奥商集团面试总结

Php:

1. 中英文字符串如何计算字符串长度

在字符编码中，英文字符占一个字节,而中文字符gbk占两个字符,utf8占三个字符,

Php常用计算字符串长度就是strlen()函数,不过,它计算的是字节的长度而非字符的长度,那么如何获取一个字符串中字符的长度呢？还有有mb\_strlen(). 还可以使用正则表达式将字符分割以后进行统计，

* $str = 'PHP点点通';

echo strlen($str); //3\*1+3\*3=12    
echo mb\_strlen($str, 'gb2312'); //3\*1+3\*2=9    
echo mb\_strlen($str, 'utf-8'); //6

注意，mb系列函数不是php核心函数，默认没有开启

* 利用正则表达式进行字符统计

<?php    
function \_strlen($str)    
{    
        preg\_match\_all("/./us", $str, $matches);    
        return count(current($matches));    
}    
    
echo \_strlen("PHP点点通");  //6    
?>

* 补充知识关于中英文混排字符串的解决办法

使用正则最快

<?php  
echo $str = "43fdf测试fdsfadaf43543543职工问防盗锁防盗锁5345gfdgd";  
preg\_match\_all("/[0-9]{1}/",$str,$arrNum);  
preg\_match\_all("/[a-zA-Z]{1}/",$str,$arrAl);  
preg\_match\_all("/([/x{4e00}-/x{9fa5}]){1}/u",$str,$arrCh);  
echo "<pre>";  
echo "数字个数:".count($arrNum[0])."<br/>";  
echo "字母个数:".count($arrAl[0])."<br/>";  
echo "中文个数:".count($arrCh[0]);  
?>

* 关于快速判断字符串长度的办法

If(!isset($str{12}))判断字符串是否大于12个字节

strlen()函数函数执行起来相当快，因为它不做任何计算，只返回在zval 结构(C的内置数据结构，用于存储PHP变量)中存储的已知字符串长度。但是，由于strlen()是函数，多多少少会有些慢，因为函数调用会经过诸多步骤，如字母小写化(译注：指函数名小写化，PHP不区分函数名大小写)、哈希查找，会跟随被调用的函数一起执行。  
在某些情况下，使用isset() 技巧可以加速执行你的代码。因为与isset()作为一种语言结构，意味着它的执行不需要函数查找和字母小写化。也就是说，实际上在检验字符串长度的顶层代码中你没有花太多开销。  
  
因此调用isset()比strlen()快。

1. 知道PSR（PHP代码规范）么？ 说几条

<https://github.com/PizzaLiu/PHP-FIG> PHP编码规范

举例PSR-1 基本代码规范

PHP代码文件**必须**以 <?php 或 <?= 标签开始；

PHP代码文件**必须**以 不带BOM的 UTF-8 编码；

PHP代码中**应该**只定义类、函数、常量等声明，或其他会产生 从属效应 的操作（如：生成文件输出以及修改.ini配置文件等），二者只能选其一；

1. Php反射

是什么： 反射是指在PHP运行状态中，扩展分析PHP程序，导出或提出关于类、方法、属性、参数等的详细信息，包括注释。这种动态获取信息以及动态调用对象方法的功能称为反射API。

怎么用：

$student=new person();

$student->name='Tom';

$student->gender='male';

$student->age=24;

// 获取对象属性列表

$reflect = new ReflectionObject($student);

$props　= $reflect->getProperties();

foreach ($props as $prop) {

  print $prop->getName() ."\n";

}

// 获取对象方法列表

$m=$reflect->getMethods();

foreach ($m as $prop) {

  print $prop->getName() ."\n";

}

使用场景及优缺点：

在平常开发中，用到反射的地方不多：一个是对对象进行调试，另一个是获取类的信息。在MVC和插件开发中，使用反射很常见，但是反射的消耗也很大，在可以找到替代方案的情况下，就不要滥用。

