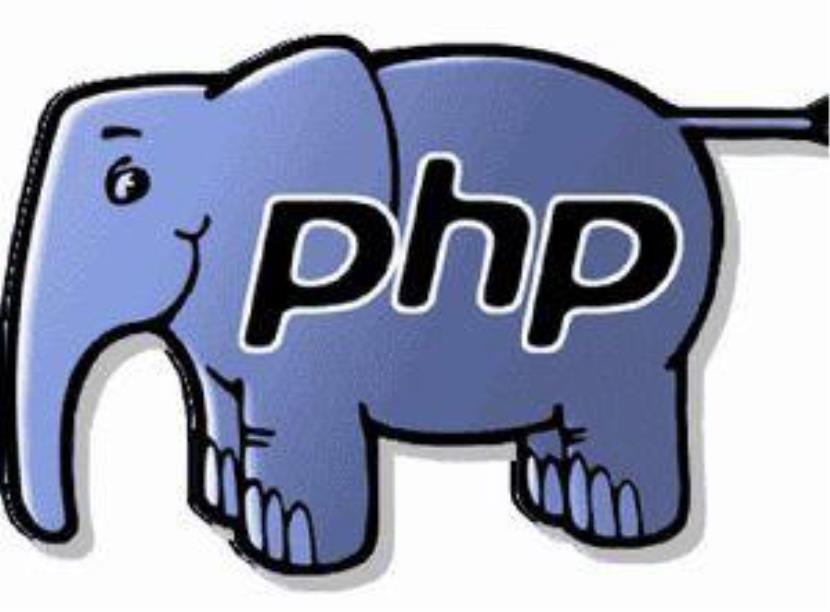
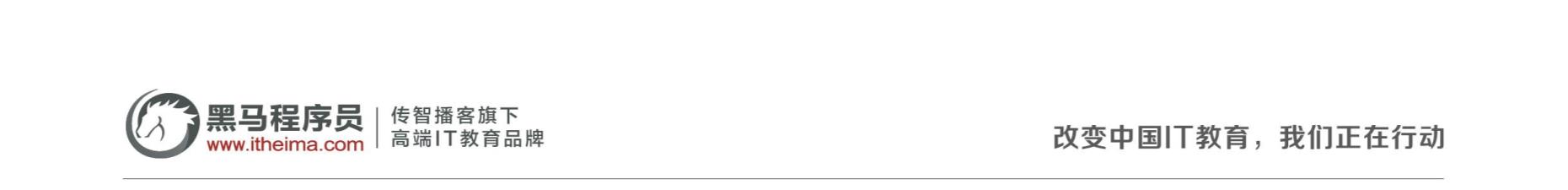
PHP 常用函数总结

上海校区就业部 PHP 组——薛杰

2017 年 5 月

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



一、数学函数

1.abs(): 求绝对值

$abs = abs(-4.2);

输入：任意数字

输出结果：4.2（数字绝对值数字）

2.ceil(): 进一法取整

echo ceil(9.999);

输入：任意数字

输出结果：10（浮点数进一取整）

3.floor(): 舍去法取整

echo floor(9.999);

输入：任意数字

输出结果：9（浮点数直接舍去小数部分）

4.fmod(): 浮点数取余

$x = 5.7;

$y = 1.3;

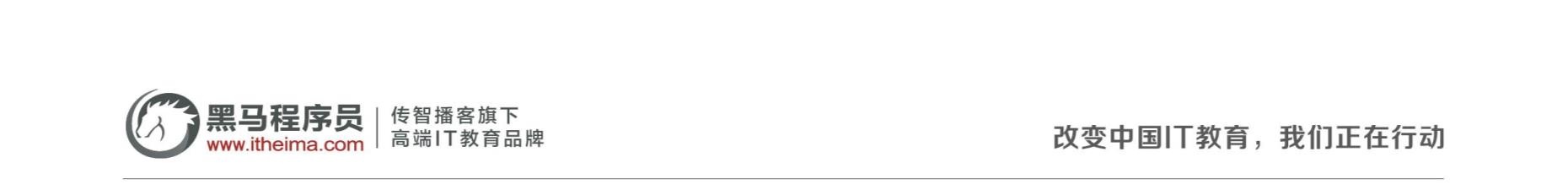
// 两个浮点数,x>y 浮点余数

$r = fmod($x, $y);

输出结果：0.5（5.7/1.3=4···0.5）

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



5.pow(): 返回数的 n 次方

echo pow(-1, 20);

输出结果：1（基础数|n 次方乘方值）

6.round(): 浮点数四舍五入

echo round(1.95583, 2);

输出结果：1.96（一个数值|保留小数点后多少位,默认为 0 舍入后的结果）

7.sqrt(): 求平方根

echo sqrt(9);

输出结果：3（被开方的数平方根）

8.max(): 求最大值

echo max(1, 3, 5, 6, 7);

输入: 多个数字或数组

输出结果：7（返回其中的最大值）

9.min(): 求最小值

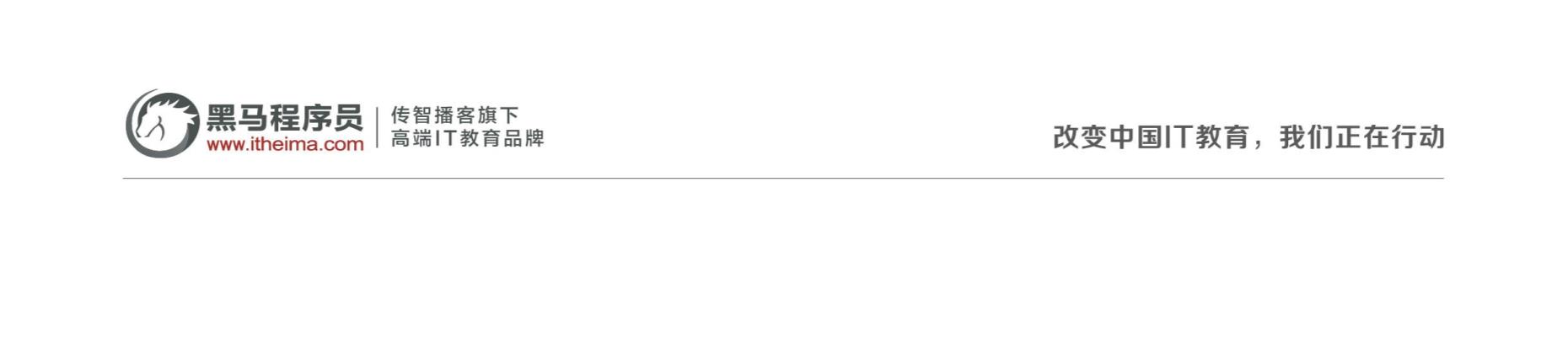
echo min(1, 3, 5, 6, 7);

输入: 多个数字或数组

输出结果：1（返回其中的最小值）

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



10.mt\_rand(): 更好的随机数

echo mt\_rand(0,9);

输入: 最小|最大,

输出: 随机数随机返回范围内的值

11.rand(): 随机数

echo rand(0,9);

输入: 最小|最大,

输出: 随机数随机返回范围内的值

12.pi(): 获取圆周率值

echo pi();

输出结果：3.1415926535898....

二、去空格或或其他字符:

13.trim(): 删除字符串两端的空格或其他预定义字符

$str = "\r\nHello World!\r\n";

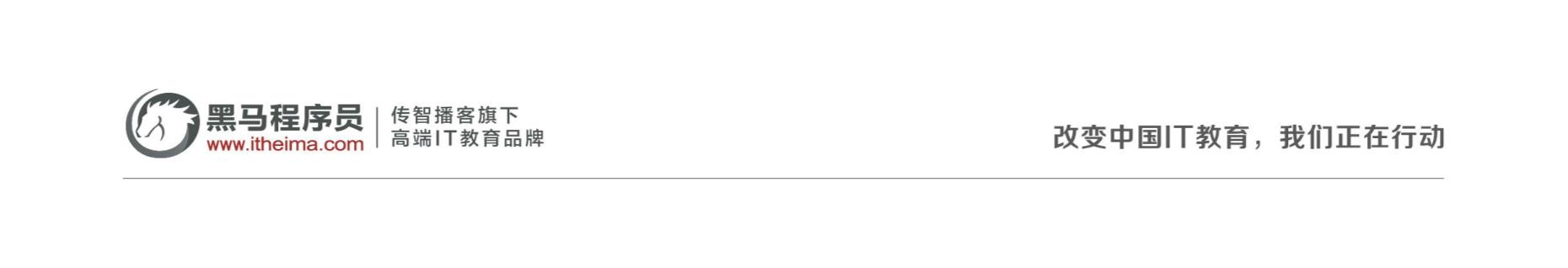
echo trim($str);

输入: 目标字符串

输出结果：Hello World!（清除后的字符串）

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



14.rtrim(): 删除字符串右边的空格或其他预定义字符

$str=”Hello World !\n\n”;

echo rtrim($str);

输入: 目标字符串

输出结果：Hello World!（清除后的字符串）

15.chop(): rtrim()的别名

16.ltrim(): 删除字符串左边的空格或其他预定义字符

$str = "\r\nHello World!";

echo ltrim($str);

输入: 目标字符串

输出结果：Hello World!（清除后的字符串）

17.dirname(): 返回路径中的目录部分

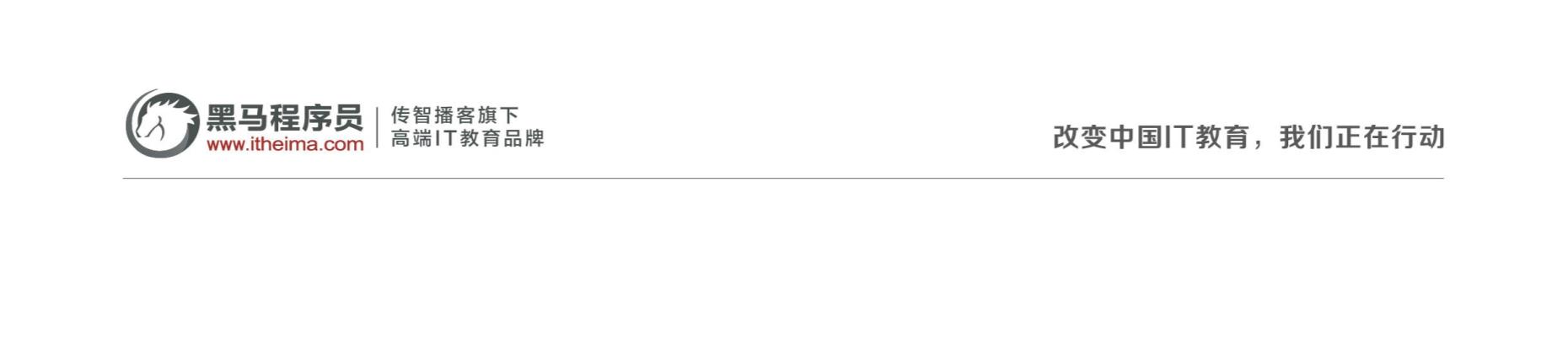
echo dirname("c:/testweb/home.php");

输入: 一个包含路径的字符串

输出结果：C:/testweb（返回文件路径的目录部分）

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



三、字符串生成与转化

18.str\_pad(): 把字符串填充为指定的长度

$str = "Hello World";

echo str\_pad($str,20,".");

输入: 要填充的字符串|新字符串的长度|供填充使用的字符串, 默认是空白

输出结果: Hello World.........（完成后的字符串）

19.str\_repeat(): 重复使用指定字符串

echo str\_repeat(".",13);

输入：要重复的字符串|字符串将被重复的次数

输出结果：............. （13 个点）

20.str\_split(): 把字符串分割到数组中

print\_r(str\_split("Hello"));

输入: 要分割的字符串|每个数组元素的长度，默认 1

输出结果: Array ( [0] => H [1] => e [2] => l [3] => l [4] => o )（拆分后的字符串数组）

21.strrev(): 反转字符串

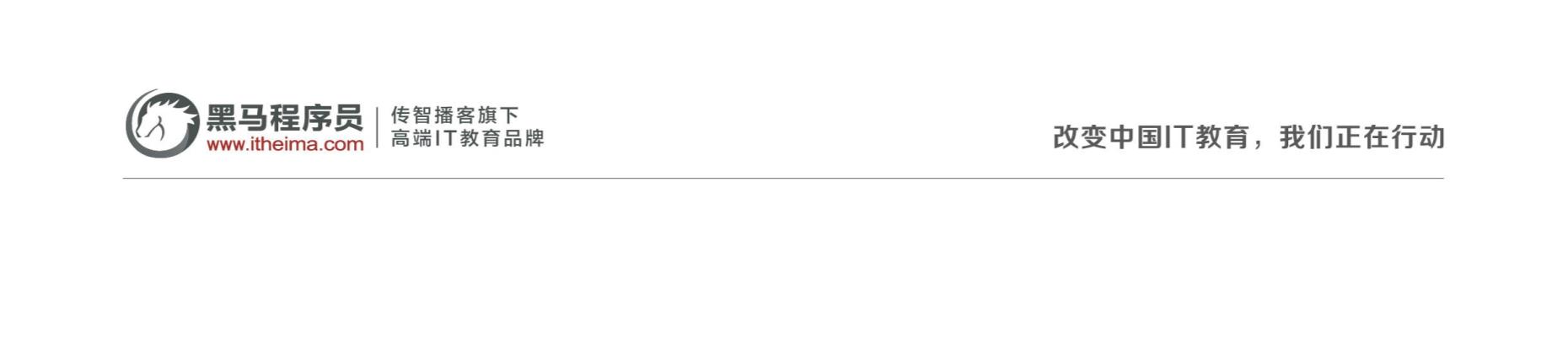
echo strrev("Hello World!");

输入：需要翻转的字符串

输出结果: !dlroW olleH（目标字符串颠倒顺序后的字符串）

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



22.wordwrap(): 按照指定长度对字符串进行折行处理

$str = "An example on a long word is:Supercalifragulistic";

echo wordwrap($str,15);

输入: 目标字符串|最大宽数

输出结果: An example on a long word is:Supercalifragulistic（折行后的新字符串）

23.str\_shuffle(): 随机地打乱字符串中所有字符

echo str\_shuffle("Hello World");

输入: 目标字符串顺序

输出: 打乱后的字符串

24.parse\_str(): 将字符串解析成变量

parse\_str("id=23&name=John%20Adams", $myArray);

print\_r($myArray);

