搭建服务器

|  |  |
| --- | --- |
| 服务器 | 安装了特定功能的公共计算机，装了这个特定功能的软件，软件就可以提供特定功能的服务 |
| 服务类型 | Web服务器、数据库服务器、文本服务器 |
| 客户端 | 服务过程中这个服务的设备--网络端点 |
|  |  |

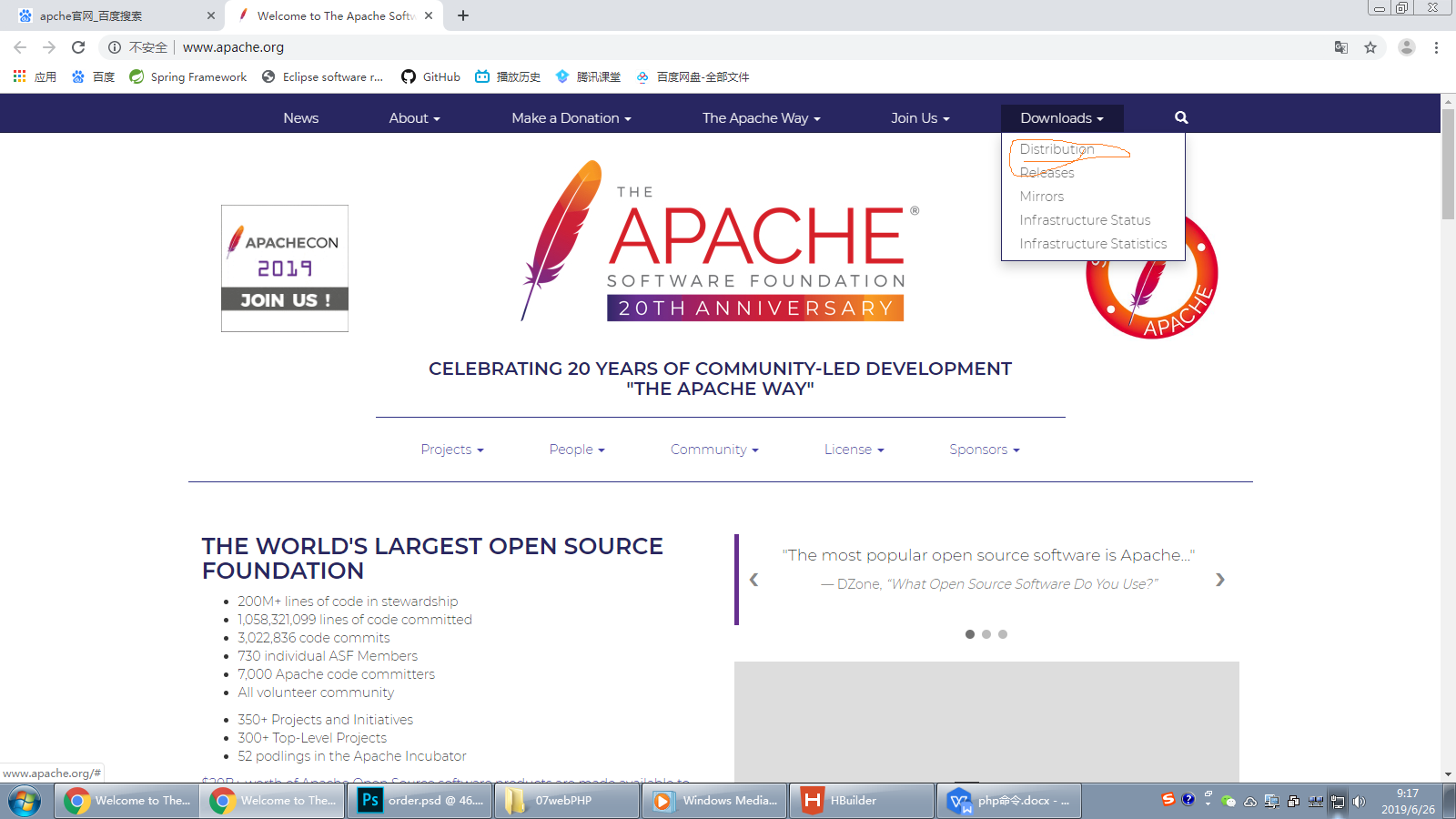
服务器软件

|  |  |
| --- | --- |
| IIS | 微软服务器 |
| Apache | 常与php搭配 |
| Nginx | 反向代理 |
| Tomcat | 常与Java搭配 |

