<!DOCTYPE **html**>  
<**html**>  
<**head**>  
 <**title**>silite</**title**>  
</**head**>  
<**body bgcolor = "#ffefd5"**>  
  
**<?php**   
**?>**</**body**>  
</**html**>

**加在 <?php > 中间**

**双引号所包含的变量会被替换 单引号原本输出 效率单引号最高**

**用<p>段落标记**

**用<br>来当\n**

**用<<<来自定义标识符 结束符必须顶格 有;**

**定义数组用圆括号 键和值之间用=>**

**Null不区分打小写**

**用is\_\*\*\*\*() 来判断数据的类型 返回布尔值**

**转换类型和c一致 也可以用settype(var,type); 此函数直接改变原值**

**用Is\_numeric()检测变量中的数据是否为数字**

**用define()来定义宏 第三个参数为true 对大小写不敏感**

**用defined()来判断宏是否被定义**

**用constant\_name()来动态输出**

**用function example() 默认参数要放在定义参数的右边 否则可能会出错**

**用static 定义静态变量 c一致**

**在自定义函数中使用全局变量 需用global来重新声明该全局变量才能使用**

**可变变量用两个$来使用 类似c的指针**

**除余时被除数是负数 结果为负**

**用 === 和 ！== 来判断恒等 元素类型要一致**

**符号逻辑优先级(||和&&) 大于 (or和and) 的优先级 同级所有且大于或优先级**

**命名 类属性前用m 类名不用下划线 引用(&)变量前用r 全局变量前用g 静态变量前用s**

**\* print\_r 只用于数组 带键值输出**

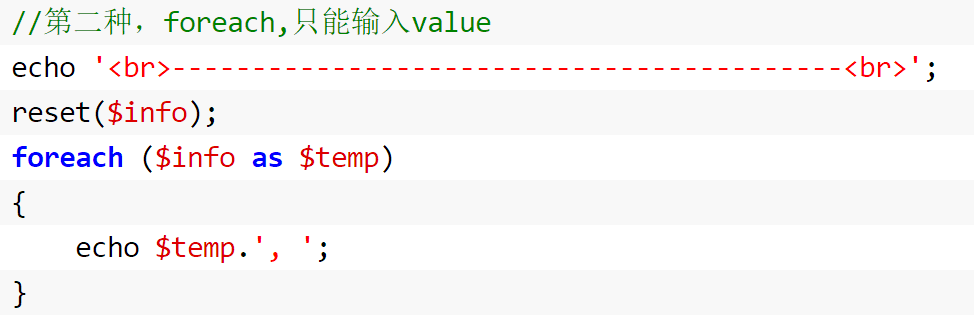
**\* print 只能输出一个字符串 成功返回true 所以速度稍慢**

**\* echo 更灵活 能输出一个以上的字符串**

**\* sprintf 把所有东西放在一个变量里 再用其他输出**

**\* printf 跟c一致 格式化输出**

**\* var\_dump 可以任意输出 常用作于调试**



**用rand()伪随机**

**用break num;来选择跳出的层数**

**用定界符<<<的话 无需转义**

**\*用[l,r]trim(str[,charlist])去除(左右)首尾字符 charlist可按需过滤**

**\*用addslashes()将单双引号 反斜线 NUL(NULL)字符前加反斜线**

**用stripslashes()反转义**

**用 addcslashes(str,[charlist]) 将指定字符转义 默认全部转义 一般对缓存数据的值采用此函数**

**用 stripcslashes() 将上个函数转义的字符串还原**

**用strlen()获得字符串长度 同C**

**用count(arrary) 和sizeof() 获得数组长度**

**用count(array,COUNT\_RECURSIVE) 递归获得多维数组长度**

**用substr(str,start[,length])来截取制定字符串的长度 支持类py负数操作 也可以将超出页面限制用.链接...