输入: 要解析的字符串|存储变量的数组名称

输出: Array( [id] => 23 [name] => John Adams)

25.number\_format(): 通过千位分组来格式化数字

echo number\_format(20000000000);

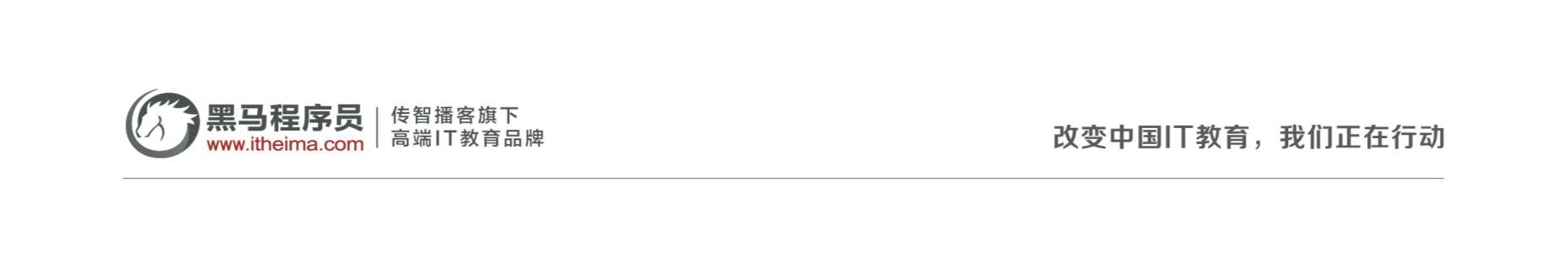
输入: 要格式化的数字|规定多少个小数|规定用作小数点的字符 串|规定用作千位分

隔符的字符串

输出: 20,000,000,000

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



四、大小写转换

26.strtolower(): 字符串转为小写

echo strtolower("Hello WORLD!");

输入：目标字符串

输出结果：hello world!（全部小写字符串）

27.strtoupper(): 字符串转为大写

echo strtoupper("Hello WORLD!");

输入：目标字符串

输出结果: HELLO WORLD!（全部大写字符串）

28.ucfirst(): 字符串首字母大写

echo ucfirst("hello world");

输入：目标字符串

输出结果：Hello world（字符串首个字母大写）

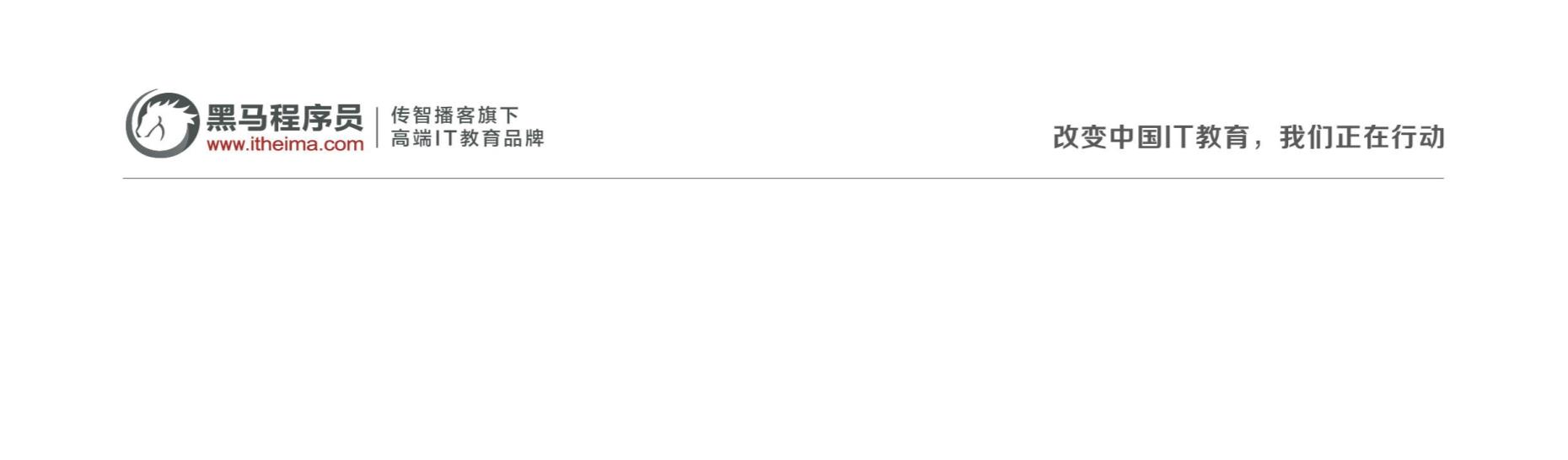
29.ucwords(): 字符串每个单词首字符转为大写

echo ucwords("hello world");

输出结果：Hello World（每个单词的首写字母都大写）

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



五、html 标签关联

30.htmlentities(): 把字符转为 HTML 实体

$str = "John & 'Adams'";

echo htmlentities($str, ENT\_COMPAT);

输出结果：John & 'Adams'

31.htmlspecialchars(): 预定义字符转 html 编码

$str = "This is some <b>bold</b> text.";

echo htmlspecialchars($str);

输出结果：This is some <b>bold</b> text.

32.nl2br(): \n 转义为 标签

echo nl2br("One line.\nAnother line.");

输出结果: One line.

Another line.（处理后的字符串）

33.strip\_tags(): 剥去 HTML、XML 以及 PHP 的标签

echo strip\_tags("Hello world! ");

输出结果：Hello world!

34.addcslashes():在指定的字符前添加反斜线转义字符串中字符

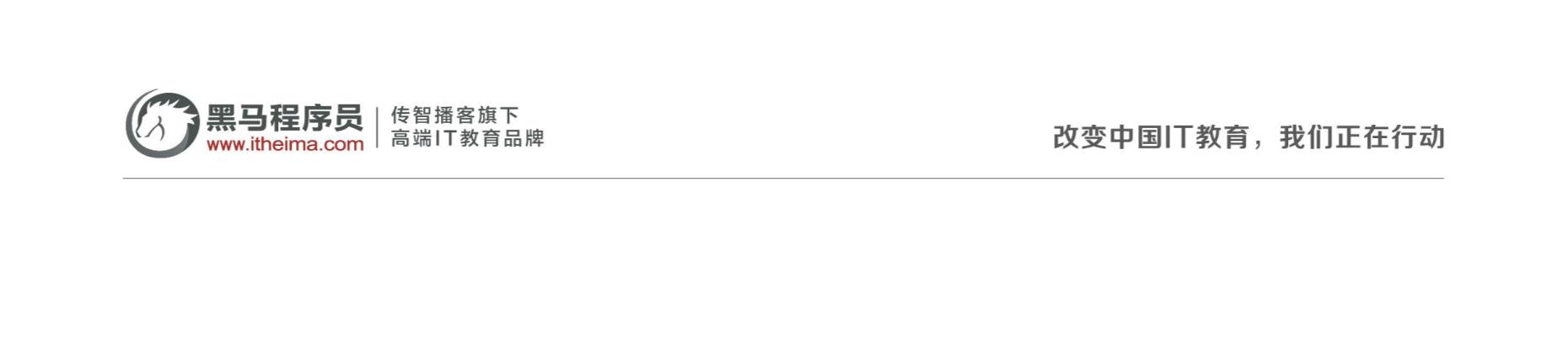
$str = "Hello, my name is John Adams.";

echo addcslashes($str,'m');

输入: 目标字符串|指定的特定字符或字符范围

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



输出结果：Hello,\my na\me is John Ad\ms.

35.stripcslashes(): 删除由 addcslashes()添加的反斜线

echo stripcslashes("Hello, \my na\me is Kai Ji\m.");

输入：目标字符串

输出结果：Hello, my name is Kai Jim.

36.addslashes(): 指定预定义字符前添加反斜线

$str = "Who's John Adam’s?";

echo addslashes($str);

输出结果: Who\'s John Adam\'s? （把目标串中的' " \和 null 进行转义处理）

37.stripslashes(): 删除由 addslashes()添加的转义字符

echo stripslashes("Who\'s John Adams?");

输出结果：Who's John Adams?（清除转义符号）

38.quotemeta(): 在字符串中某些预定义的字符前添加反斜线

$str = "Hello world. (can you hear me?)";

echo quotemeta($str);

输出结果：Hello world\. \(can you hear me\?\)

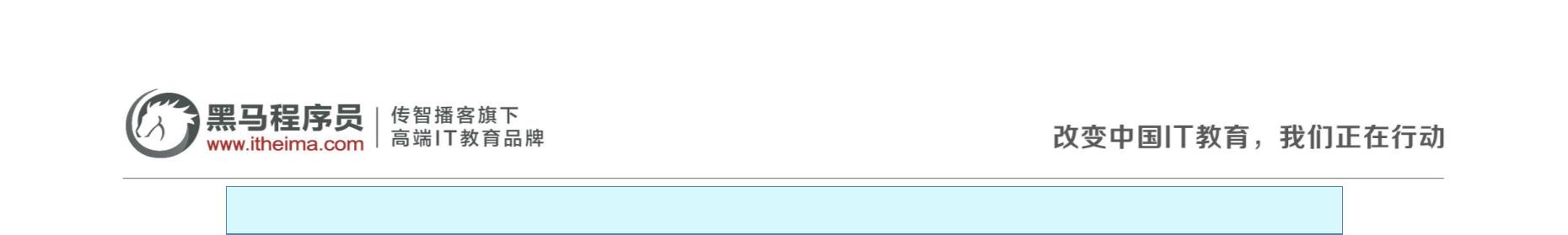
39.chr(): 从指定的 ASCII 值返回字符

echo chr(052);

输出结果：\*（ASCII 值返回对应的字符）

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



40.ord(): 返回字符串第一个字符的 ASCII 值

echo ord("hello");

输出结果：104（字符串第一个字符的 ASCII 值）

六、字符串比较

41.strcasecmp(): 不区分大小写比较两字符串

echo strcasecmp("Hello world!","HELLO WORLD!");

输入: 两个目标字符串

输出结果：0（如果两个字符串相等，0|如果 string1 小于 string2，小于 0|如果 string1 大

于 string2，大于 0）

42.strcmp(): 区分大小写比较两字符串

echo strcmp("Hello world!","Hello world!");

输入：两个目标字符串

输出结果：0（如果两个字符串相等，0|如果 string1 小于 string2，小于 0|如果 string1 大

于 string2，大于 0）

43.strncmp(): 比较字符串前 n 个字符,区分大小写

echo strncmp("Hello world!","Hello earth!",6);

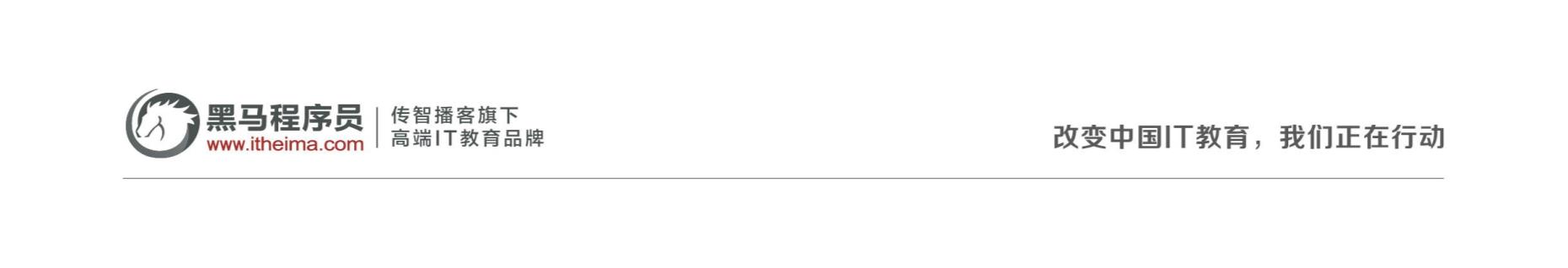
输入：两个比较的字符串和指定的比较长度

输出结果：0（如果两个字符串相等，0|如果 string1 小于 string2，小于 0|如果 string1

大于 string2，大于 0）

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



44.strncasecmp(): 比较字符串前 n 个字符,不区分大小写

echo strncasecmp("I love China!","i LOVE Shanghai!",6);

输入：两个比较的字符串和指定的比较长度

输出结果：0（如果两个字符串相等，0|如果 string1 小于 string2，小于 0|如果 string1

大于 string2，大于 0）

45.strnatcmp(): 自然顺序法比较字符串长度,区分大小写

echo strnatcmp("2Hello world!","10Hello world!");

输入: 目标字符串

输出结果：-1（如果两个字符串相等，0|如果 string1 小于 string2，小于 0|如果 string1

大于 string2，大于 0）

46.strnatcasecmp(): 自然顺序法比较字符串长度, 不区分大小写

echo strnatcasecmp("10Hello world!","2Hello WORLD!");

输入目标：输入: 目标字符串

输出结果：1（如果两个字符串相等，0|如果 string1 小于 string2，小于 0|如果 string1

大于 string2，大于 0）

七、字符串切割与拼接

47.chunk\_split()：将字符串分成小块

$str = "Shanghai";

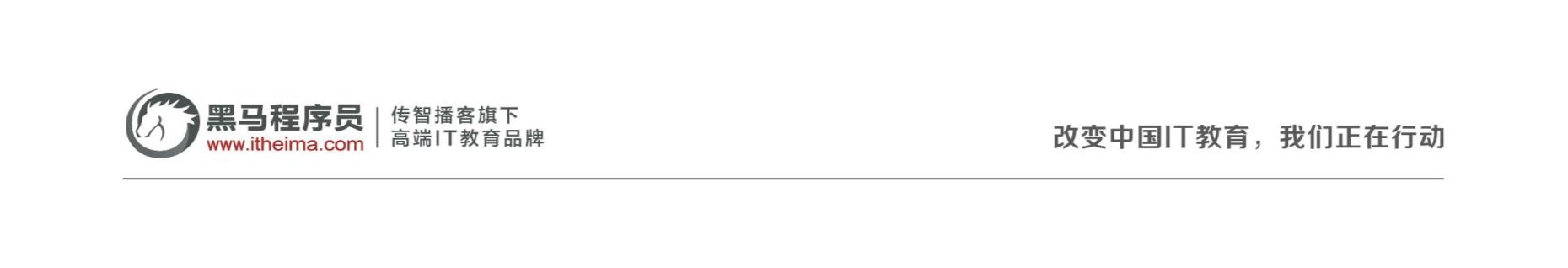
echo chunk\_split($str,1,".");

输入: 目标字串，长度，插入结束符

输出: S.h.a.n.g.h.a.i.（分割后的字符串）

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



48.strtok(): 切开字符串

$string = "Hello world. Beautiful day today.";

$token = strtok($string, " ");

输入：目标字符串

输出结果：Hello

world.