Apache官网

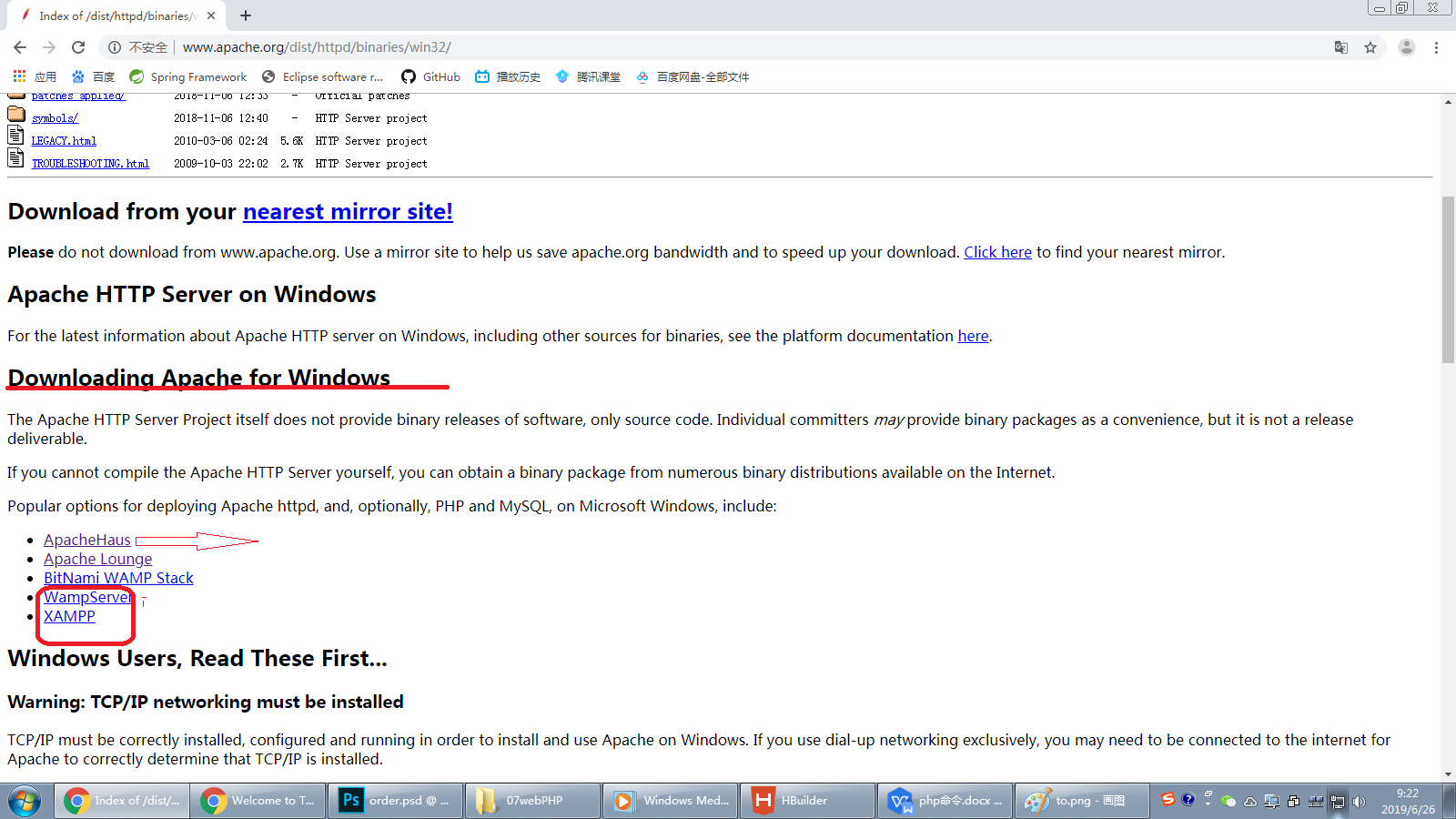
已经不提...编译后的版本只提供源码

下载安装



Ctrl+f 搜索 httpd

进入httpd binarie win32



Vc11指c++环境

解压进步bin目录

Httpd -k install -n Apache ( -n指定服务名)

|  |  |
| --- | --- |
| Testtin.httpd.conf 提示语法错误39行 | Define SRVROOT "D:/peng/apaches/Apache" |
| 80端口被占用了 | listen 8085 也可以在后面追加一个listen 8089 |
| 目录访问权限 | <Directory>Require all granted</Directory> |
| net start Apache | 开启服务 |
| Httpd -t 测试 |  |
| Httpd.conf 38行指定安装路径（是正斜杆） | |
|  | |