**

**用 strcmp()和stecasecmp() 来比较两个字符串 同C 后者不区分大小写**

**用strnatcmp()和strnatcmp()自然比较 2比10小 后者不区分大小写 前者反之**

**用strncmp()比较前n个字符**

**用strstr(haystack,needle)检索needle 可以限制图片上传格式**

**用strrchr()检索最后一个needle 功能同上**

**用substr\_count(haystack,needle)检索needle出现的次数 类搜索引擎**

**用str\_ireplace(search,replace,subject[,count])和str\_replace()替换字符串 search支持array 要搜索的值 前者不区分大小写**

**可用上述函数来替换字体颜色 replace = “<font color = ‘blue’>”.search.”</font>”**

**用substr\_replace(原始字符串(支持array),需求替换,起始位置[,长度(0表示插入)]) 替换长度待定字符串**

**用number\_format(num[,小数位数[,制定小数点的字符[,千分符]]])格式化字符串 同py 可1,2,4个参数 默认取整加千分符**

**用explode(边界,str[,返回个数]) 同py split() 正数为分割次数 负数返回除了-limit个数数组**

**用implode(边界,array)合成字符串**

**用array(‘key’=>value)建立关联数组 同py的字典**

**用foreach( as )遍历数组 同py 操作时类似py的贴片**

**用while(list($name,$value) = each(array)) 遍历数组 each()用于返回当前位置的数组值**

**用array\_pop() 同py pop()**

**用array\_search(needle,array[,bool]) bool检查搜索的类型**

**用isset()检测变量是否被设置 用empty()检测变量是否为空**

**用array\_push(array,var) 压入数据 同py expend() 可同时添加多个值**

**用array\_unique()删除数组重复元素**

**用phpinfo();检查php内容**

**表单创建<form> 里enctpye=’multipart/form\_data’ 实现多文件上传**

**用sort()将制定数组进行升序排列 同py**

**<form> 表单中 action 发送数据的位置target 中\_blank 新网页打开 \_parent 显示在父级窗口**

**\_self 当前窗口显示 \_top 顶级窗口显示 cellpadding=’’ 单元格边距 cellspacing=’’单元格间距**

**用&nbsp占位符 类似空格**

**本地到服务器 POST安全 服务器到本地 大多用GET GET使用url或cookie传参 POST将数据放在body中 所以POST安全 GET的url会有长度限制 POST可以非常大 url过长对服务器有很大的负担**

**用<input type = ‘radio’ checked> 设置表单按钮 同类name应一致 checked为默认值**

**用<input type = ‘text’> 设置文本输入框**

**用<input type = ‘password’> 设置密码输入框**

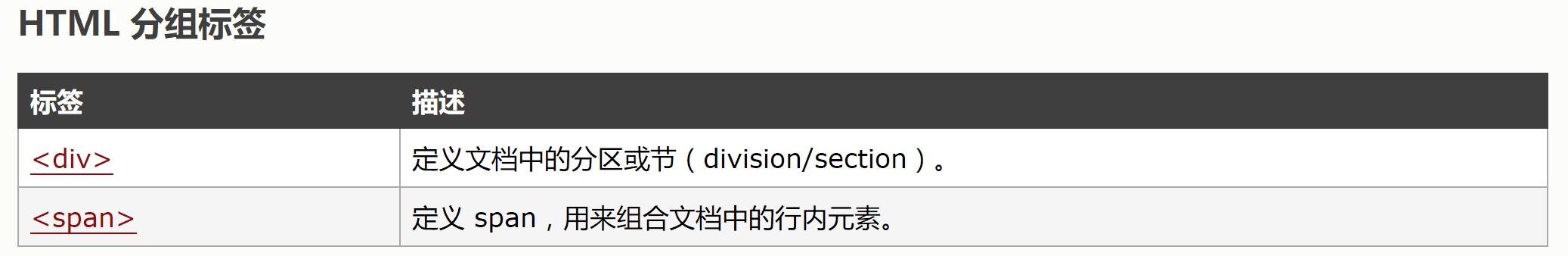
**用<input type = ‘submit’> 设置提交按钮**

**用<input type = ‘checkbox’ size=’’> 设置复选框 同类name应一致**

**用<input type = ‘reset’> 设置重置按钮**

**if**(**isset**($\_POST[**'array'**]) && $\_POST[**'array'**] != **null**)  
 **for**($i = 0;$i<*count*($\_POST[**'array'**]);$i++)  
 **echo** $\_POST[**'array'**][$i];