Beautiful

day

today（以$token 为标志切割返回切割后的字符串）

49.explode(): 使用一个字符串为标志分割另一个字符串

$str = "Hello world. I love Shanghai!";

print\_r (explode(" ",$str));

输入: 分割符，目标字符串，返回数组最多包含元素数

输出: Array ( [0] => Hello [1] => world. [2] => I [3] => love [4] => Shanghai! )（字符串被分割

后形成的数组）

50.implode(): 同 join,将数组值用预订字符连接成字符串

$arr = array('Hello','World!','I','love','Shanghai!');

echo implode(" ",$arr);

输入：分隔符，用””则直接相连

输出结果：Hello World! I love Shanghai!

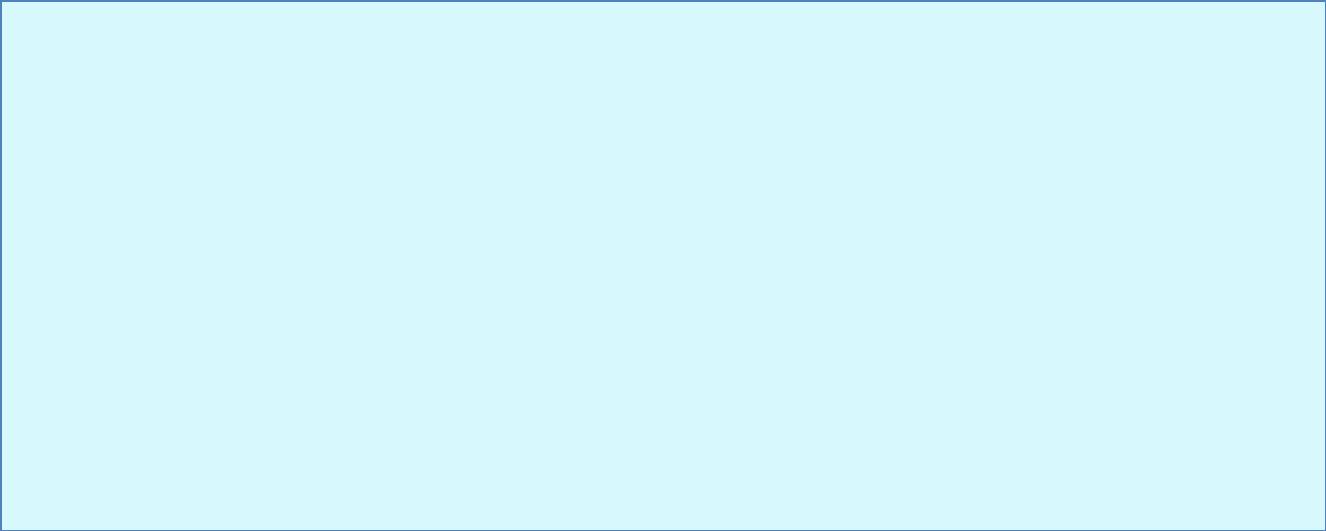
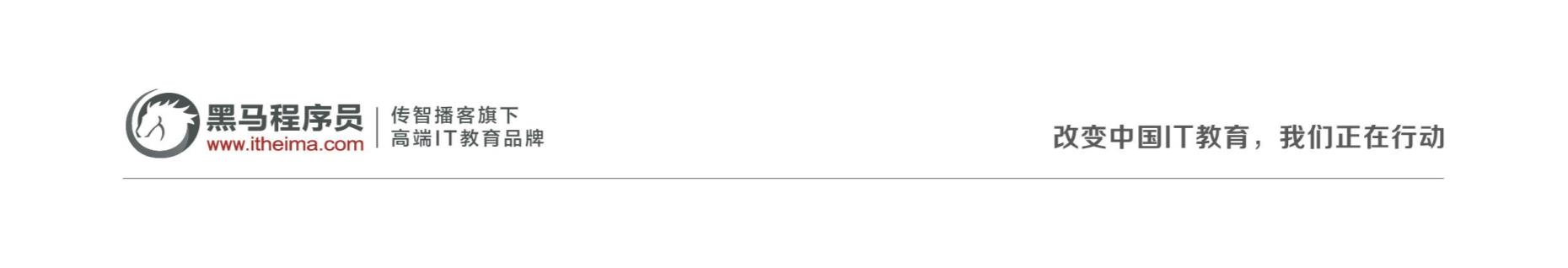
51.substr(): 截取字符串

echo substr("Hello world",6);

输入：目标字符串，开始下标位置（下标从 0 开始）

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



输出结果：world

八、字符串查找替换

52.str\_replace(): 字符串替换操作,区分大小写

echo str\_replace("world","Shanghai","Hello world!");

输入: 查找的字符串，替换的字符串，被查找字串

输出: Hello Shanghai（返回替换后的结果）

53.str\_ireplace() 字符串替换操作,不区分大小写

echo str\_ireplace("WORLD","Shanghai","Hello world!");

输入: 查找的字符串，替换的字符串，被查找字串

输出: Hello Shanghai!（返回替换后的结果）

54.substr\_count(): 统计一个字符串,在另一个字符串中出现次数

echo substr\_count("I love Shanghai. Shanghai is the biggest city in china.","Shanghai");

输入：被统计字符串，检查的目标字符串

输出结果：2

55.substr\_replace(): 替换字符串中某串为另一个字符串

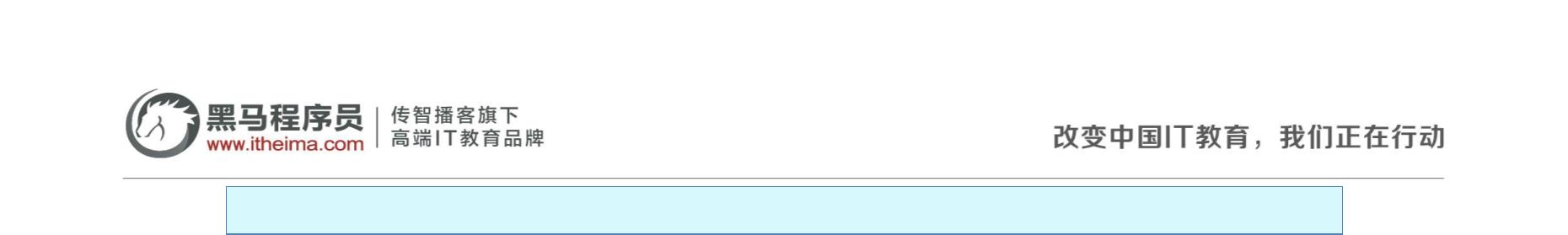
echo substr\_replace("Hello","world",0);

输入：查找字符串，替换上的字符串，起始位置（下标为 0 开始）

输出结果：world

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



56.similar\_text(): 返回两字符串相同字符的数量

echo similar\_text("Hello World","Hello Shanghai");

输入: 两个比较的字符串

输出: 6（整形,相同字符数量）

57.strrchr(): 返回一个字符串在另一个字符串中最后一次出现位置开始到末尾的字符串

echo strrchr("I love Shanghai!","Shanghai");

输出结果：Shanghai!

58.strstr(): 返回一个字符串在另一个字符串中开始位置到结束的字符串

echo strstr("I love Shanghai!","Shanghai");

输出结果：Shanghai!

59.strchr(): strstr()的别名,返回一个字符串在另一个字符串中首次出现的位置开始到末尾的字符

串

echo strchr("Hello world!","world");

输出结果：world!

60.stristr(): 返回一个字符串在另一个字符串中开始位置到结束的字符串，不区分大小写

echo stristr("Hello world!","WORLD");

输出结果：world!

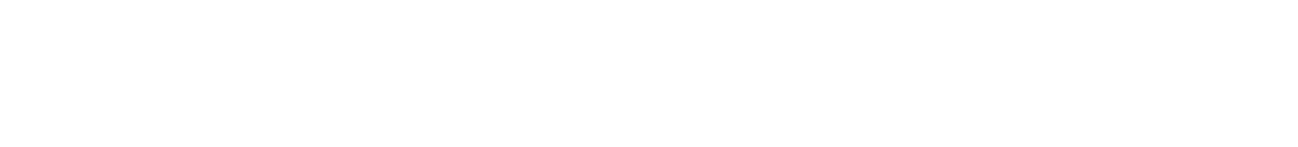
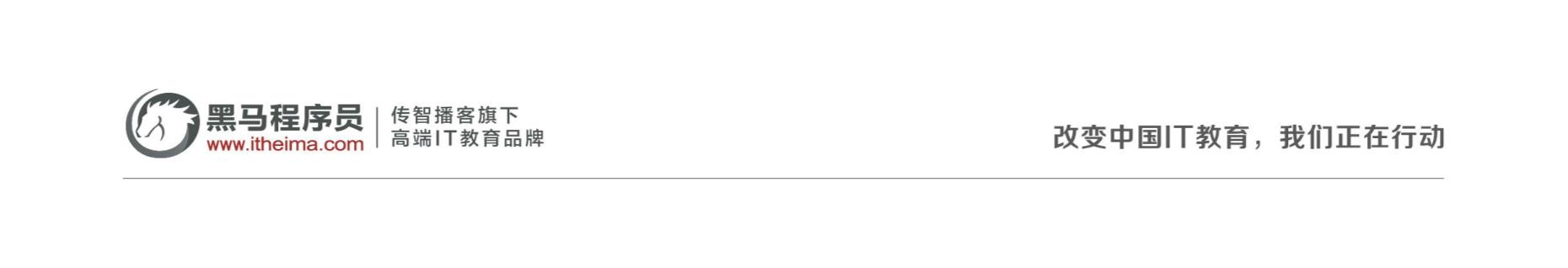
61.strtr(): 转换字符串中的某些字符

echo strtr("Hilla Warld","ia","eo");

输入：目标字符串，要替换掉的字符，替换上的字符

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



输出结果：Hello World

62.strpos(): 寻找字符串中某字符最先出现的位置

echo strpos("You love php, I love php too!","php");

输出结果：9（下标从 0 开始）

63.stripos(): 寻找字符串中某字符最先出现的位置,不区分大小写

echo stripos("You love php, I love php too!","PHP");

输出结果：9（下标从 0 开始）

64.strrpos(): 寻找某字符串中某字符最后出现的位置

echo strrpos("You love php, I love php too!","php");

输出结果：21（下标从 0 开始）

65.strripos(): 寻找某字符串中某字符最后出现的位置,不区分大小写

echo strripos("You love php, I love php too!","PHP");

输入：目标字符串，查找字符串

输出结果：21（下标从 0 开始）

66.strspn(): 返回字符串中首次符合 mask 的子字符串长度

echo strspn("Hello world!","kHlleo");

输入：目标字符串，

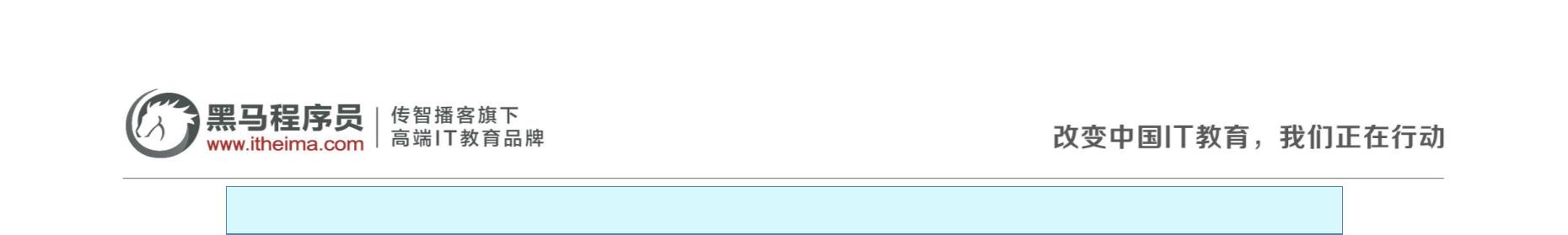
输出结果：5（"kHlleo"在"Hello world!"中，有 5 个字符串 Hello 符合）

67.strcspn(): 返回字符串中不符合 mask 的字符串的长度

echo strcspn("Hello world!","w");

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



输入: 被查询，查询字符串，开始查询的字符，是查询长度

输出: 返回从开始到第几个字符

九、字符串统计

68.str\_word\_count(): 统计字符串含有的单词数

echo str\_word\_count("I love Shanghai!");

输入: 目标字符串

输出: 3（统计处的数量）

69.strlen(): 统计字符串长度 int strlen(str $str)

echo strlen("Shanghai");

输入: 目标字符串

输出结果: 8（整型长度）

70.count\_chars(): 统计字符串中所有字母出现次数(0..255)

$str = "Hello World!";

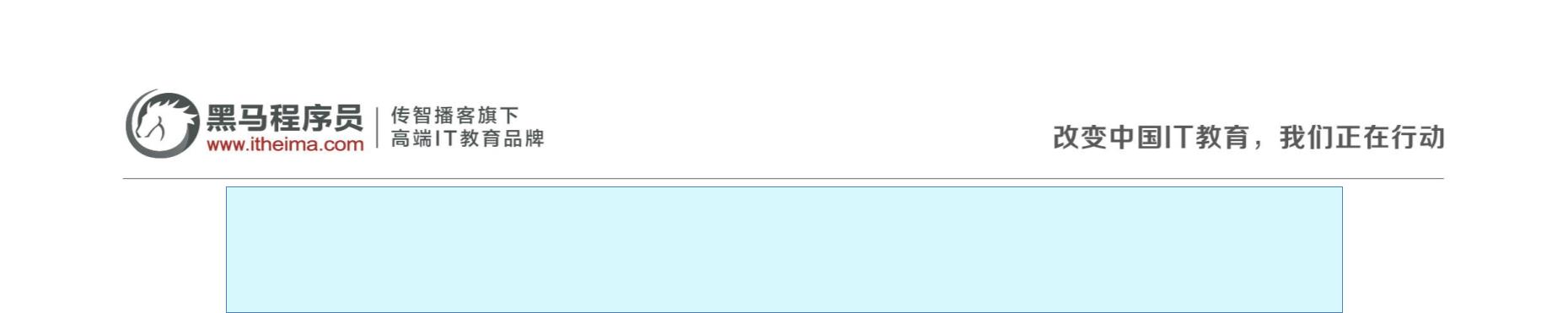
echo count\_chars($str,3);