浏览器localhost回车



网络基础

|  |  |
| --- | --- |
| ip | Internet Protocol Address ,互联网协议地址 |
| 路由器标识网络设备 | 192.168.1.131 |
| 本地回环地址 | 127.0.0.1 |
| 域名 | 由于ip地址都是没有规律的一些数字组成，很难被人记住，不利于传播，所有就有人提出来，把ip取一个（别名）---域名 |
| DNS服务 | 通过宽带运营商提供了一个服务器，用来专门记录域名对应得ip，这个过程叫寻址  用来帮助你完成DNS寻址的这台机器叫DNS服务器 |
| host | 专门记录域名对应的ip |
| 端口 | 计算机本身就是一个封闭的环境，就好像一个大楼，如果进行数据通信，必须有一个入口，这个门的术语中就叫端口，每一个端口有一个编号  这个端口是你的，别人不许用，所以请个人专门给他看门，这个过程就叫监听，listen |
| Netstat -an | 监听端口的使用情况 |

请求与响应过程

用户在浏览器输入url

浏览器通过DNS服务器，获取当前网站的ip地址

浏览器获取ip地址后，向这个ip发起一个请求

此时服务器就会接收到这个请求，进行响应处理

服务器端处理完的结果返回给客户端的浏览器

浏览器接收的是一堆数据，浏览器把这些数据渲染呈现在界面上

Apache中

解压目录

Define SRVROOT "D:/peng/apaches/Apache"

更改端口

listen 8085

网站根目录

DocumentRoot “D:/pneg/apaches/vv”

<Directory “D:/peng/apache/apaches/vvc”>

访问权限

<Directory “D:/peng/apache/apaches/vvc”>

Require all greanted

</Directory>

默认文档

<IfModule dir\_module>

DirectoryIndex index.html index.php

</IfModule>

测试 httpd -t httpd -M

PHP在Apache中

#加载PHP7版本

LoadModule php7\_module 'D:/peng/apaches/php7/php7apache2\_4.dll'

#加载PHP7配置文件

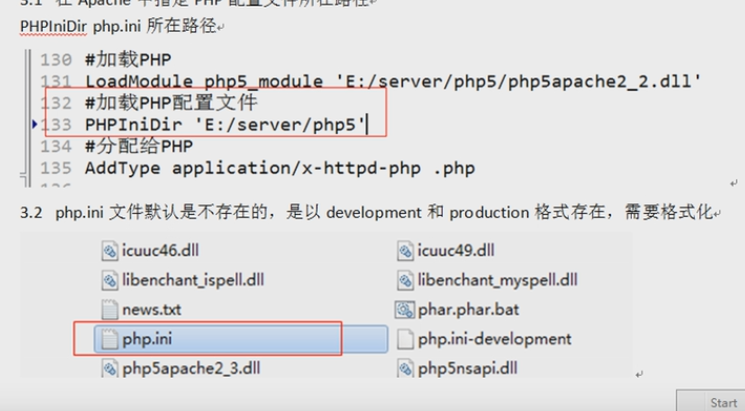
PHPIniDir 'D:peng/apaches/php7'

#增加处理PHP的模块

AddType application/x-httpd-php .php

浏览器测试

<http://localhost:8085/index.php>



MySQL在PHP中php.ini

扩展

Extension=php\_mysqli.dll

扩展路径

extension\_dir = "D:/peng/apaches/php7/ext"

时区

date.timezone =PRC

视频教程 <https://www.bilibili.com/video/av12863134/?p=10>

输出mysql数据库

|  |  |
| --- | --- |
| 关闭服务 |  |
| 程序删除 |  |
| 文件删除 |  |
| regedit注册表删除 |  |

多个网站目录时配置虚拟机

<VirtualHost \*:8085>

ServerAdmin webmaster@dummy-host2.example.com

DocumentRoot "D:/peng/apaches/vvb"

ServerName dummy-host2.example.com

AddDefaultCharset UTF-8

<Directory "D:/peng/apaches/vvb">

AllowOverride all

Require all granted

</Directory>

ErrorLog "logs/dummy-host2.example.com-error.log"

CustomLog "logs/dummy-host2.example.com-access.log" common

</VirtualHost>

180#启动虚拟主机功能

LoadModule vhost\_alias\_module modules/mod\_vhost\_alias.so

引入虚拟主机配置文件位置（默认配置会失效）

Include conf/extra/httpd-vhosts.conf

主要的配置虚拟主机的文件 博客链接

<https://blog.csdn.net/qq_38125058/article/details/81186226>

dummy-host2.example.com

是一种动态开发的脚本语言，动态语言特点是交互性，是面向对象的

超文本预处理器，是一种通用开源脚本语言

脚本在服务器上执行

PHP可免费下载

第一个程序

$contents=file\_get\_contents("target.txt");//文件读取到字符串

$lines=explode("\n", $contents);/\*通过换行分割为数组\*/

<?php foreach($lines as $value){if(!$lines) continue;$cols=explode('|',$value); ?>

