

0x01 前言

最近写PHP代码比较多，也算不上什么精通，感觉要使用php就得知道这门语言的底层原理，这里决定了解一下PHP底层的工作原理。

0x02 什么是PHP

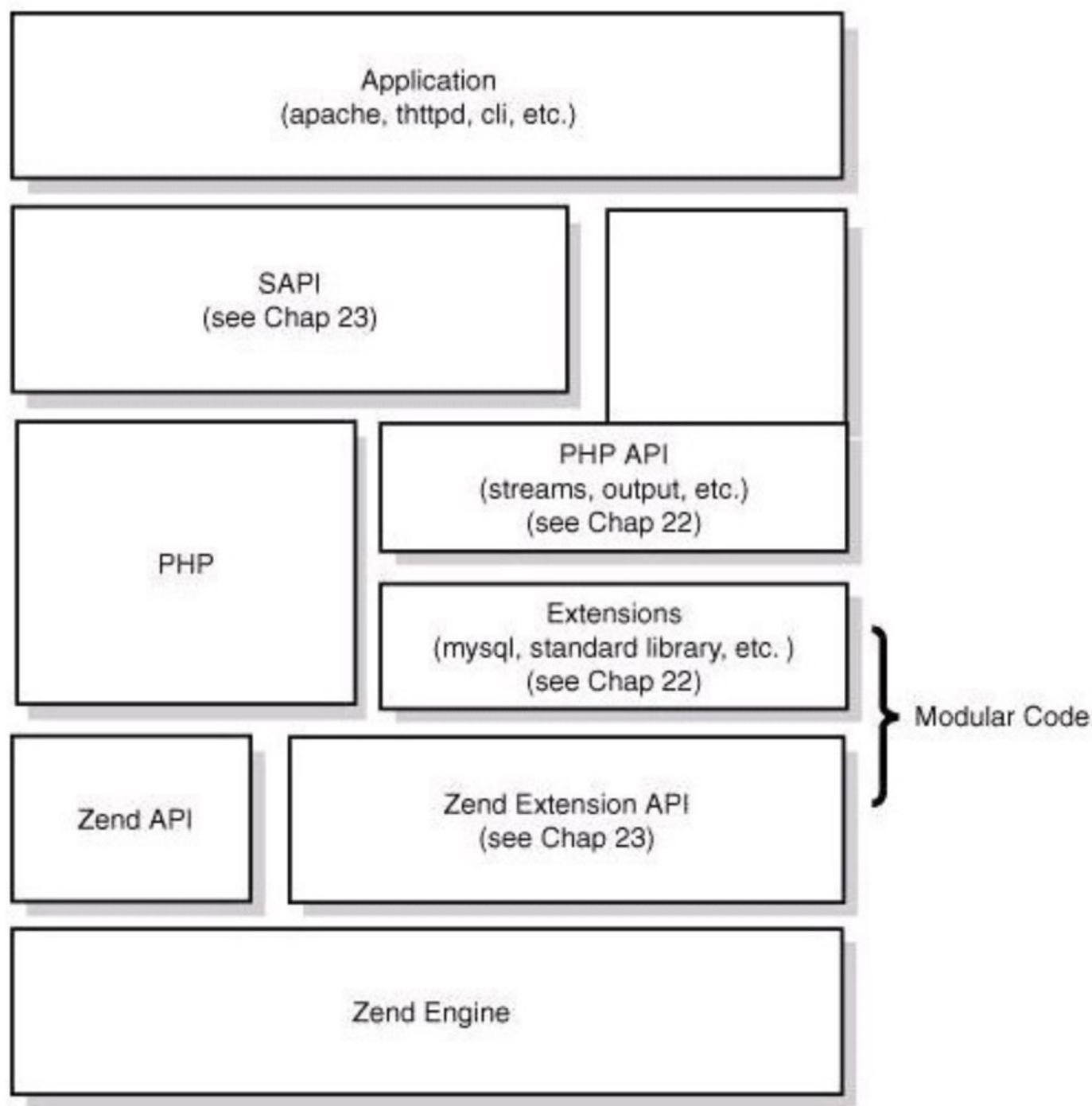
PHP是一种适用于Web开发的动态语言，是一个用C语言实现，包含大量组件的软件框架。狭义方面来说，可以说是个UI框架(用户界面,也可以说是UI模版([UI定义](#)))

0x03 PHP设计理念与特点

- 多进程模型：由于PHP是多进程模型，不同请求间互不干涉，这样保证了一个请求挂掉不会对全盘服务造成影响，当然，随着时代发展，PHP也早已支持多线程模型。
- 弱类型语言：和C/C++、Java、C#等语言不同，PHP是一门弱类型语言。一个变量的类型并不是一开始就确定不变，运行中才会确定并可能发生隐式或显式的类型转换
- 引擎(Zend)+组件(ext)的模式降低内部耦合。
- 中间层(sapi)隔绝web server和PHP。
- 语法简单灵活，没有太多规范。