输入：目标字符串

输出结果：!HWdelor（根据 ASCII 码先后显示输出）

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



十、字符串编码

71.md5(): 字符串 md5 编码

$str = "Hello";

echo md5($str);

输入：目标加密字符串

输出结果：8b1a9953c4611296a827abf8c47804d7

十一、数组创建

72.array(): 生成一个数组

$a=array("Dog","Cat","Horse");

print\_r($a);

输入：数组值或键=>值一个数组型变量

输出结果：Array ( [0] => Dog [1] => Cat [2] => Horse )

73.array\_combine(): 生成一个数组,用一个数组的值 作为键名,另一个数组值作为值

$a1=array("a","b","c","d");

$a2=array("Cat","Dog","Horse","Cow");

print\_r(array\_combine($a1,$a2));

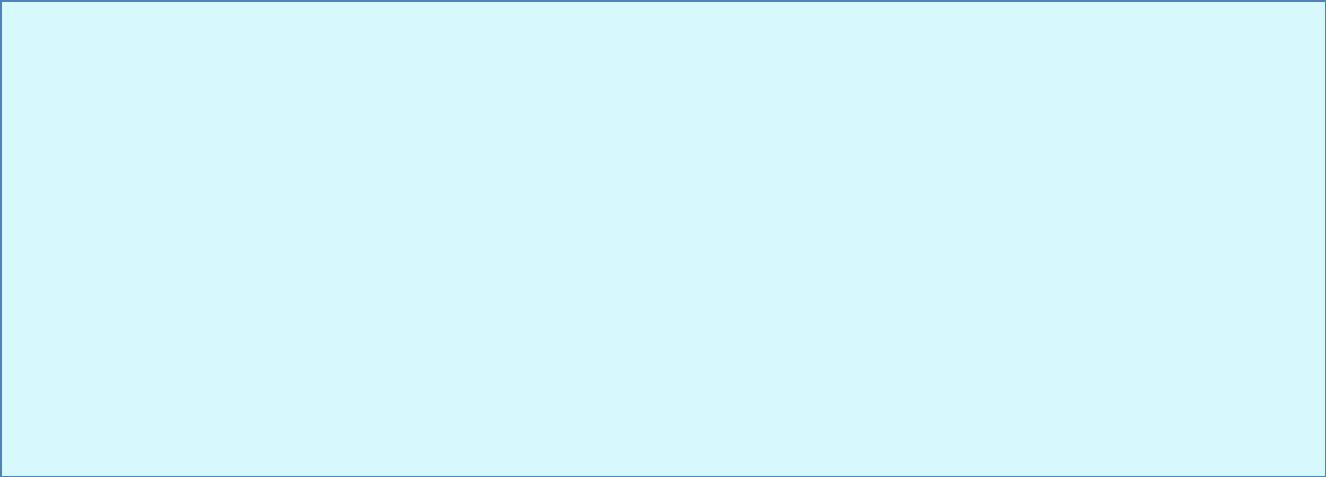
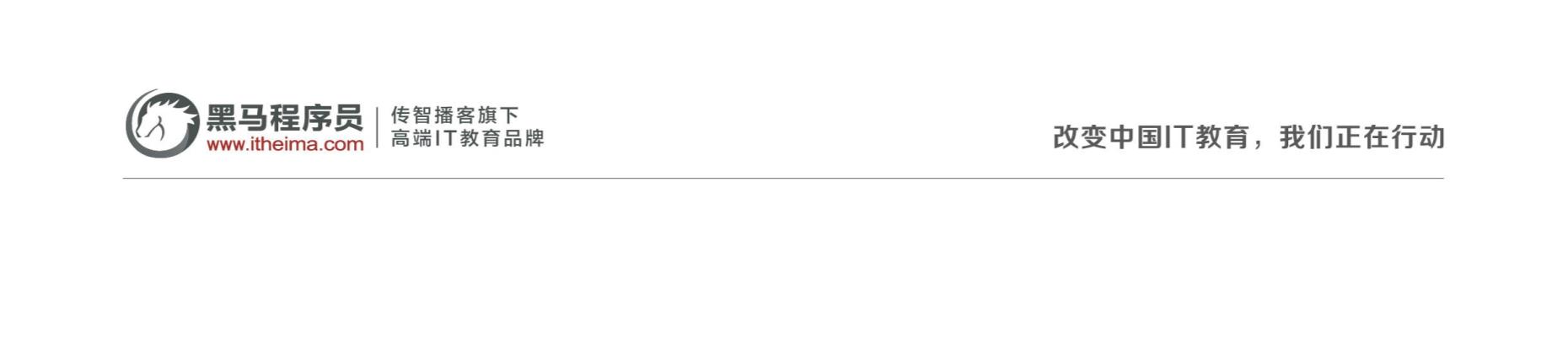
输入：$a1 为提供键，$a2 提供值

输出结果：Array ( [a] => Cat [b] => Dog [c] => Horse [d] => Cow ) （合

成后的数组）

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



74.range(): 创建并返回一个包含指定范围的元素的数组。

$number = range(0,50,10);

print\_r ($number);

输入: 最小值，最大值，步长

输出结果：Array ( [0] => 0 [1] => 10 [2] => 20 [3] => 30 [4] => 40 [5]

=> 50 )（合成后的数组）

75.compact(): 创建一个由参数所带变量组成的数组

$firstname = "Peter";

$lastname = "Griffin";

$age = "38";

$result = compact("firstname", "lastname", "age");

print\_r($result);

输入：变量或数组

输出结果：Array ( [firstname] => Peter [lastname] => Griffin [age] =>

38 )（返回由变量名为键,变量值为值的数组,变量也可以为多维数组.会递归处理）

76.array\_fill(): 用给定的填充(值生成)数组

$a=array\_fill(2,3,"Dog");

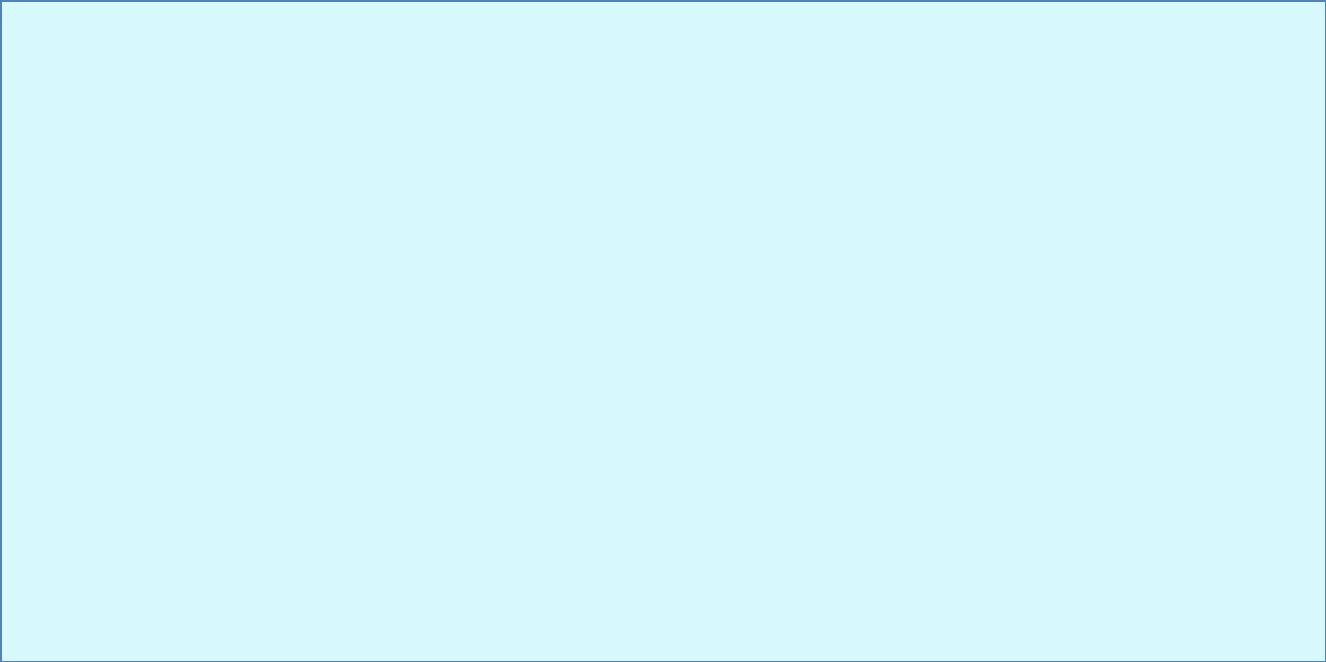
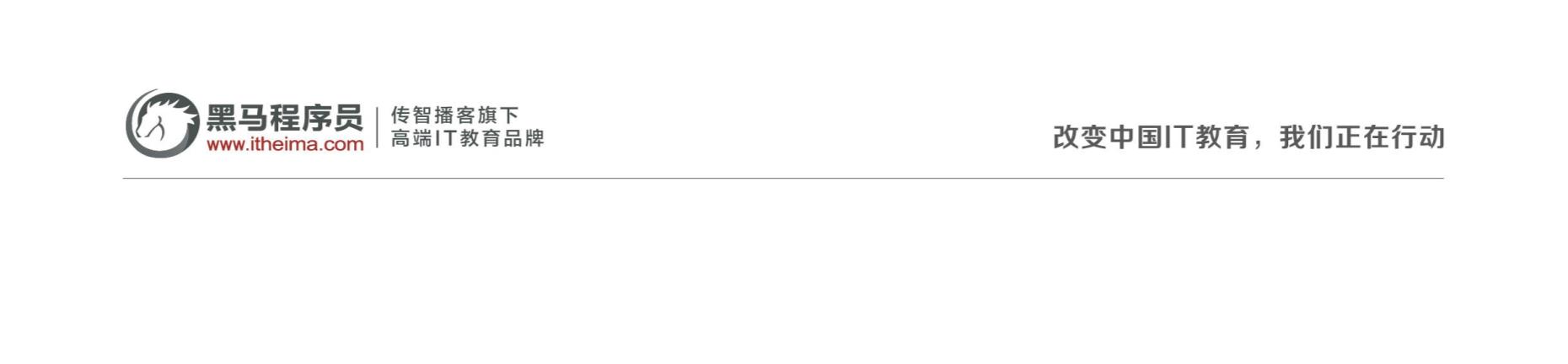
print\_r($a);

输入：键，填充的数量，填充内容

输出结果：Array ( [2] => Dog [3] => Dog [4] => Dog )（返回完成的数组）

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



十二、数组合并和拆分

77.array\_chunk(): 把一个数组分割为新的数组块

$a=array("a"=>"Cat","b"=>"Dog","c"=>"Horse","d"=>"Cow");

print\_r(array\_chunk($a,2));

输入：一个数组

输出结果：Array ( [0] => Array ( [0] => Cat [1] => Dog ) [1] => Array

( [0] => Horse [1] => Cow ) )（分割后的多维数组，规定每个新数组包含 2 个元素）

78.array\_merge(): 把两个或多个数组合并为一个数组。

$a1=array("a"=>"Horse","b"=>"Dog");

$a2=array("c"=>"Cow","b"=>"Cat");

print\_r(array\_merge($a1,$a2));

输入: 两个数组

输出: Array ( [a] => Horse [b] => Cat [c] => Cow ) （返回完成后的数组）

79.array\_slice(): 在数组中根据条件取出一段值，并返回。

$a=array(0=>"Dog",1=>"Cat",2=>"Horse",3=>"Bird");

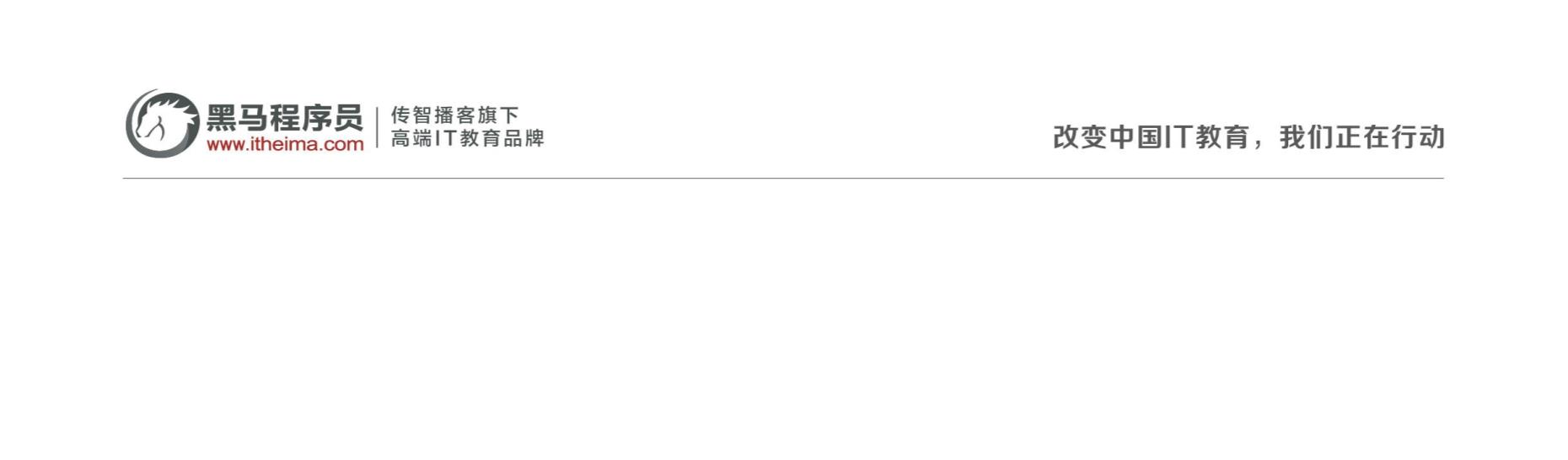
print\_r(array\_slice($a,1,2));

输入: 一个数组

输出: Array ( [0] => Cat [1] => Horse )（1 为从'Cat'开始，2 为返回两个元素）

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



十三、数组比较

80.array\_diff(): 返回两个数组的差集数组

$a1=array(0=>"Cat",1=>"Dog",2=>"Horse");

$a2=array(3=>"Horse",4=>"Dog",5=>"Fish");

print\_r(array\_diff($a1,$a2));