<tr>

<?php foreach($cols as $ge){ ?>

<!--去掉空格并加超链接-->

<td>

<?php if(strpos($ge,"http://")==1){ ?>

<a href="<?php echo strtolower($ge);?>" target="\_blank"><?php echo substr($ge,8) ?></a>

<?php }else{ ?>

<?php echo(trim($ge)); ?>

<?php } ?>

</td>

<?php } ?>

</tr>

<?php } ?>

说说echo,print,print\_r,var\_dump的区别

|  |  |
| --- | --- |
| echo | 是带输出功能的语法结构，只能输出简单的数据 |
| print | 是带输出功能的函数，只能输出简单数据 |
| print\_r | 能输出一些复杂的数据 |
| var\_dump | 更详细的输出数据 |
| 注意 | echo不是函数，是语言结构 |
| 了解 | |
| %% | 返回一个百分号 |
| %b | 二进制数 |
| %c | ASCII 值对应的字符 |
| %d | 包含正负号的十进制数（负数、0、正数） |
| %e | 使用小写的科学计数法（例如 1.2e+2） |
| %E | 使用大写的科学计数法（例如 1.2E+2） |
| %u | 不包含正负号的十进制数（大于等于 0） |
| %f | 浮点数（本地设置） |
| %F | 浮点数（非本地设置） |
| %g | 较短的 %e 和 %f |
| %G | 较短的 %E 和 %f |
| %o | 八进制数 |
| %s | 字符串 |
| %x | 十六进制数（小写字母） |
| %X | 十六进制数（大写字母） |
| 测试过 | $s字符串 |
| 使用 | sprintf(“你好%%，%b 向你问好”,3) |

传值与传址的区别

|  |  |
| --- | --- |
| 传值 | $b=$a $b与$a引用内存中的地址是不是同一个 |
| 传址 | $b=&$a $b与$a引用内存中的地址是同一个 |
| 注意 | 传址需要用&符号 |

单引号与双引号的区别

|  |  |
| --- | --- |
| 单引号 | 单引号中字符串出现的变量不会变量的值代替 |
| 双引号 | 双引号中字符串出现的变量会被变量的值代替 |
| 注意 | 双引号中的变量不想被变量的值代替使用大括号把变量包裹起来{} |

php中加号

|  |
| --- |
| +不代表拼接字符串 |
| .代表拼接字符串 |

php中对象命名规则

|  |  |
| --- | --- |
| 对象命名 | class Pic{ var $m=5;function x(){global $m;return $m+3;}} |
| 对象使用 | $n=new Pic; $n->m; $n->x(); |

php与js中不同的遍历对象

|  |  |
| --- | --- |
| js遍历对象 | for(arry in i){ console.log(i) } |
| php中遍历对象 | foreach($arry as $key=>$val){ echo $key.”:”.$val.”,”; } |

时间以及时间戳

|  |  |
| --- | --- |
| time() | 格林威治时间 1970 年 1 月 1 日 00:00:00）到当前时间的秒数 |
| mktime() | 时间戳 mktime(hour,minute,second,month,day,year,is\_dst); |
| date("Y-m-d") | 电脑日期必须指定格式 |
| 电脑日期 | date("Y-m-d") |
| 电脑详细时间 | date("Y/m/d H:i:s",time()+(60\*60\*8));) |
| 注意 | date()函数使用里面得放有格式 为date(“Y-m-d)); |

日期了解

|  |
| --- |
| Y - 年，四位数字; 如: "1999" |
| y - 年，二位数字; 如: "99" |
| m - 月份，二位数字，若不足二位则在前面补零; 如: "01" 至 "12" |
| n - 月份，二位数字，若不足二位则不补零; 如: "1" 至 "12" |
| M - 月份，三个英文字母; 如: "Jan" |
| F - 月份，英文全名; 如: "January" |
| d - 几日，二位数字，若不足二位则前面补零; 如: "01" 至 "31" |
| j - 几日，二位数字，若不足二位不补零; 如: "1" 至 "31" |
| h - 12 小时制的小时; 如: "01" 至 "12" |
| H - 24 小时制的小时; 如: "00" 至 "23" |
| g - 12 小时制的小时，不足二位不补零; 如: "1" 至 12" |
| G - 24 小时制的小时，不足二位不补零; 如: "0" 至 "23" |
| i - 分钟; 如: "00" 至 "59" |
| s - 秒; 如: "00" 至 "59" |
| a - "am" 或是 "pm" |
| A - "AM" 或是 "PM" |
| D - 星期几，三个英文字母; 如: "Fri" |
| l - 星期几，英文全名; 如: "Friday" |
| w - 数字型的星期几，如: "0" (星期日) 至 "6" (星期六) |
| z - 一年中的第几天; 如: "0" 至 "365" |
| t - 指定月份的天数; 如: "28" 至 "31" |
| U - 总秒数 |
|  |

