把 Unicode 字符集中的字符转换成二进制信息的编码方式。

目前最常用的是 UTF -8 编码。

## 乱码的产生

即使是同一串二进制数字，在不同的字符集和编码方式下，也会对应不同的字符。

所以，如果用错误的编码打开文本，就会出现乱码。

比如，你大概率看见过这些：

锟?斤?拷? ? ? ? ? ? ? ?

在打开一些年代久远的小说txt文件，也可能会发现一堆? ? ? ? ? ?

在早期的Windows系统中，简体中文版的记事本的默认编码是GBK。

如今，新版的 windows 系统已经把记事本的默认编码改成了 UTF-8，很多常用软件也做了一些应对措施，尽量避免乱码。

## 浏览器

---

HTML全称为超文本标记语言，是一种用来定义网页的一种标记语言

- CSS用于修饰网页元素，使其表现更丰富
- JS即JavaScript，是一种支持面向对象的解释型语言，最常用于开发

浏览器的主要区别在于渲染引擎和JS解析引擎。

渲染引擎（内核）：把网页内容（html、css）转换成可见的页面  
JS解析引擎：解析JS代码转换为二进制编码文件。

以下是几大主流浏览器的渲染引擎和JS解析引擎：

浏览器	JS解析引擎	渲染引擎（内核）
Chrome	V8	blink
Edge	Edge JScript（IE3.0-IE8.0 / Chakra（IE9+之后）	EdgeHTML
Opera	Linear A（4.0-6.1） / Linear B（7.0-9.2） / Futhark（9.5-10.2） / Carakan（10.5-）	blink
Safari	Nitro（4-）	webkit
Firefox	SpiderMonkey（1.0-3.0） / TraceMonkey（3.5-3.6） / JaegerMonkey（4.0-）	Gecko

建议使用 Chrome/Edge

## HTTP协议

HTTP协议(超文本传输协议HyperText Transfer Protocol)，简单来说就是客户端和服务端进行数据传输的一种规则，用于进行前后端的交互，是互联网数据传辑和资源共享的基础。

客户端（前端）发送请求，服务端（后端）返回响应。

## URL

客户端发送请求，首先要知道URL，URL也就是常说的网址，通常由协议，域名，端口号，文件路径，参数构成。

```
http://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2
```

## 协议

http:// 为协议名,标明了请求需要使用的协议,通常使用的是 HTTP 协议或者安全协议 HTTPS  
HTTPS（Hypertext Transfer Protocol Secure：超文本传输安全协议）:提供数据加密的传输协议  
HTTP 标准端口是80，而 HTTPS 的标准端口是443

## 域名

`www.example.com` 为域名,标明了需要请求的服务器的地址。

## 端口号

`:80` 是端口号,标明了获取服务器资源的入口, **端口号用于区分服务的端口。**

一台服务器可以提供许多服务, 通过“IP地址+端口号”来区分不同的服务。如果把服务器比作房子,端口号可以看做是通向不同服务的门。

## 文件路径

`/path/to/myfile.html` 表示服务器上资源的路径, 路径表示的只是一个抽象地址,并不指代任何物理地址。

服务端收到URL并对其进行解析并返回相应的数据。

## 参数

`?key1=value1&key2=value2` 是请求里提供的参数.这些**参数是以键值对的形式发送,通过 & 符号分隔。**

```
http: //127.0.0.1: 8080
```

**http**是协议, **127.0.0.1**是本地IP地址, **8080**是默认的端口号

## HTTP请求

在浏览器上输入某个URL地址或者单击网页上的一个超链接时, 浏览器将使用GET方式发送请求。

HTTP请求信息由3部分组成:

请求方法URI协议/版本, 请求头, 请求正文

1.请求方法URI协议/版本

**请求方式**常见有以下几种, **HTTP请求方式默认为GET请求:**

方式	描述
GET	获取资源
POST	发送资源
PUT	更新资源
DELETE	删除资源

一个GET请求的实例:

<http://weibo.com/signup/signup.php?inviteCode=2388493434>

GET请求的数据会附加在URL之后，以?分割URL和传输数据，多个参数用&连接。

URL的编码格式采用的是ASCII编码，而不是unicode，即是说所有的非ASCII字符都要编码之后再传输。

使用GET请求长度有限制，发送数据量小；POST请求长度无限制，发送数据量大；

GET请求数据信息可以在URL中直接看到。POST通过request body传递参数，用户无法直接看到。所以**POST的安全性比GET的高**。

## 2.请求头(Request Header)

请求头包含许多有关的客户端环境和请求正文的有用信息。例如，浏览器所用的语言，请求发送的时间等。

```
▼ Request Headers view source
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9
Cache-Control: max-age=0
Connection: keep-alive
Host: localhost:63342
If-Modified-Since: Wed, 07 Mar 2018 23:13:52 GMT
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.186 Safari/537.36
```

## 3.请求正文

**请求头和请求正文之间是一个空行**，它表示请求头已经结束，接下来的是请求正文。请求正文中可以包含用户提交的查询字符串信息。

```
▼ Query String Parameters view source view URL encoded
blogId: 161525
blogApp: tughenhua0707
```

# HTTP响应

HTTP响应由3个部分构成：

协议状态版本代码描述，响应头，响应正文

```
HTTP/1.1 200 OK
Date: Sat, 31 Dec 2005 23:59:59 GMT
Content-Type: text/html; charset=ISO-8859-1
Content-Length: 122

<html>
<head>
<title>Wrox Homepage</title>
</head>
<body>
<!-- body goes here -->
</body>
</html>
```

状态行

消息报头

空行

下面的就是响应正文了

## 1.协议状态版本代码描述



#### ▼ General

Request URL: http://localhost:63342/webpack/index.html

Request Method: GET

Status Code: 200 OK

Remote Address: 127.0.0.1:63342

Referrer Policy: no-referrer-when-downgrade

HTTP 状态码由三个十进制数字组成，第一个十进制数字定义了状态码的类型。

响应分为五类：信息响应(100-199)，成功响应(200-299)，重定向(300-399)，客户端错误(400-499)和服务器错误 (500-599)

状态码	响应
100——199	信息响应
200——299	成功响应
300——399	重定向[301 Moved Permanently]
400——499	客户端错误[404 Not Found]
500——599	服务端错误[500 Internal Server Error]

#### 2.响应头(Response Header)

包含了例如服务器类型、日期时间、内容类型和长度等信息。

#### ▼ Response Headers

[view source](#)

cache-control: private, must-revalidate

content-length: 189

content-type: text/html

date: Thu, 08 Mar 2018 00:23:52 GMT

last-modified: Thu, 08 Mar 2018 00:23:48 GMT

server: WebStorm 2016.1.1

#### 3.响应正文

即服务器返回的HTML页面

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>webpack</title>
</head>
<body>
  <div>12345</div>
</body>
</html>
<script src="bundle.js"></script>
```

# 建议参考文档

---

[HTTP | MDN \(mozilla.org\)](#)