输入: 两个或多个数组

输出结果: Array ( [0] => Cat )（$a1 与$a2 的不同之处）

81.array\_intersect(): 返回两个或多个数组的交集数组

$a1=array("a"=>"red","b"=>"green","c"=>"blue","d"=>"yellow");

$a2=array("e"=>"red","f"=>"green","g"=>"blue");

$result=array\_intersect($a1,$a2);

print\_r($result);

输入：两个或多个数组

输出结果: Array ( [a] => red [b] => green [c] => blue )（$a1 与$a2 的相

同之处）

十四、数组查找替换

82.array\_search(): 在数组中查找一个值，返回一个键，没有返回返回假

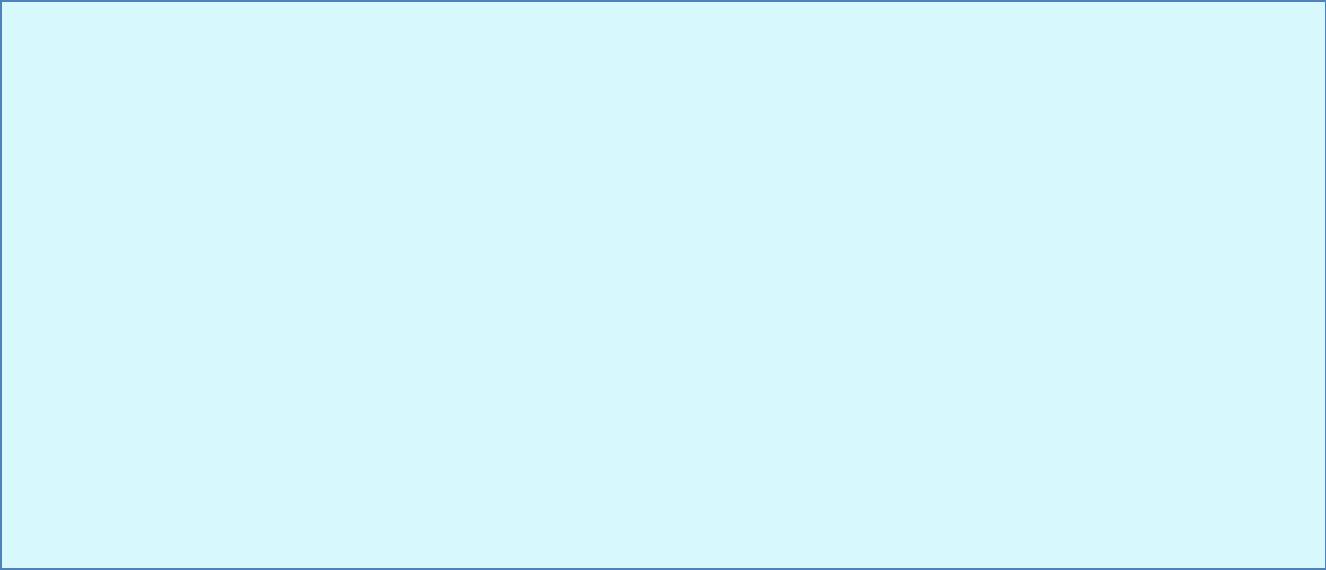
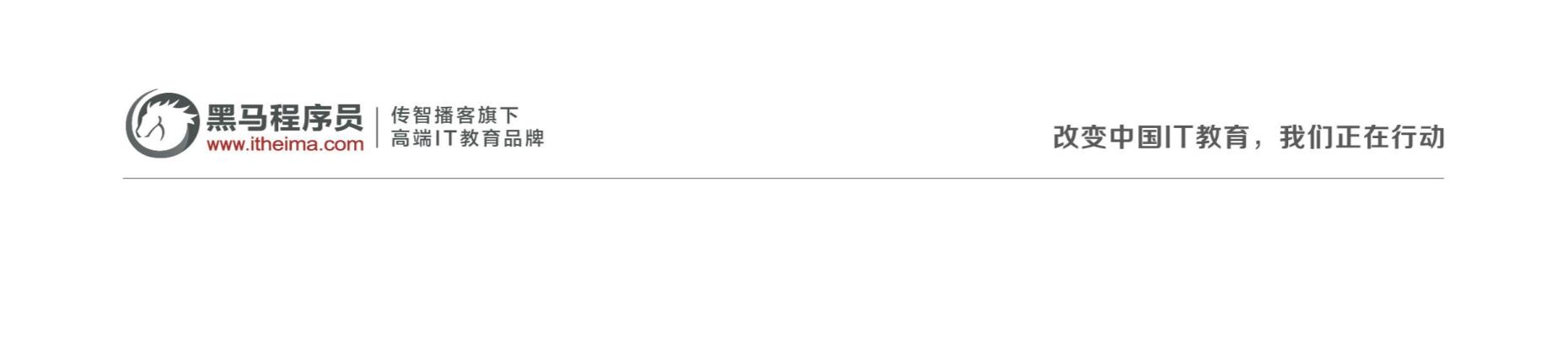
$a=array("a"=>"Dog","b"=>"Cat","c"=>"Horse");

echo array\_search("Dog",$a);

输入: 一个数组

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



输出: a（成功返回键名,失败返回 false）

83.array\_splice(): 把数组中一部分删除用其他值替代

$a1=array(0=>"Dog",1=>"Cat",2=>"Horse",3=>"Bird");

$a2=array(0=>"Tiger",1=>"Lion");

array\_splice($a1,0,2,$a2);

print\_r($a1);

输入: 一个或多个数组

输出:Array ( [0] => Tiger [1] => Lion [2] => Horse [3] => Bird )（$a1

被移除的部分由$a2 补全）

84.array\_sum(): 返回数组中所有值的总和

$a=array(0=>"5",1=>"15",2=>"25");

echo array\_sum($a);

输入: 一个数组

输出: 45（返回和）

85.in\_array(): 在数组中搜索给定的值,区分大小写

$people = array("Peter", "Joe", "Glenn", "Cleveland");

if (in\_array("Glenn",$people) ){

echo "Match found";

} else{

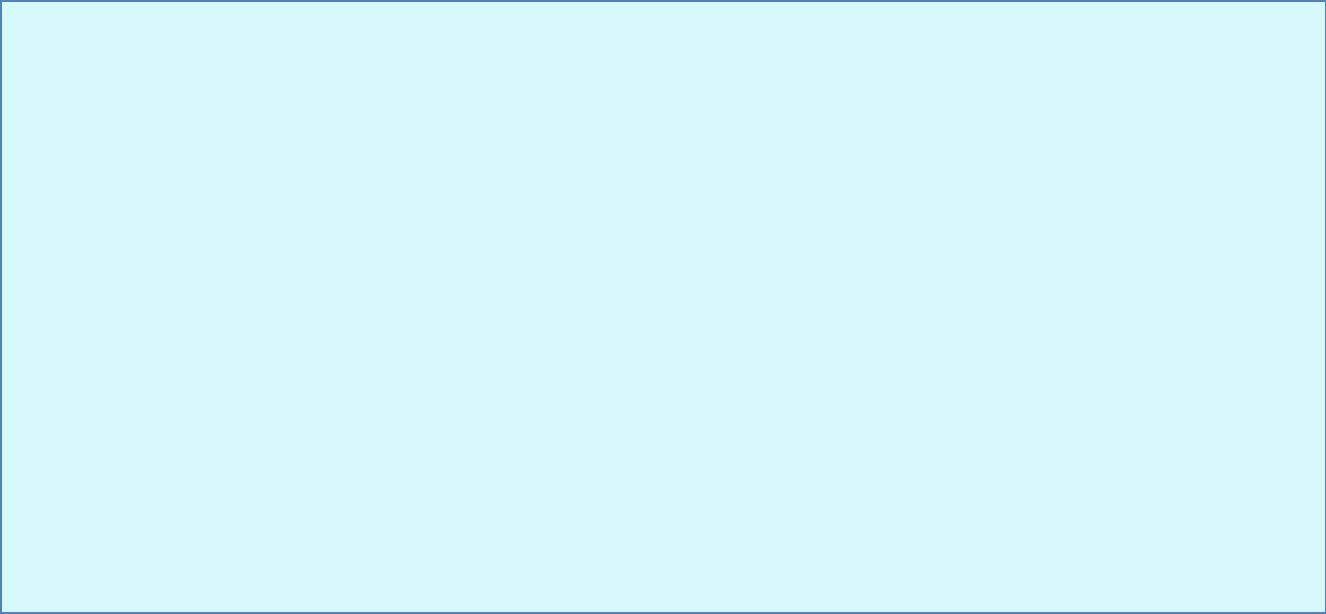
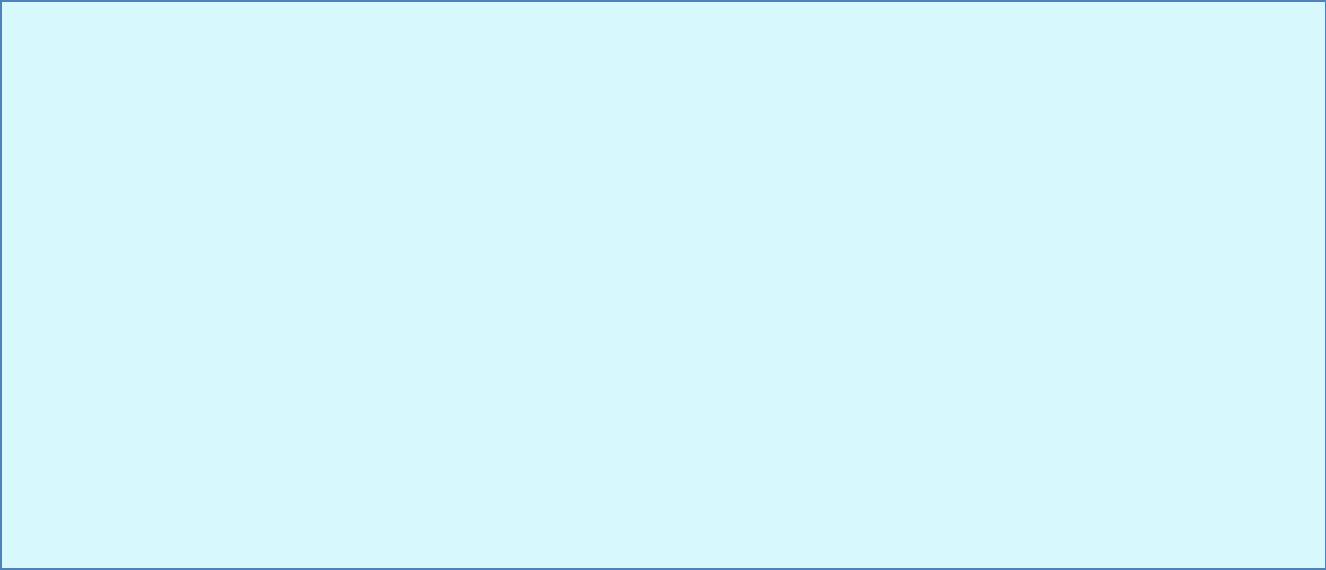
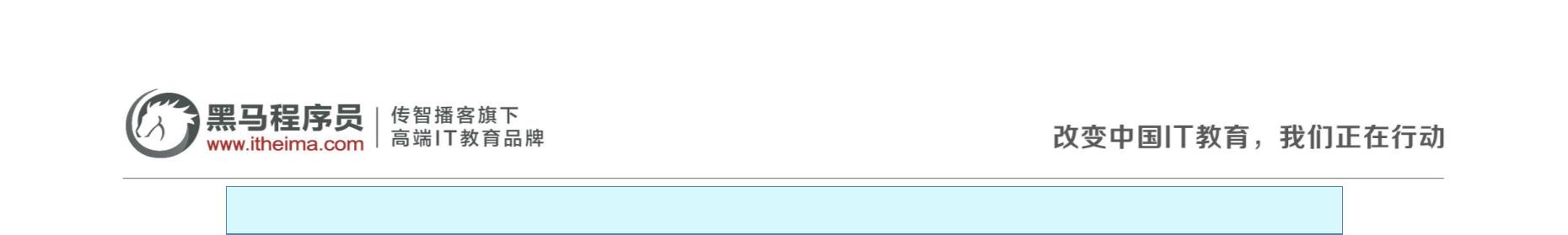
echo "Match not found";

}

输入: 需要搜索的值|数组

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



输出结果: Match found

86.array\_key\_exists(): 判断某个数组中是否存在指定的 key

$a=array("Volvo"=>"XC90","BMW"=>"X5");

if (array\_key\_exists("Volvo",$a))

{echo "键存在！";}

else {echo "键不存在！";}

输入: 需要搜索的键名|数组

输出结果：键存在！

十五、数组指针操作

87.key(): 返回数组内部指针当前指向元素的键名

$people = array("Bill", "Steve", "Mark", "David");

echo "键的当前位置是：" . key($people);

输入：目标数组或者数组变量

输出结果：键的当前位置是：0

88.current(): 返回数组中的当前元素(单元).