**用<select size=’’> 设置下拉列表 size大于1 为列表框 元素多于size 自动设置滚动条**



**用<input type = ‘file’> 设置文件上传**

**用<a>做超链接 目的链接用href=’’**

**用**date\_default\_timezone\_set(**'PRC'**);  
**echo** *date*(**'H:i:s'**,*mktime*()); **更改时区**

**用mktime() 和 time()输出时间戳**

**用date([ymdhis],mktime())输出指定格式时间**

**用strtotime()进行时间计算**

**用setcookie()创建cookies 参数expire设置失效时间 path为/ 整个domain内都有效 domain为cookie的有效域名 secure为1 只能在https上有效 失效在关闭浏览器前有本地记录**

**用上代码删除cookie setcookie(name,””,time()-1)**

**用$\_COOKIE[]调用cookie 也可以获取value**

**用session\_start()在页面开始位置调用 会话变量被登录到数据$\_SESSION**

**用session\_register() 会隐含的调用上述函数**

**用$\_SESSION[‘name’] = null; 来声明一个名为admin的变量 并设空**

**判断session是否为空用empty()**

**用unset($\_SESSION[‘user’]) 注销单个会话**

**用$\_SESSION = array(); 注销多个会话**

**用session\_destroy(); 结束会话**

**用session\_set\_cookie\_params($time); 设置session失效时间 不推荐**

**用setcookie(session\_name(),session\_id(),time()+$time,”/”) 手动设置失效时间 session\_id是判断客户端用户的标识 因为session\_id是随机产生的**

**用$\_GET[session\_name] 来取得session\_id session\_id(id\_name) 关键步骤**

**用session\_save\_path() 存储session临时文件 缓解服务器压力 在start前调用**

**用session\_cache\_limiter(‘public’ or ‘private’) 来写入session缓存 在客户端**

**用session\_cache\_expire() 设置缓存时间 单位是分钟**

session\_cache\_limiter(**'private'**);  
$chache\_limit = session\_cache\_limiter();  
*session\_cache\_expire*(30);  
$cache\_expire = *session\_cache\_expire*();  
*session\_start*();

**开启客户端缓存**

**用session\_set\_save\_handler() 建立session缓存数据库 open(path,name)找到地址 取出变量值 close()不需参数 关闭数据库 read(key) 读取session键值 key对应session\_id write(key,data) data对应设置的session变量 destroy(key) 注销session对应session的键值 gc() 清除过期的session记录**

**用resource fopen(path,mode[,use\_include\_path]) 打开文件 第三个希望服务器在这个路径下打开所指定的文件 可以设置1 or true #$fopen = fopen($filepath,’a’) or die(‘文件不存在’);**

**文件同C**

**用flose() 关闭文件**

**用readfile() file() file\_get\_contents(filename[,offset[,maxlen]]) 读取整个文件**

**Readfile() 不需要关闭文件 file() 按行存放在数组内 后者读入字符串 适用于读取二进制**

**用fgets(filename[,len]) 一次读取一行数据**

**用fgetss()同上 可以滤掉html和php标记 可以使用allowable\_tags指定不被过滤的标记**

**用fgetc()读取一个字符**

**用fread(filename,len)读取任意长度字符**

**用fwrite(handle,str[,len]) 和 file\_put\_contents(filename,data[,flags])写入文件 使用后者相当于依次调用**

**Fopen() fwrite() fclose()**

**用flock() 锁定文件**

**创建类时无() 初始化类有() 用new字段**

**用const var = value 定义类常量**

**用 类名::常量名 输出类常量**

**用\_\_construct () 初始化对象 构造方法**

**class** Object{  
 **public $var1**;  
 **public $var2**;  
 **public function** \_\_construct($var1,$var2)  
 {  
 $this->**$var1** = $var1;  
 $this->**$var2** = $var2;  
 }  
 **public function** name(){  
 }  
}