PHP有Token函数，可以通过这个机制实现一些反射功能。从简单灵活的角度讲，使用已经提供的反射API是可取的。

很多时候，善用反射能保持代码的优雅和简洁，但反射也会破坏类的封装性，因为反射可以使本不应该暴露的方法或属性被强制暴露了出来，这既是优点也是缺点。

Mysql:

1. mysql优化说几条

使用limit

使用 ENUM 而不是 VARCHAR

为每张表设置一个ID

为搜索字段建索引

 当只要一行数据时使用 LIMIT 1

1. Int(1)和int(10)的区别

INT[(M)] [UNSIGNED] [ZEROFILL]

普通大小的整数。带符号的范围是-2147483648到2147483647。无符号的范围是0到4294967295。

INT(1) 和 INT(10)本身没有区别，但是加上(M)值后，会有显示**宽度**的设置。

如代码所示：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | mysql> create table test(id int(3));  Query OK, 0 rows affected (0.47 sec)  mysql> insert into test values(12);  Query OK, 1 row affected (0.12 sec)  mysql> insert into test values(1234);  Query OK, 1 row affected (0.10 sec)  mysql> select \* from test;  +------+| id   |+------+|   12 || 1234 |+------+ |

 加上zerofill。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | mysql> create table test1(id int(3) zerofill);  Query OK, 0 rows affected (0.32 sec)  mysql> insert into test1 value(12);  Query OK, 1 row affected (0.07 sec)  mysql> insert into test1 value(1234);  Query OK, 1 row affected (0.05 sec)  mysql> select \* from test1;  +------+| id   |+------+|  012 || 1234 |+------+ |

 这下注意12前面输出多了个0，int(M) 的值多了个0，这就是显示宽度的限制。而多出来的还会显示出来。只是系统判定12显示宽度不足，会补0来补全显示宽度

但是要注意插入负数的时候：

没有设置zerofill的时候负数正常显示

|  |  |
| --- | --- |
| 1  2  3  4 | mysql> insert into test value(-1234);  Query OK, 1 row affected (0.07 sec)  mysql> select \* from test;  +-------+| id    |+-------+|    12 ||   123 || -1234 |+-------+3 rows in set (0.00 sec) |

 再来看看设置 zerofill的时候：

|  |  |
| --- | --- |
| 1  2  3  4 | mysql> insert into test1 value(-1234);  Query OK, 1 row affected, 1 warning (0.11 sec)  mysql> select \* from test1;  +------+| id   |+------+|  012 || 1234 ||  000 |+------+ |

 输出为000，插入是-1234 。显示是000。  
 **原来添加zerofill的时候系统会给自动添加上unsigned属性。就是非负数。而设置的显示宽度为3位。所以就会输出000。**

1. 手写添加一个普通索引

1.添加PRIMARY KEY（主键索引）   
mysql>ALTER TABLE `table\_name` ADD PRIMARY KEY ( `column` )   
2.添加UNIQUE(唯一索引)   
mysql>ALTER TABLE `table\_name` ADD UNIQUE (   
`column`   
)   
3.添加INDEX(普通索引)   
mysql>ALTER TABLE `table\_name` ADD INDEX index\_name ( `column` )   
4.添加FULLTEXT(全文索引)   
mysql>ALTER TABLE `table\_name` ADD FULLTEXT ( `column`)   
5.添加多列索引   
mysql>ALTER TABLE `table\_name` ADD INDEX index\_name ( `column1`, `column2`, `column3` )

Nginx:

1. Nginx默认并发数是多少，如何设置增大并发数

worker\_processes 8;

Linux:

1. 手写：显示php-fpm的进程的个数（非列表只要个数）

*netstat -napo |grep "[php-fpm](https://www.baidu.com/s?wd=php-fpm&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Yzn1c4uhD4nWPBP1DYuhnY0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPW64nWDsrHbvPHm4nHbLn1cY" \t "_blank)" | wc -l*

Larave：

1. Laravel用的版本
2. Laravel 路由
3. Laravel 容器
4. Laravel 中间件

其他：

1. 知道什么设计模式，用过什么设计模式，在哪用的
2. 自己的技术优势在哪里
3. 对之前问到的还有什么补充