数学 Math

|  |  |  |
| --- | --- | --- |
| 函数 | 理解 | 例子 |
| abs() | 绝对值 | abs(-10)=10 |
| ceil() | 向上取整 | ceil(10.2)=12 |
| floor() | 向下取整 | floor(10.5)=10 |
| intval() | 丢弃小数，保留整数 | intval(12.26)=12 |
| round() | 四舍五入,第二个参数保留小数的个数 | round(4.5,2)=5 |
| sqrt() | 求平方根 | sqrt(9)=3 |
| pow() | n次方 | pow(2,3)=16 |
| pi() | 圆周率 | pi()=3.1515926+535895 |

重点Math

|  |  |  |
| --- | --- | --- |
| max() | 最大值 | max([“A”,”z”])=z |
| min() | 最小值 | min([1,2,3,4,])=1 |
| mt\_rand() | 安全随机数 | mt\_rand(10,100)/100 |

字符串

|  |  |  |
| --- | --- | --- |
| strlen() | 获得字符串长度，空格也算 | strlen(“123456”)=3 |
| explode() | 分割字符串 | explode(“.”,52.23.1,2)=[52,23] |
| trim() | 去除空格 | trim(“dfsfsef”,f)=dsse |
| substr() | 截取字符串 | substr(“abcdefgh”,0,3)=abc |
| str\_ireplace() | 替换字符串 | str\_replace(“a”,”b”,”abcde”)=bbcde |
| strtoupper() | 转大写 | strtoupper(“abc”)=ABC |
| ucfirst() | 第一个字母大写 | ucfirst(explode(“b”,”abc)[0])=A |
| htmlentities() | 打印原汁原味的字符串，标签也不例外 | htmlentities(“<h1>kjj</h1>”)=kjj |
| number\_format() | 数字格式化输出 | number\_format(“123456.12,1,”a”,”b”)=123b456a1 |
| substr\_count() | 返回第二个参数在第一个字符串里出现的次数 | substr\_count(“abcdd”,”ab”)=2 |
| strpos() | 返回第二个参数在第一个字符串里出现的位置 | strpos(“abcde”,”b”)=1 |
| strstr() | 第一次出现的参数到最后 | strstr(“abcde”,”bc”)=bcde |
| strrchr() | 最后一次出现的参数到最后 | strrchr(“abcdebc”,”bc”)=bc |
| addaslasshes() | 将参数二中每一个字符在参数一相同字符前加”\” | addaslasshes(“abc”,”bc”)=a\b\c |
| strcmp() | 依次比较两字符串对应字符阿斯克码值，第一对不一样的，如果参数一里大于参数二里的返回1,反之返回-1，两字符串完全一样返回0 | strcmp(2,1)=1  strcmp(a,a)=0  strcmp(1,2)=-1 |
|  |  |  |
|  |  |  |
| 注意 | str\_ireplace()才不区分大小写 |  |
|  |  |  |

重点

|  |  |
| --- | --- |
| hmtlentities() | strpos() |
| number\_format() | strrchr() |
| ucfirsrt() | explode() |
| strlen() | strstr() |
| substr() | trim() |

指针的概念

|  |
| --- |
| 程序从上到下，从左到右的运行，到数组时，指针指向数组中的键值对，从左到右。 |
| 指针也指程序中的运行顺序。 |

数组

|  |  |  |
| --- | --- | --- |
| 分类 | 特征 | 例子 |
| 索引数组 | 跟js数组一样 | [12,13,14,15,16,17,18,19] |
| 关联数组 | 有箭头指向右边,类似js的对象 | [“name”=>”张三”,”age”=>14] |