$people = array("Bill", "Steve", "Mark", "David");

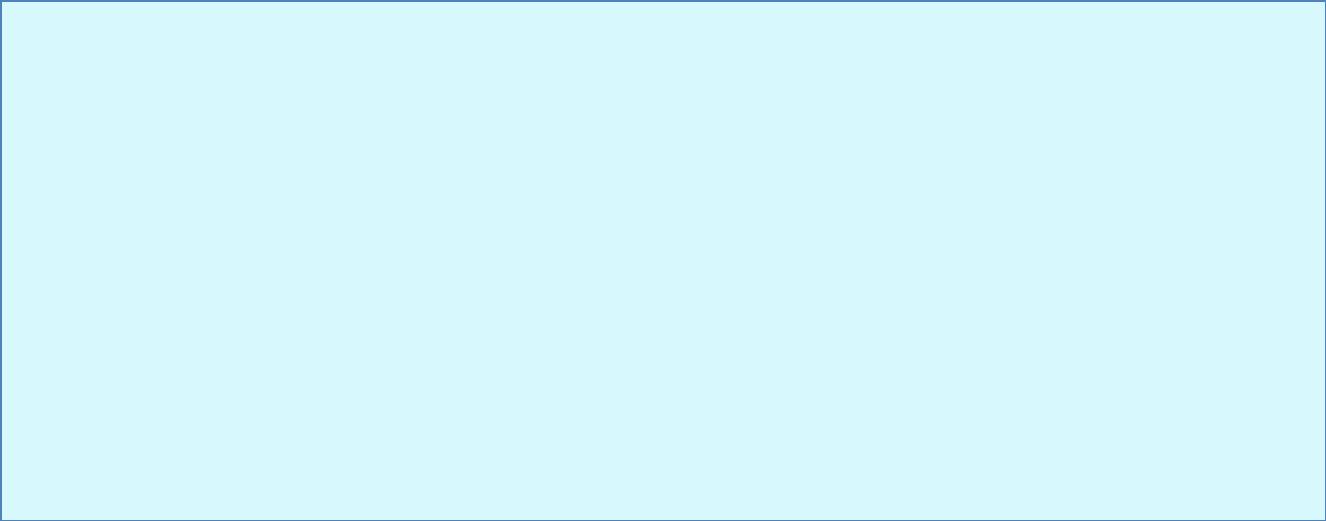
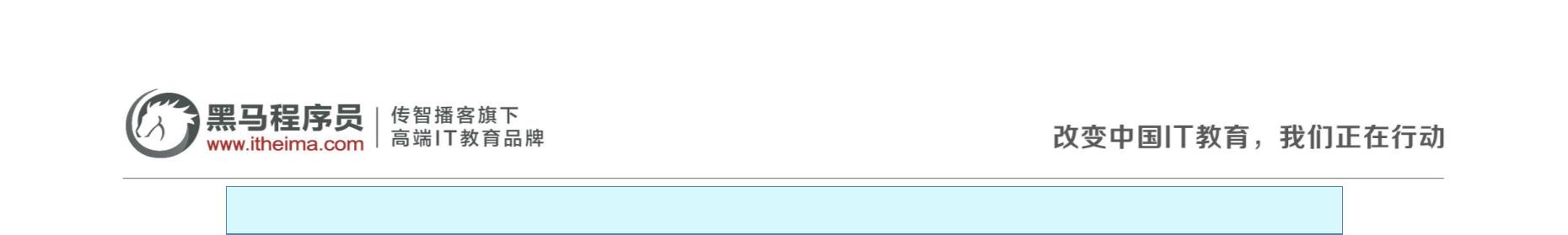
echo current($people) . "<br>";

输入：目标数组或者数组变量

输出结果：Bill

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



89.next(): 把指向当前元素的指针移动到下一个元素的位置,并返回当前元素的值

$people = array("Bill", "Steve", "Mark", "David");

echo current($people) . "<br>";

echo next($people);

输入：目标数组或者数组变量

输出结果：Bill

Steve（查询当前指针的下个指针指向的内容）

90.prev(): 把指向当前元素的指针移动到上一个元素的位置,并返回当前元素的值

$people = array("Bill", "Steve", "Mark", "David");

echo current($people) . "<br>";

echo next($people) . "<br>";

echo prev($people);

输入：目标数组或者数组变量

输出结果：Bill

Steve

Bill

91.end(): 将数组内部指针指向最后一个元素，并返回该元素的值(如果成功)

$people = array("Bill", "Steve", "Mark", "David");

echo current($people) . "<br>";

echo end($people);

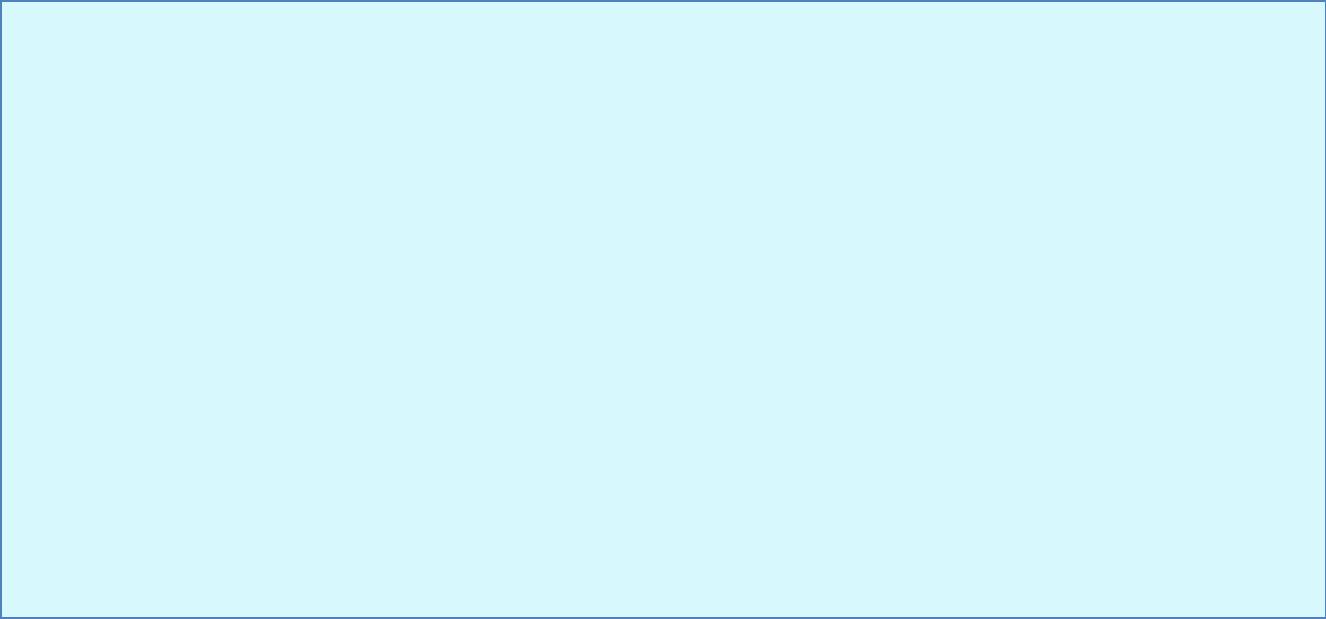
输入：目标数组或者数组变量

输出结果：Bill

David

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



92.reset(): 把数组的内部指针指向第一个元素，并返回这个元素的值

$people = array("Bill", "Steve", "Mark", "David");

echo current($people) . "<br>";

echo next($people) . "<br>";

echo reset($people);

输出结果：Bill

Steve

Bill

93.list(): 用数组中的元素为一组变量赋值

$my\_array = array("Dog","Cat","Horse");

list($a, $b, $c) = $my\_array;

echo "I have several animals, a $a, a $b and a $c.";

输入: 需要赋值的变量

输出结果: I have several animals, a Dog, a Cat and a Horse.（变量分别匹配数组中的值）

94.array\_shift(): 删除数组中的第一个元素，并返回被删除元素的值

$a=array("a"=>"Dog","b"=>"Cat","c"=>"Horse");

echo array\_shift($a);

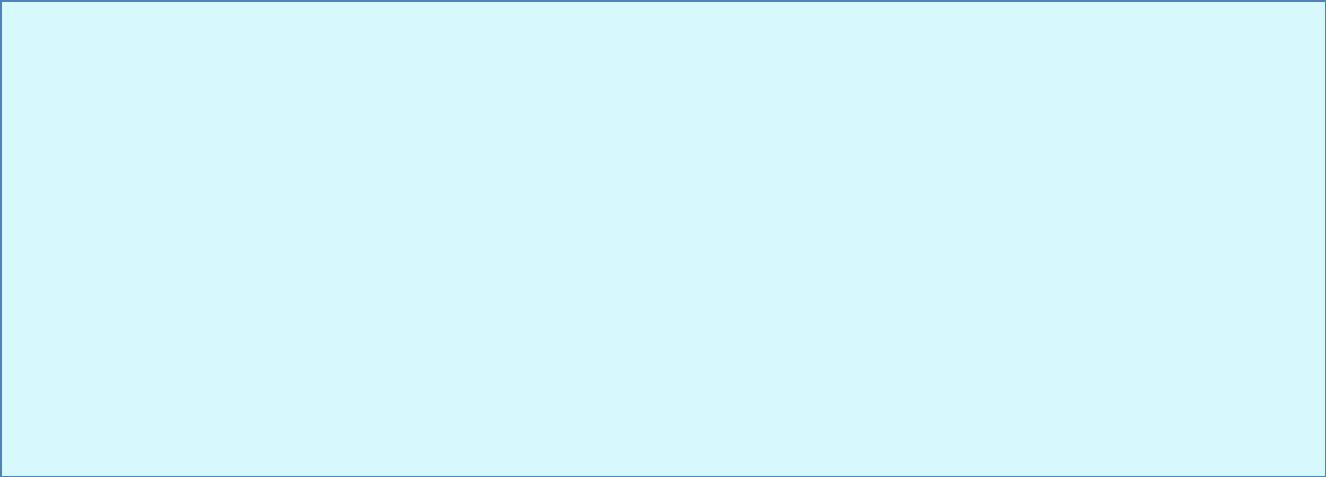
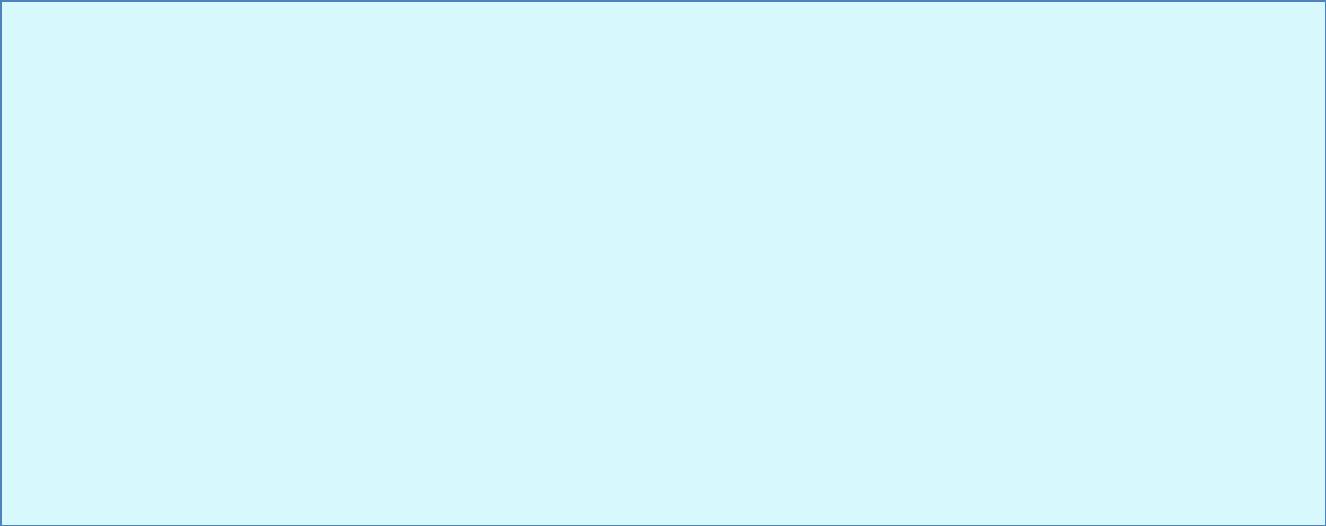
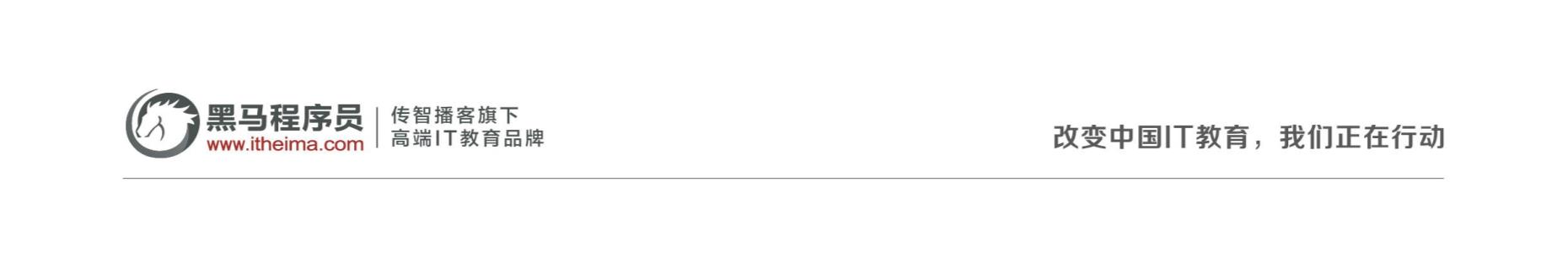
print\_r ($a);

输出结果：Dog

Array ( [b] => Cat [c] => Horse )

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



95.array\_unshift(): 在数组开头插入一个或多个元素

$a=array("a"=>"Cat","b"=>"Dog");

array\_unshift($a,"Horse");

print\_r($a);

输出结果：Array ( [0] => Horse [a] => Cat [b] => Dog )

96.array\_push(): 向数组最后压入一个或多个元素

$a=array("Dog","Cat");

array\_push($a,"Horse","Bird");

print\_r($a);

输入: 目标数组|需要压入的值

输出结果: Array ( [0] => Dog [1] => Cat [2] => Horse [3] => Bird ) （新

的数组）

97.array\_pop(): 取得（删除）数组中的最后一个元素

$a=array("Dog","Cat","Horse");

array\_pop($a);