0x04 PHP核心架构

php四层体系如下：



从下到上的四层体系:

- Zend引擎：Zend整体用C语言实现，是PHP的内核部分。它将PHP代码翻译，实现了基本的数据结构，内存分配机制及管理，提供了相应的api供外部调用，是一切的核
- Extensions：围绕Zend引擎，extensions通过组件式的方式提供各种基础服务，我们常见的各种内置函数、标准库等都是通过extension来实现，用户也可以根据需要实现自己的extension。
- Sapi：Aerver Application Programming Interface，即服务端应用编程接口，是PHP和web server的中间层。Sapi通过钩子函数，使PHP能和外部交互数据，这也将PHP和上层应用解耦。
- 上层应用：就是我们平时编写的PHP程序，通过不同的sapi方式得到各种各样的应用模式

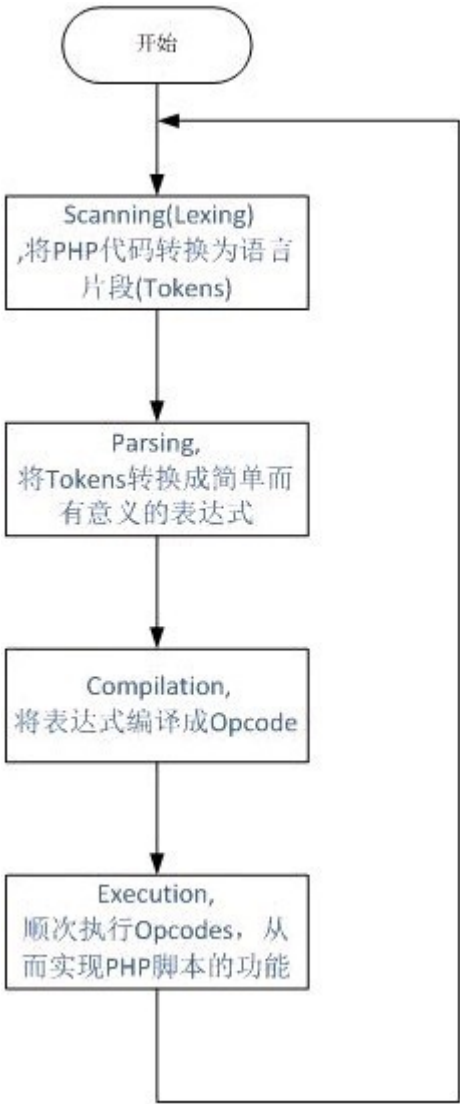
这里有个很好的通俗比喻:如果PHP是一辆车，那么车的框架就是PHP本身，Zend是车的引擎（发动机），Extensions下面的各种组件就是车的轮子，Sapi可以看做是公路，

0x05 Sapi

常见的Sapi有：

- apache2handler：以apache作为webserver，采用mod_PHP模式运行时候的处理方式，也是现在应用最广泛的一种。
- fast-cgi：这是webserver和PHP直接的另一种交互方式，也就是大名鼎鼎的fastcgi协议，在最近fastcgi+PHP得到越来越多的应用，也是异步webserver所唯一支持的方式
- cli：命令行调用的应用模式

0x06 PHP执行流程



拿到一段代码后，经过词法解析、语法解析等阶段后，源程序会被翻译成一个指令（ opcodes ），然后ZEND虚拟机顺次执行这些指令完成操作。PHP本身是用C实现的，因此PHP的执行的核​​心是翻译出来的一条一条指令，也即opcode。

Opcode是PHP程序执行的最基本单位。一个opcode由两个参数(op1,op2)、返回值和处理函数组成。PHP程序最终被翻译为一组opcode处理函数的顺序执行。

0x07 PHP变量

PHP是一门弱类型语言，本身不严格区分变量的类型。PHP变量可以分为简单类型(int、string、boolean)、集合类型(array、resource、object)和常量(const)。所有变量在内存中都是通过zval来存储的。

zval主要由三部分组成：

```
type■■■■■■■■■■
refcount&is_ref■■■■■■■■■■
value■■■■■■■■■■(■■)■■■■■■■■■■zvalue■■■■union■■■■■■■■■■
```

0x08 相关文章

- <https://sphenginx.github.io/2016/07/11/php-theory/>
- <https://seminelee.github.io/2017/11/19/php/>
- <http://www.nowamagic.net/librarys/veda/detail/102>

具体内容可以参考上面几篇文章

- [动动手指，沙发就是你的了！](#)

[登录](#) 后跟帖

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

目录

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)