数组函数

|  |  |  |
| --- | --- | --- |
| count() | 返回数组的个数 | count([12,45,3])=3 |
| implode() | 拼接数组类似js中join() | implode(“-”,[12,15,45])=12-15-45 |
| in\_array() | 判断第二个参数是否有第一个参数元素 | in\_array(“15”,[12,15])=true |
| array\_key\_exists() | 判断第二个参数是否有第一个参数额键值，返回真假 | array\_key\_exists(“k1”,[“k1”=>12])=true |
| array\_unique() | 原数组中多余重复元素剔除掉的新数组，原数组不改变s | array\_unique([12,12,13,15])=[12,13,15] |
| current($arr0) |  |  |
| array\_combine() | f返回将第一个参数数组中每一个元素值依次作为键值付给参数二数组的数组，两数组长度必须一致，原数组  不改变 | array\_combine(["a","b","c"],[1,2,3])= |
| shuffle($arr) | 将原数组元素与对应键值的对应关系重新随机排列返回真假 | shuffle() |
| range() | 将第一个参数作为第一个元素，每次自增参数三的值，自增后再作为一个元素存在数组中，直到值达到参数二的值存到数组中为止并返回这个数组 | range(3,9,2)=9 |
| array\_sum | 计算所有数值型元素的和 | array\_sum([“a”,12,”b2”] |
| sort() | 将原数组键值打破重新以元素值的阿斯克码值从小到大排序，索引从数字0开始重计 | sort([1,122,1,3])=[1,1,3,122] |
| rsort() | 以元素值阿斯柯码值大小降序重新排序，索引从0重新计 | sort([1,122,1,3])=[122,3,1,1] |
| array\_merge() | 将两个数组合并并返回原数组不变 |  |
| array\_splice() | 截取功能和array\_slice()一样，只是将截取部分在原数组中剔除 | $arr=[12,13,14,15];  array\_splice($arr,2,1);  var\_dump($arr);=[12,13,15] |
| array\_slice() | 在第一个参数数组中从第二个参数数值位置开始截取到第三个参数数值长度的数组键值+元素并返回，数组 | $arr=[12,13,14,15];  var\_dump(array\_slice($arr,2,1));=[14] |
| array\_pad() | 让第一个参数数组达到第二个参数数值长度，将第三个参数作为元素添加到第一个参数数组的末尾，索引从  最小没用过数值开始计并返回，原数组不改变 | $arr=[12,13,14,15];  var\_dump(array\_pad($arr,5,"kjj"));=[12,13,14,15,”kjj”] |
| end() | 将数组指针指向最后一位元素，并返回最后一位元素值 | $arr=[12,13,14,15];  var\_dump(end($arr));=15 |
| reset() | 数组指针重置指向第一个元素并返回元素值 | $arr=[12,13,14,15]  reset($arr)=12 |
| prev() | 将数组指针推向上一位，再返回指针移动后指向的元素值 | $arr=[12,13,14,15];  var\_dump(prev($arr));=false |
| next() | 先将数组指针推向下一位，再返回指针移动后指向的元素值 | $arr=[12,13,14,15];  var\_dump(next($arr));=15 |
| each() |  |  |
| current() | 返回当前数组指针指向的元素值 | $arr=[12,13,14,15];  var\_dump(current($arr)); |

重点

|  |  |  |
| --- | --- | --- |
| sort() | array\_pop() | array\_unshift |
| rsort() | array\_push() | array\_unique() |
| array\_slice() | array\_splice() |  |
|  |  |  |

符串与json 数组对象的关系

|  |  |
| --- | --- |
| javascript中字符串与json 数组对象的关系 | |
| 字符串转json数组 | JSON.parse(str) |
| 数组对象装json字符串 | JSON.stringify(arry) |
| php中字符串与json数组对象的关系 | |
| 字符串转json数组 | json\_decode($str,JSON\_UNESCAPED\_UNICODE) |
| 数组对象装json字符串 | json\_encode($arry,JSON\_UNESCAPED\_UNICODE) |
| 注意 |  |

json中文输出显示编码是二进制，应加上第二个，JSON\_UNESCAPED\_UNICODE，利润

$arr1=array("中文","b","c");

$str\_arr=json\_encode($arr1,JSON\_UNESCAPED\_UNICODE);

文件操作

|  |  |  |
| --- | --- | --- |
| fopen() | 打开文件 | fopen(“json.txt”,”a+”) |
| fcloce() | 关闭文件 | fcloce(“json.txt”) |
| fwrite() | 写入文件 | fwrite(“json.txt”,”123456”) |
| fread() | 读取文件 | fread(“json.txt”,filesize(“json.txt”)) |
| fileseze() | 文件大小 | fread(“json.txt”,filesize(“json.txt”)) |
| is\_file() | 判断是否为文件 |  |
| feof() | 判断是否读取到底部 | while(!feof(“json.txt”)){echo fgets($fh),”<br/”; } |
| fgets() | 逐行读取文件 |  |
| file() | 将文件读取到数组中，各元素由换行符分隔 | $arr=file(“json.txt”);  print\_r($arr) |
| file\_put\_contents() | 字符串写入文件 | $str = "hello";  file\_put\_contents("test.txt", $str); |
| file\_get\_contents() | 读取文件字符串 | file\_put\_contents("test.txt", $str); |

打开文件权限

|  |  |
| --- | --- |
| r | 打开文件为只读。文件指针在文件的开头开始。 |
| w | 打开文件为只写。删除文件的内容或创建一个新的文件，如果它不存在。文件指针在文件的开头开始。 |
| a | 打开文件为只写。文件中的现有数据会被保留。文件指针在文件结尾开始。创建新的文件，如果文件不存在。 |
| x | 创建新文件为只写。返回 FALSE 和错误，如果文件已存在。 |
| r+ | 打开文件为读/写、文件指针在文件开头开始。 |
| w+ | 打开文件为读/写。删除文件内容或创建新文件，如果它不存在。文件指针在文件开头开始。 |
| a+ | 打开文件为读/写。文件中已有的数据会被保留。文件指针在文件结尾开始。创建新文件，如果它不存在。 |
| x+ | 创建新文件为读/写。返回 FALSE 和错误，如果文件已存在。 |