**用\_\_destruct() 销毁对象 析构方法 一般不需要主动使用**

**用class 子类 extends 父类{} 继承**

**用private声明的类变量或方法 只能在类内部被访问**

**用protected声明的类变量或方法 可以在子类使用 其他地方不可以使用**

**用self::成员 调用静态成员 访问量类似**

**用final声明的类 不能被继承 也不能有子类 声明的方法 子类中不可重写 不可覆盖**

**用abstract声明的类 只能作为父级使用 抽象类必须至少包含一个抽象方法 抽象方法没有方法体**

**使用;结束声明 功能只能在子类中实现**

**用interface声明接口 实现多继承 类中只能包含未实现的方法和一些成员变量 继承时用 implements 实现接口 所有方法均需在子类中全部实现**

**\*实例化对象互相赋值 类似于& 实现复制功能 需用clone**

**用魔法方法\_\_clone 在类中声明方法 克隆时能保持一些自己的属性 并非全部依附于被克隆对象**

**用instanceof检测对象属于哪一类**

**魔术方法~：**

**\_\_set和\_\_get:当程序试图写入一个不存在或不可见变量时 \_\_set包含两个参数 分别表示变量名称和变量值 二者不可省略**

**当程序调用一个未定义或不可见的成员变量时 可以通过\_\_get获取变量值 \_\_get有一个参数 表示要调用的变量名**

**使用魔术方法时 要在类中声明 使用方法类似py**

**魔术方法均用public修饰**

**\_\_call方法类似py 类中无调用对象时 优先进入此方法 两个参数 第一个是方法名 第二个是方法的个数**

**用serialize()函数可以序列化对象 此时实例化对象包含\_\_sleep方法 会优先执行 \_\_sleep方法的目的是关闭对象可能具有的数据库连接等类似的善后工作**

**用unserialize()还原被序列化的对象 \_\_wekeup方法则是恢复在序列化中可能丢失的数据库连接及相关工作**

**\_\_toString 魔法方法 当时用echo或print输出对象时 将对象转化为字符串 如果没有此方法 直接输出对象会发生致命错误(fatal error) 输出时 只能输出对象 不能掺杂其他字符**

**\_\_autoload 当程序需要用到一个类 但该类还没有被实例化 使用此魔法方法 在制定的路径下自动查找和该类名称相同的文件 找不到 会报错**

**加密:**

**用crypt(str,salt)完成单向加密功能 salt为干扰串**

**用md5(str[,bool]) bool为true 则函数返回一个二进制式的密文 默认为false**

**注册用户的密码都是先用MD5加密 然后再保存到数据库中的 用户登陆时 程序把用户输入的密码计算成**

**MD5的值 进行比较**

**用sha1() 安全哈希算法 结构同上 true返回一个20位二进制数 false返回40位**

**用Mcrypt进行加密解密操作时 首先要创建一个初始化向量(iv) 用mcrypt\_create\_iv(int size[,int sourse])**

**其中sourse可取MCRYPT\_RAND 系统随机数 MCRYPT\_DEV\_RANDOM 和\_URANDOM 都是UNIX系统的**

**初始化向量后 用mcrypt\_encrypt(加密算法,密钥,需要加密的数据,算法模式,初始化向量)**

**用同上\_decrypt解密**

**CUDR 数据库操作**

**ADODB**

**用$mysqli = new mysqli("localhost", "root", "password", "testgame");**

**$mysqli->close();  来关闭数据库**

**<?php** $dbms = **'mysql'**;  
 $dbName = **'eb\_borrow\_books'**;  
 $host = **"localhost"**;  
 $user = **"root"**;  
 $password = **"silite"**;  
 $dsn = **"**$dbms**:host=**$host**;dbname=**$dbName**"**;  
 **try** {  
 $connID = **new** PDO($dsn , $user, $password);  
 **echo "right"**;  
 }**catch**(Exception $e){  
 **echo** $e->getMessage().**"<br>"**;  
 }  
**?>**