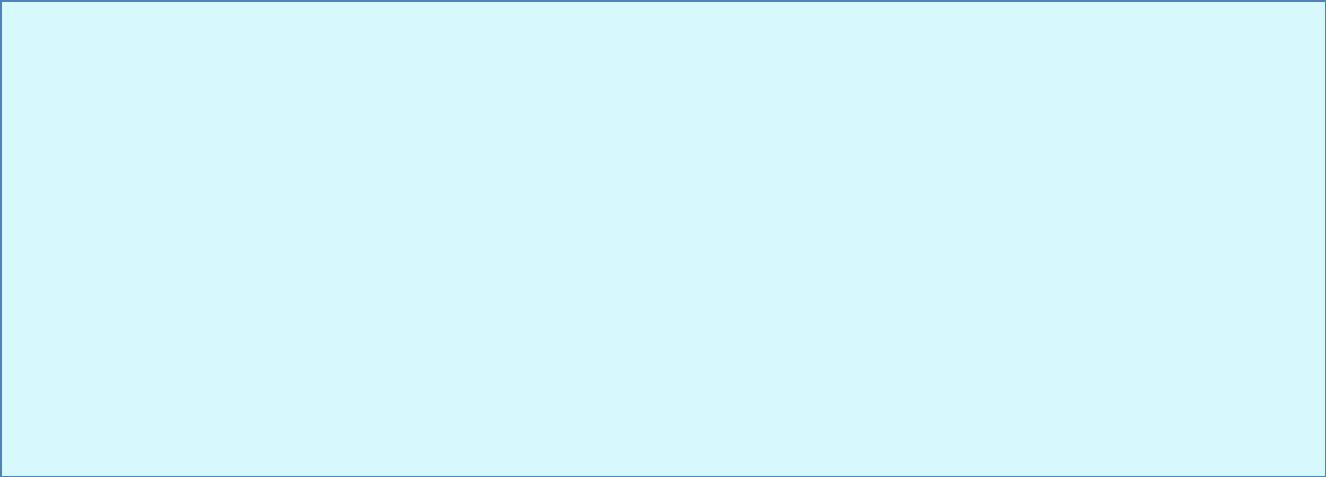
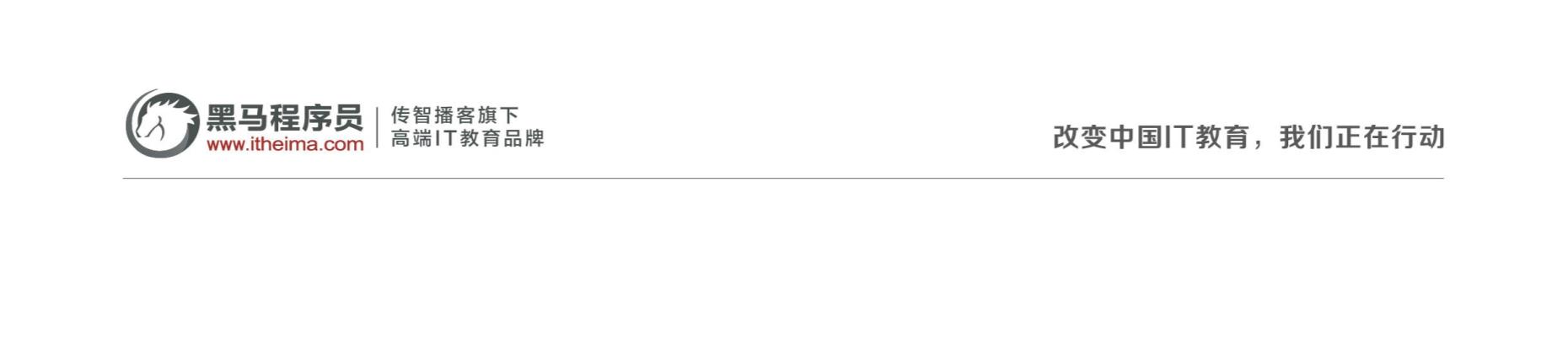
print\_r($a);

输入: 目标数组

输出结果: Array ( [0] => Dog [1] => Cat ) （返回数组剩余元素）

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



十六、数组键值操作

98.shuffle(): 将数组打乱,保留键名

$my\_array = array("a" => "Dog", "b" => "Cat");

shuffle($my\_array);

print\_r($my\_array);

输入: 一个或多个数组

输出结果: 顺序打乱后的数组

99.count(): 计算数组中的单元数目或对象中的属性个数

$people = array("Peter", "Joe", "Glenn", "Cleveland");

$result = count($people);

echo $result;

输入: 目标数组

输出结果: 4（输出元素个数）

100.array\_flip(): 返回一个键值反转后的数组

$a=array(0=>"Dog",1=>"Cat",2=>"Horse");

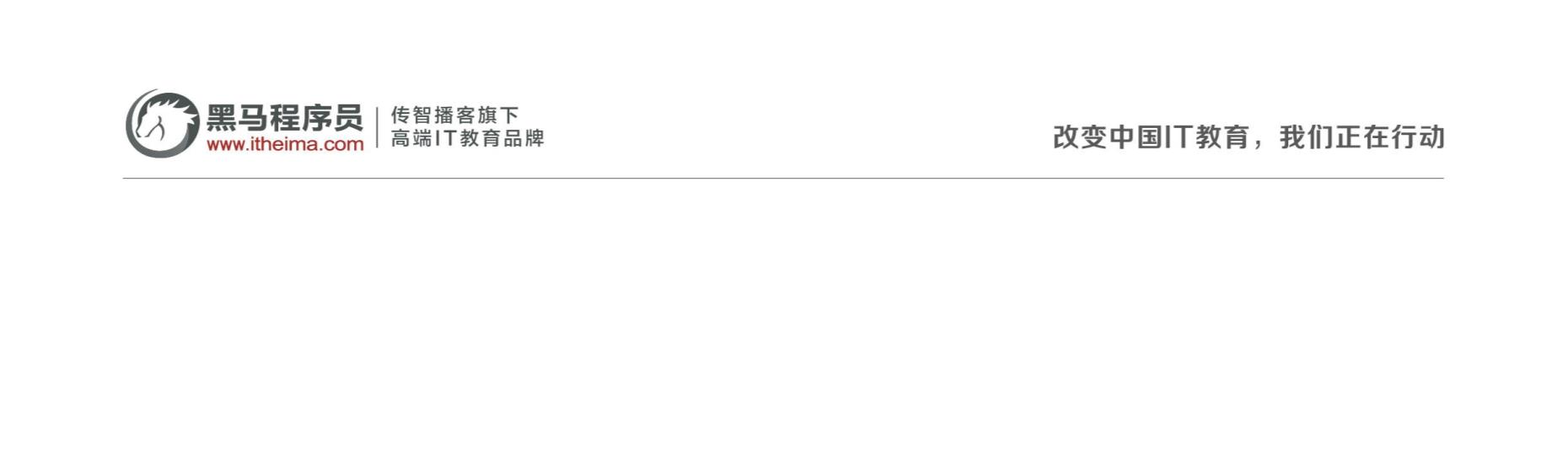
print\_r(array\_flip($a));

输出结果: Array ( [Dog] => 0 [Cat] => 1 [Horse] => 2 )（返回完成后的数

组）

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



101.array\_keys(): 返回数组所有的键,组成一个数组

$a=array("a"=>"Horse","b"=>"Cat","c"=>"Dog");

print\_r(array\_keys($a));

输出结果:Array ( [0] => a [1] => b [2] => c )（返回由键名组成的数组）

102.array\_values(): 返回数组中所有值，组成一个数组

$a=array("Name"=>"Bill","Age"=>"60","Country"=>"USA");

print\_r(array\_values($a));

输出结果: Array ( [0] => Bill [1] => 60 [2] => USA )（返回由键值组成的

数组）

103.array\_reverse(): 返回一个元素顺序相反的数组 元素顺序相反的一个数组，键名和键值依然

匹配

$a=array("a"=>"Volvo","b"=>"BMW","c"=>"Toyota");

print\_r(array\_reverse($a));

输出结果：Array ( [c] => Toyota [b] => BMW [a] => Volvo )

104.array\_count\_values(): 统计数组中所有的值出现的次数

$a=array("Cat","Dog","Horse","Dog");

print\_r(array\_count\_values($a));

输出结果: Array ( [Cat] => 1 [Dog] => 2 [Horse] => 1 )（返回数组原键值

为新键名，次数为新键值）

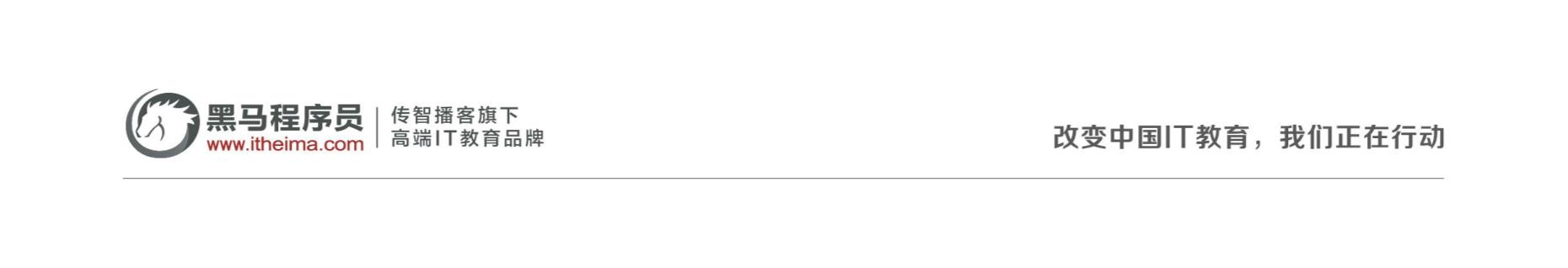
105.array\_rand(): 从数组中随机抽取一个或多个元素,注意是键名!!!

$a=array("a"=>"red","b"=>"green","c"=>"blue","d"=>"yellow");

print\_r(array\_rand($a,2));

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



输入：目标数组, 返回多少随机键名

输出结果：返回 2 个随机的键名

106.each(): 返回数组中当前的键／值对并将数组指针向前移动一步 调用 array each ( array

&$array )

$people = array("Bill", "Steve", "Mark", "David");

print\_r (each($people));

输出结果：Array ( [1] => Bill [value] => Bill [0] => 0 [key] => 0 )

在执行 each() 之后，数组指针将停留在数组中的下一个单元或者当碰到数组结尾时停

留在最后一个单元。如果要再用 each 遍历数组，必须使用 reset()。

返回值: 数组中当前指针位置的键／值对并向前移动数组指针。键值对被返回为四个单

元的数组，键名为 0，1，key 和 value。单元 0 和 key 包含有数组单元的键名，1 和 value

包含有数据。 如果内部指针越过了数组的末端，则 each() 返回 FALSE。

107.array\_unique(): 删除重复值，返回剩余数组

$a=array("a"=>"Cat","b"=>"Dog","c"=>"Cat");

print\_r(array\_unique($a));

输入: 目标数组

输出结果: Array ( [a] => Cat [b] => Dog )（返回无重复值数组，键名不变）

十七、数组排序

108.sort(): 按升序对给定数组的值排序,不保留键名

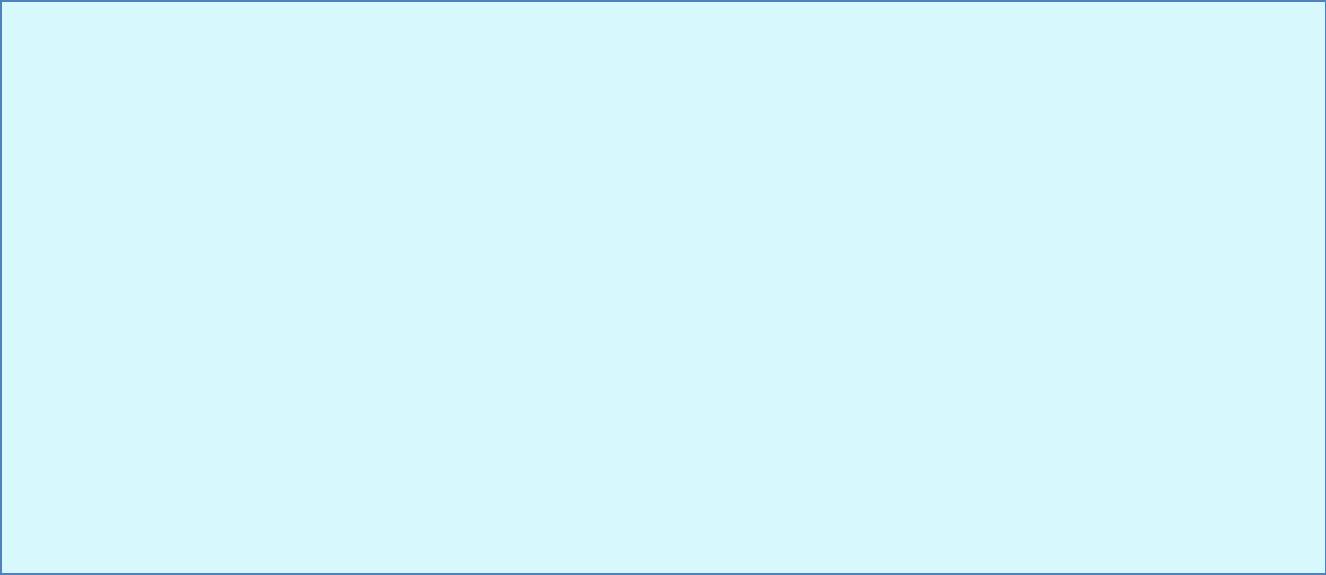
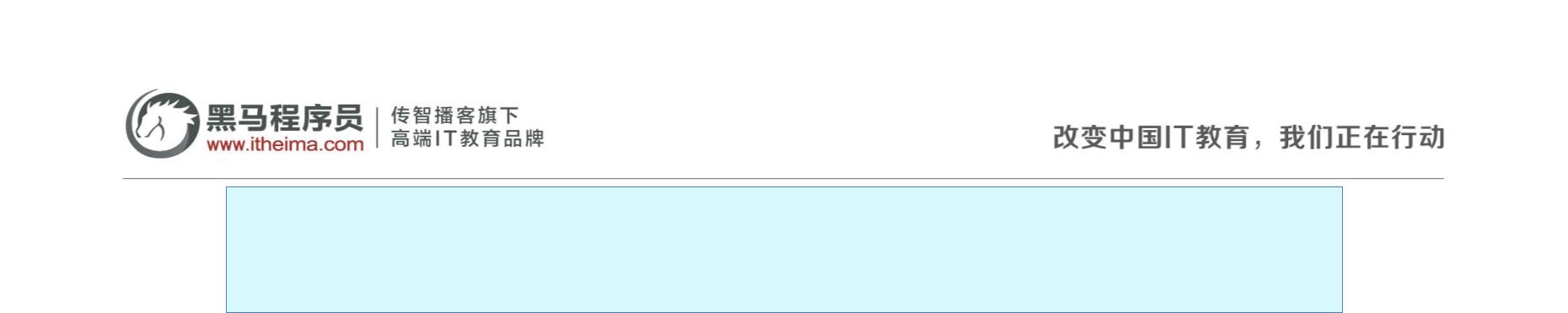
$cars=array("Volvo","BMW","Toyota");

sort($cars);

输出结果：BMW

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



Toyota

Volvo

109.rsort(): 按降序对给定数组的值排序,不保留键名

$cars=array("Volvo","BMW","Toyota");

rsort($cars);

输出结果：Volvo

Toyota

BMW

110.asort(): 按升序对关联数组的键值排序,保持索引关系

$age=array("Bill"=>"60","Steve"=>"56","Mark"=>"31");

asort($age);