其它文件操作

|  |  |  |
| --- | --- | --- |
| copy() | 将文件从 source 拷贝到 dest。如果成功则返回 TRUE，失败则返回 FALSE。 | copy(“json.txt”,”min.txt”)  copy ( string source, string dest ) |
| unlink() | 删除文件，如果删除成功返回true, 否则返回false; | unlink ( string filename ) |
| file\_exists() | 检查文件或目录是否存在，如果存在返回true, 否则返回false |  |

文件目录操作

|  |  |  |
| --- | --- | --- |
| opendir() | 打开目录 |  |
| closedir() | 关闭目录 |  |
| readdir() | 返回由dir\_handle指定目录中的各个元素。可以使用此函数列出给定目录中的所有文件和子目录 |  |
| scandir() | 返回一个包含有 directory 中的文件和目录的数组; |  |
| rmdir() | 删除文件夹时需要保证文件夹是空的，即里面没有任何文件，如果有则删除失败。 |  |
| mkdir() | 尝试新建一个由 pathname 指定的目录。 | mkdir ( string pathname) |
| is\_dir() | 判断是否为目录 |  |

注意点

|  |
| --- |
| scandir()返回的是一个包含文件，及文件夹的数组，数组从根目录开始，一个点根目录，两个点上一级目录。 |
| fwrite()写入函数必须处在fopen()与fclose()的范围之间 |
| file\_put\_contents()函数跟fwrite类似 |

删除文件夹目录

function remodir($val){

$scan=scandir($val);

for($i=2;$i<count($scan);$i++){

if(is\_file($val."/".$scan[$i])){

unlink($val."/".$scan[$i]);

}else{

remodir($val."/".$val[$i]); //是目录继续调用函数来删除文件

}

}

/\*删除完，并清除空目录\*/ rmdir($val);

}

remodir("yun");

文件输出

while(!feof(“json.txt”)){ }

while(($str=fgets(“json.txt”)){ }

GET与POST表单 get与post

|  |
| --- |
| get |
| get表单输入的数据被加载到URL地址后面 如 test.php?user=pengchihui&pass=23ji |
| get方式提交数据只能传递文本，能够提交的数据量大小有限，安全性差 |
|  |

|  |
| --- |
| post |
| post提交数据的方式把表单的数据打包放入http请求中 |
| post能够提交更多的数据 如上传文件，图片 |

处理多值表单

|  |
| --- |
| 需要对表单名（元素的name属性值）增加一堆中括号并且指定value值，如 |
| <input type=”checkbox” name=”hobby[]” value=”夏明”/> |

PHP设置 php.ini 集成工具 wampserver的PHP中，php.ini文件

|  |  |
| --- | --- |
| file\_uploads | 是否允许通过HTTP上传文件，默认为ON |
| upload\_max\_filesize | 允许上传文件大小的最大值，默认为2M，此指令必须小于post\_max\_size; |
| upload\_tmp\_dir | 指定上传文件的临时存放路径，这个目录对于拥有此服务器进程的用户必须是可写的;如果未指定则使用系统默认值; |
| post\_max\_size | 控制POST方式提交数据php所能够接收的最大数据量; |
| memory\_limit | 指定单个脚本程序可以使用的最大内存容量; |
| max\_execution\_time | 此指令确定php脚本可以执行的最长时间，以秒为单位，默认为30秒; |

$\_FILES 特性

|  |  |
| --- | --- |
| form 允许上传 enctype="multipart/form-data" | |
| $\_FILES 是一个二维数组，数组中有项 | |
| $\_FILES超级全局变量作用是存储各种与上传文件有关的信息; | |
| $\_FILES["userfile"]["name"] | 上传文件的名称 |
| $\_FILES["userfile"]["type"] | 上传文件的类型 |
| $\_FILES["userfile"]["size"] | 上传文件的大小, 以字节为单位 |
| $\_FILES["userfile"]["tmp\_name"] | 文件上传后在服务器端储存的临时文件名 |
| $\_FILES["userfile"]["error"] | 文件上传相关的错误代码 |

file文件上传 特性

|  |  |  |
| --- | --- | --- |
| move\_uploaded\_file | 作用是将上传文件从临时目录移动到目标目录(返回bool值) | move\_uploaded\_file ( string filename, string destination ) |
| 虽然copy()也可以实现同样功能，但move\_uploaded\_file()还提供了一种额外的  功能，它将检查由filename输入参数指定的文件确实是通过http post 上传机制  上传的。如果所指定的文件并非上传文件，则移动失败，返回false; | | |

