

数组

索引数组（数组下标唯一识别 value），关联数组（键值对）

弱数据类型

初始化：

```
$arr = array( 'one' => 1,
             'two' => 'a string',
             'three' => array(...),
             ...);
```

插入元素：

```
$arr[] = $value    or    $arr['key'] = $value
```

常用函数：

is_array(\$arr)、count(\$arr)、sort(\$array, \$sort_flag)、
array_merge(array1, array2, ..., arrayN)、
array_push(\$arr, \$value1, \$value2, ...)等

遍历数组

```
for($count = 0; $count < count($arr); $count++) {
    //code:
}

foreach($arr as $key => $value) { // $key 可不需要, $value 可用
    //&$value 取引用
    //code:
}
```

```
}
```

面向对象

- 1、 属性，成员，实例化
- 2、 构造、析构函数
- 3、 继承，封装，访问控制
- 4、 \$this 的使用

```
3 class Animal{
4     protected $age = 0;
5     protected $name = '';
6
7     public function __construct($age, $name){
8
9         $this->age = $age;
10        $this->name = $name;
11    }
12
13    public function setAge($age){
14        $this->age = $age;
15        $this->showPersonInfo();
16    }
17
18    public function setName($name){
19        $this->name = $name;
20    }
21
22    public function run(){
23        echo "I'm running\n";
24    }
25 }
```

```
5 class Person extends Animal{
6     public static $contry = 'China';
7     protected $age = 0;
8     protected $name = '';
9
10    //php不支持重载函数
11    // public function __construct(){
12    //
13    // }
14
15    public function __construct($age, $name){
16        parent::__construct($age, $name);
17        $this->showPersonInfo();
18    }
19
20    public function setAge($age){
21        $this->age = $age;
22        $this->showPersonInfo();
23    }
24
25    public function setName($name){
26        $this->name = $name;
27        $this->showPersonInfo();
28    }
29
30    private function showPersonInfo(){
31        echo 'Person Name : '. $this->name. "\n";
32        echo 'Person Age : '. $this->age. "\n";
33    }
34
35    public function __destruct(){
36        echo 'bye';
37    }
38 }
```

```
$new_person = new Person(21, 'gyl');
$new_person->setAge(22);
$new_person->setName('chris_gyl');
// $new_person->showPersonInfo(); //Error
$new_person->run(); //调用父类方法
echo Person::$contry; //静态成员变量,被每个实例对象共有
```

异常处理

```
try{  
    //可能发生异常的代码段  
    if($condition){  
        throw new Exception('ERROR MESSAGE .....');  
    }  
} catch(Exception $e) {  
    //处理异常的代码段  
    echo $e->getMessage();  
}
```

可能出现的异常：

- 1、 资源句柄为空
- 2、 不合法的操作

composer

类似 node 的 npm、python 的 pip、ruby 的 bundle，composer 是 PHP 用来管理依赖（dependency）关系的工具。你可以在自己的项目中声明所依赖的外部工具库（libraries），Composer 会帮你安装这些依赖的库文件。

- a) 你有一个项目依赖于若干个库。
- b) 其中一些库依赖于其他库。
- c) 你声明你所依赖的东西。
- d) Composer 会找出哪个版本的包需要安装，并安装它们（将它们下载到你的项目中）。

example：在 composer.json 中添加如下代码并在命令行下执行 `composer install`

```
{  
    "require": {  
        "monolog/monolog": "1.2.*"  
    }  
}
```

session 和 cookie

功能：

http 无状态协议，每次读取页面的时候都是打开新的会话，服务器也不会自动维护客户的上下文信息，session 就是一种保存上下文信息的机制，通过 session_id 区分不同用户，传达不同的信息；

区别：

cookie 存储在客户端，留在客户计算机中，当该计算机请求页面时同时会发送 cookie；

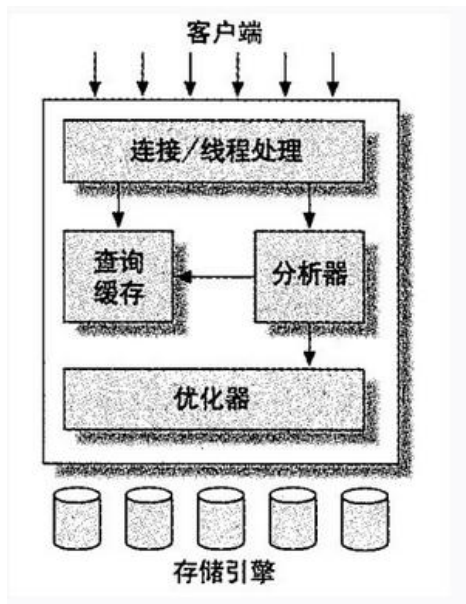
session 默认存在在服务器的一个文件里，也可以自定义存在数据库或内存中，对每个访客分配一个唯一标识即会话 ID；

关系：

Cookie 是通过客户端保持状态的解决方案，从 http 报头传递；session 依附于 cookie 的机制，在 cookie 中携带 session_id，当浏览器向服务器发出请求时将 cookie 带上，服务器从中解析出 session_id 从而获取该用户在服务器 session 中存储的信息。

mysql 的索引

mysql 引擎类型：InnoDB（支持事务）、MyISAM（支持全文索引）



最上层：连接处理、授权认证等

第二层：mysql 核心服务功能，包括查询解析、分析优化、缓存以及内置函数的实现

第三层：存储引擎，负责 mysql 中数据的存储和提取

索引类型：B-TREE、Hash

<http://blog.codinglabs.org/articles/theory-of-mysql-index.html>

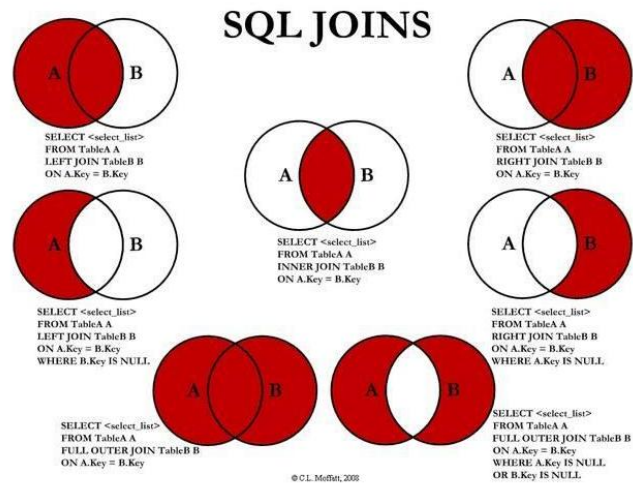
索引使用建议：

- 1、 尽量在 where 的字段上使用索引
- 2、 添加索引的字段散列程度尽可能大
- 3、 不要在列上进行运算
- 4、 不要设置过多索引

mysql 常用操作：

- 1、 连表查询
 - 1) JOIN
 - 2) LEFT JOIN

3) RIGHT JOIN



2、 内置函数

1) COUNT()

2) SUM()

3) MAX()

内置聚合函数常与 GROUP BY 联合使用