文件上传状态 $\_FILES[“picture”][“error”]==”UPLOAD\_ERR\_OK”

|  |  |
| --- | --- |
| UPLOAD\_ERR\_OK (value = 0) | 如果文件上传成功返回0; |
| UPLOAD\_ERR\_INI\_SIZE (value = 1) | 如果试图上传的文件大小超出了 upload\_max\_filesize指令指定的值，则返回1; |
| UPLOAD\_ERR\_FORM\_SIZE (value = 2) | 如果试图上传的文件大小超出了MAX\_FILE\_SIZE指令（可能嵌入在HTML表单中）指定的值，  则返回2; |
| UPLOAD\_ERR\_PARTIAL(value = 3) | 如果文件没有完全上传，则返回3; 如网络出现错误，导致上传过程中断; |
| UPLOAD\_ERR\_NO\_FILE (value = 4) | 如果用户没有指定上传的文件就提交表单，则返 |
|  |  |

form注意

|  |
| --- |
| 文件上传请求方式post,并允许上传 nctype="multipart/form-data" |
| 以键值对的方式传输，用name指定键 |
| checkbox name相同并有中括号，并有value值 |
|  |

防止重命名

|  |  |  |
| --- | --- | --- |
| uniqid() | 相同的内容，加密后不同 |  |
| md5() | 相同的内容，加密后相同 |  |
| time()+文件名 |  |  |

文件上传页面中文编码

$url="file/".time().$\_FILES["pic"]["name"];

$url=iconv('utf-8','gb2312',$url);

原生ajax传值 post

var xhr = new XMLHttpRequest();

xhr.open("post", "04.php", true); xhr.setRequestHeader("Content-type","application/x-www-form-urlencoded");

xhr.onload = function() {

console.log(this.responseText)

}

xhr.send("user="+inp[0].value+"&pass=45")

php重定向方式

|  |  |
| --- | --- |
| header(‘Location: http://www.baidu.com’) | 用的最多 |
| <script>window.location.href="http://www.baidu.com"</script> |  |
| <meta http-equiv="refresh" content="1;url=http://www.baidu.com"> |  |

序列化与反序列化

|  |  |
| --- | --- |
| 序列化(json\_decode) | 将json格式字符串转换成对象的过程 |
| 序列化json\_encode | 将变量，字符串转换成json数据 |

增加列表注意点

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

$\_SERVER['REQUEST\_METHOD']==='POST'empty($\_POST['title'];

isset($erro\_message)?null:$erro\_message;

$music['error']==0

php操作数据库

连接与关闭

$con=mysqli\_connect("localhost","root","","test");

mysqli\_close($con);

连接失败

echo mysqli\_connect\_error();

增加

$user=[

[ "name"=>"张三", "age"=>"19", "sex"=>"男", "id"=>"1"],

[ "name"=>"李四", "age"=>"20", "sex"=>"男", "id"=>"3"],

[ "name"=>"王五", "age"=>"21", "sex"=>"男", "id"=>"4"],

[ "name"=>"赵六", "age"=>"22", "sex"=>"男", "id"=>"5"]

];

for($i=0;$i<count($user);$i++){

$str="INSERT INTO `people`(`id`, `name`, `age`, `alive`) VALUES ('".$user[$i]["id"]."','".$user[$i]["name"]."','".$user[$i]["age"]."','1')";

mysqli\_query($con, $str);

}

修改

$str="UPDATE `people` SET `name`='小张三' WHERE `id`='2'";

mysqli\_query($con, $str);

查询

$str="SELECT `id`, `name`, `age`, `alive` FROM `people` WHERE `alive`='0'";

$asd=mysqli\_query($con, $str);

while($sql=mysqli\_fetch\_assoc($asd)){

var\_dump($sql);

}

mysqli\_free\_result($asd);

删除

$str="DELETE FROM `people` WHERE `id`='1'";

mysqli\_query($con, $str);

1.插入数据库编码问题

修改配置文件my.ini,在最后加上character-set-server=utf8

2.解决中文显示问题

json文件转对象，还是数组对象

json\_decode(file\_get\_contents("music.txt"),true);

3.php传值id,与接收

?id=<?php echo $value ?>

$\_GET['id'];

4.删除id出错，由于id后面的值没有用单引号进行括起来

"DELETE FROM `singer` WHERE id='".$id."'";

1. 重定向，header，一个空格

header('location: http://localhost/music/shows.php');

exit;

1. 字符串存入文本编码问题
2. 修改显示，file属性

<?php echo "value='".$result['music']."'"; ?>