输出结果：Key=Mark, Value=31

Key=Steve, Value=56

Key=Bill, Value=60

111.arsort(): 按降序对关联数组的键值排序,保持索引关系

$age=array("Bill"=>"60","Steve"=>"56","Mark"=>"31");

arsort($age);

输出结果：Key=Bill, Value=60

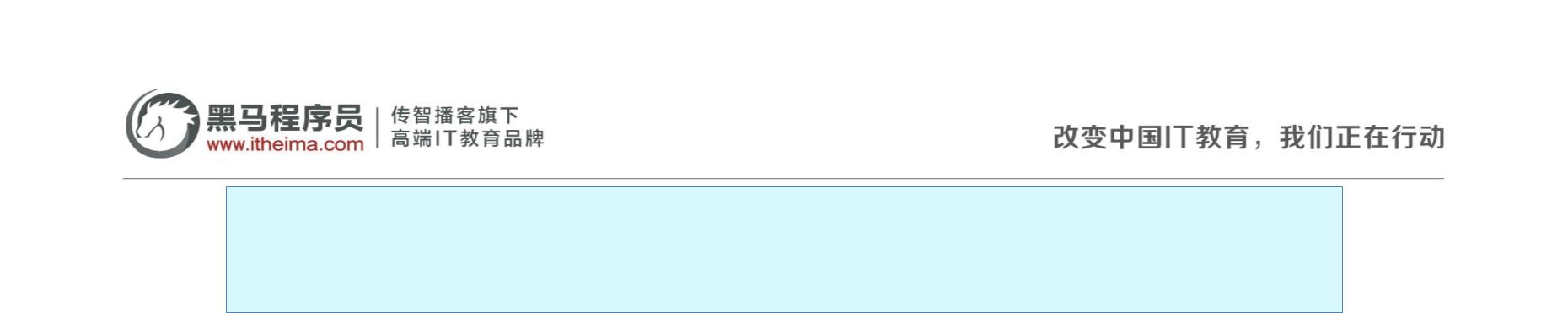
Key=Steve, Value=56

Key=Mark, Value=31

112.ksort(): 按升序对关联数组的键名排序,保持索引关系

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



$age=array("Bill"=>"60","Steve"=>"56","mark"=>"31");

ksort($age);

输出结果：Key=Bill, Value=60

Key=Steve, Value=56

Key=mark, Value=31

113.krsort(): 按降序对关联数组的键名排序,保持索引关系

$age=array("Bill"=>"60","Steve"=>"56","mark"=>"31");

krsort($age);

输出结果：Key=mark, Value=31

Key=Steve, Value=56

Key=Bill, Value=60

114.natsort(): 用自然顺序算法对数组中的元素排序

$temp\_files = array("temp15.txt","temp10.txt","temp1.txt","temp22.txt","temp2.txt");

sort($temp\_files);

echo "标准排序：";

print\_r($temp\_files);

echo "<br>";

natsort($temp\_files);

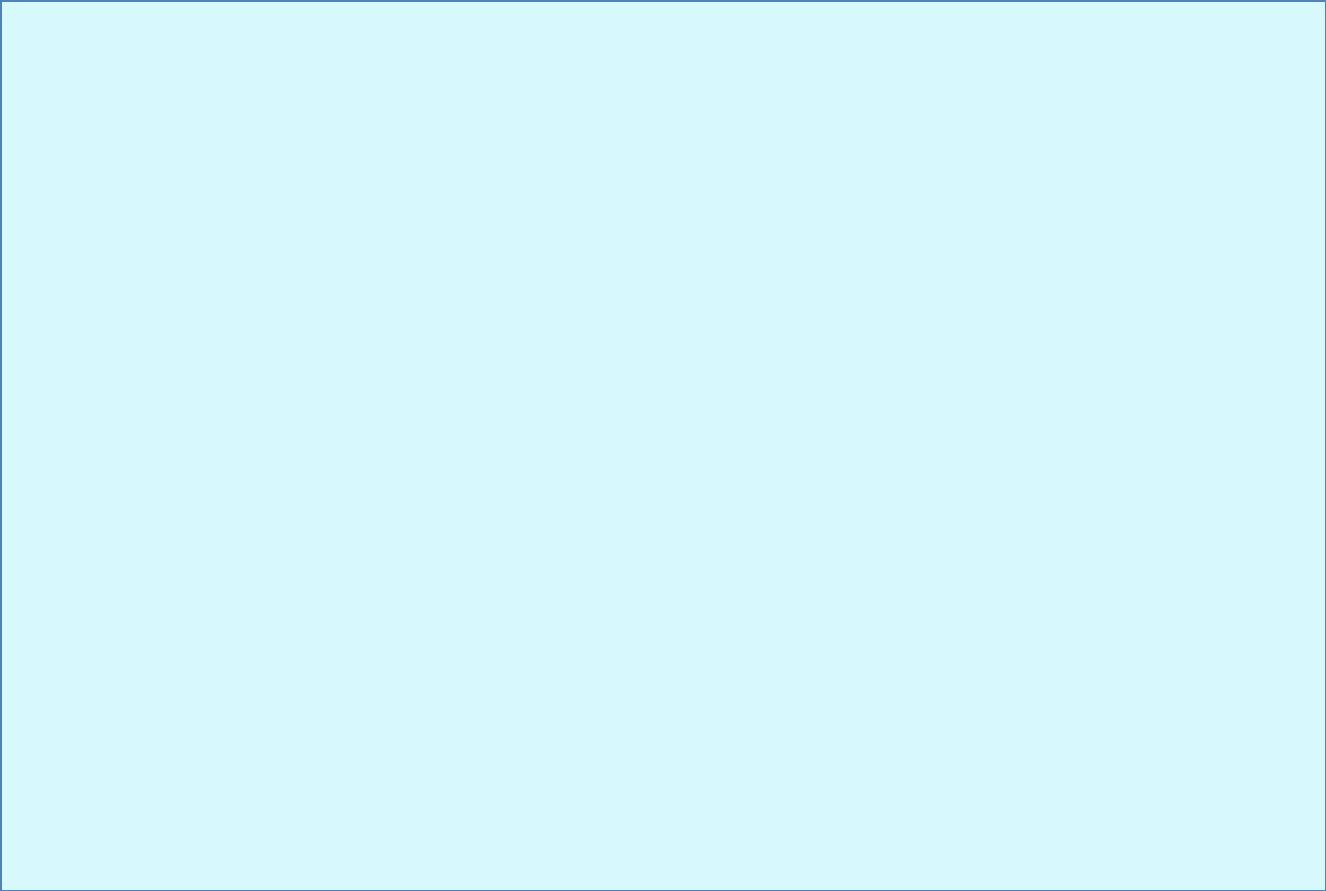
echo "自然排序：";

print\_r($temp\_files);

输出结果：

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



标准排序：Array ( [0] => temp1.txt [1] => temp10.txt [2] => temp15.txt [3] => temp2.txt [4]

=> temp22.txt )

自然排序：Array ( [0] => temp1.txt [3] => temp2.txt [1] => temp10.txt [2] => temp15.txt [4]

=> temp22.txt )

115.natcasesort(): 自然排序,不区分大小写

$temp\_files = array("temp15.txt","Temp10.txt","temp1.txt","Temp22.txt","temp2.txt");

natsort($temp\_files);

echo "自然排序：";

print\_r($temp\_files);

echo "<br />";

natcasesort($temp\_files);

echo "不区分大小写的自然排序：";

print\_r($temp\_files);

输出结果：

自然排序：Array ( [1] => Temp10.txt [3] => Temp22.txt [2] => temp1.txt [4] => temp2.txt [0]

=> temp15.txt )

不区分大小写的自然排序：Array ( [2] => temp1.txt [4] => temp2.txt [1] => Temp10.txt [0]

=> temp15.txt [3] => Temp22.txt )

十八、文件系统函数

116.fopen(): 打开文件或者 URL

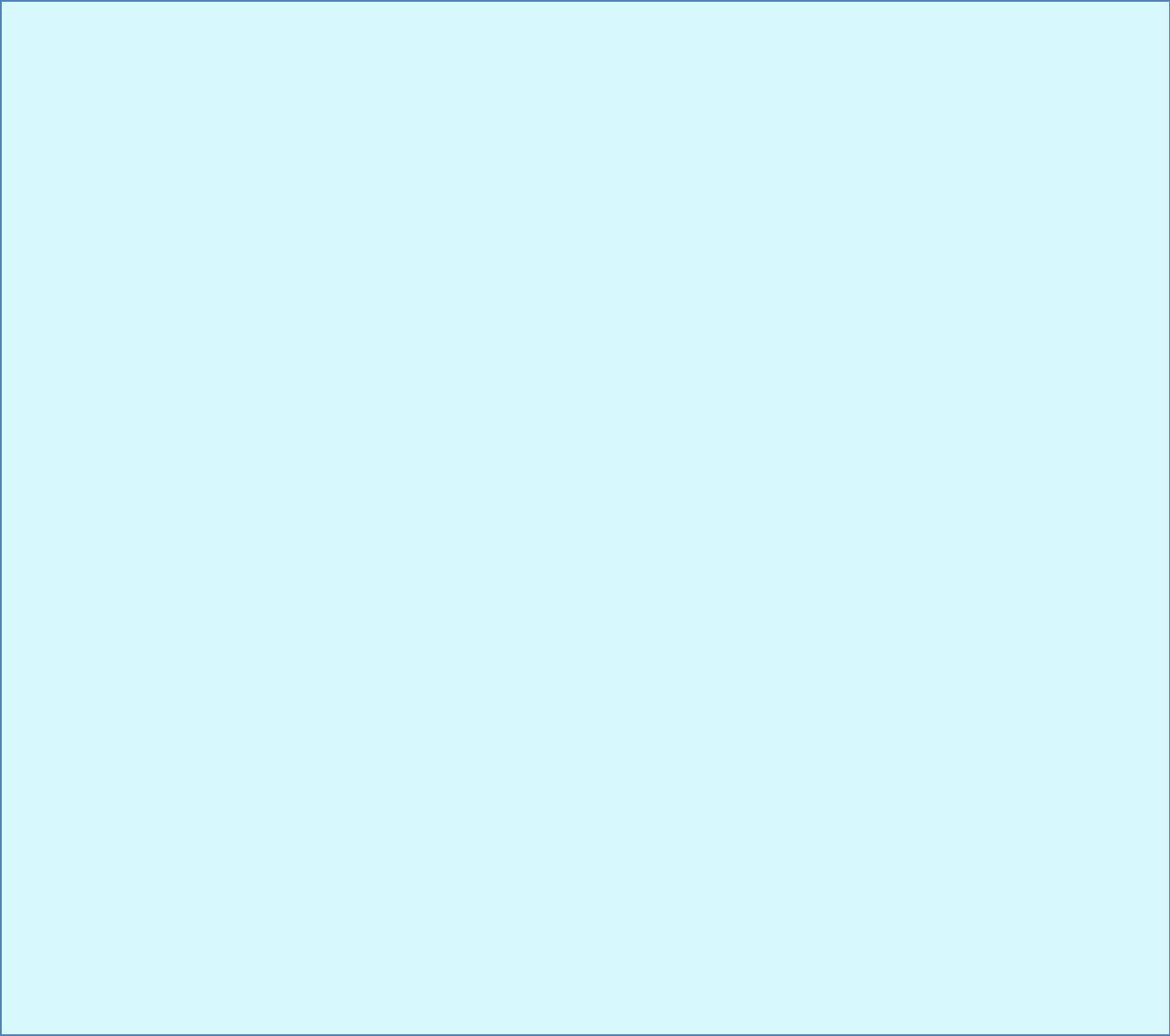
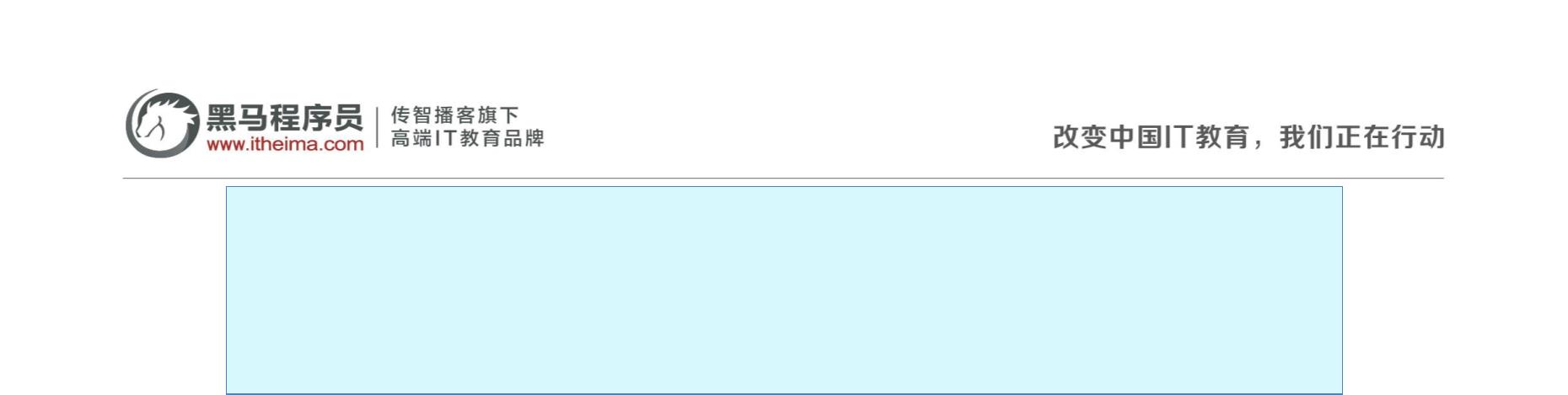
$handle = fopen("ftp://user:password@example.com/somefile.txt", "w");

返回值: 如果打开失败，本函数返回 FALSE

117.fclose(): 关闭一个已打开的文件指针

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



$handle = fopen('somefile.txt', 'r');

fclose($handle);

输出: 如果成功则返回 TRUE，失败则返回 FALSE

十九、文件属性

118.file\_exists(): 检查文件或目录是否存在

$filename = '/path/to/foo.txt';

if (file\_exists($filename)) {

echo "exists";

} else {

echo "does not exist";

}

输入: 指定的文件或目录

输出结果: 存在则返回 TRUE，否则返回 FALSE

119.filesize(): 取得文件大小

$filename = 'somefile.txt';

echo $filename . ': ' . filesize($filename) .'bytes';

输出结果: 返回文件大小的字节数，如果出错返回 FALSE，并生成一条 E\_WARNING 级

的错误

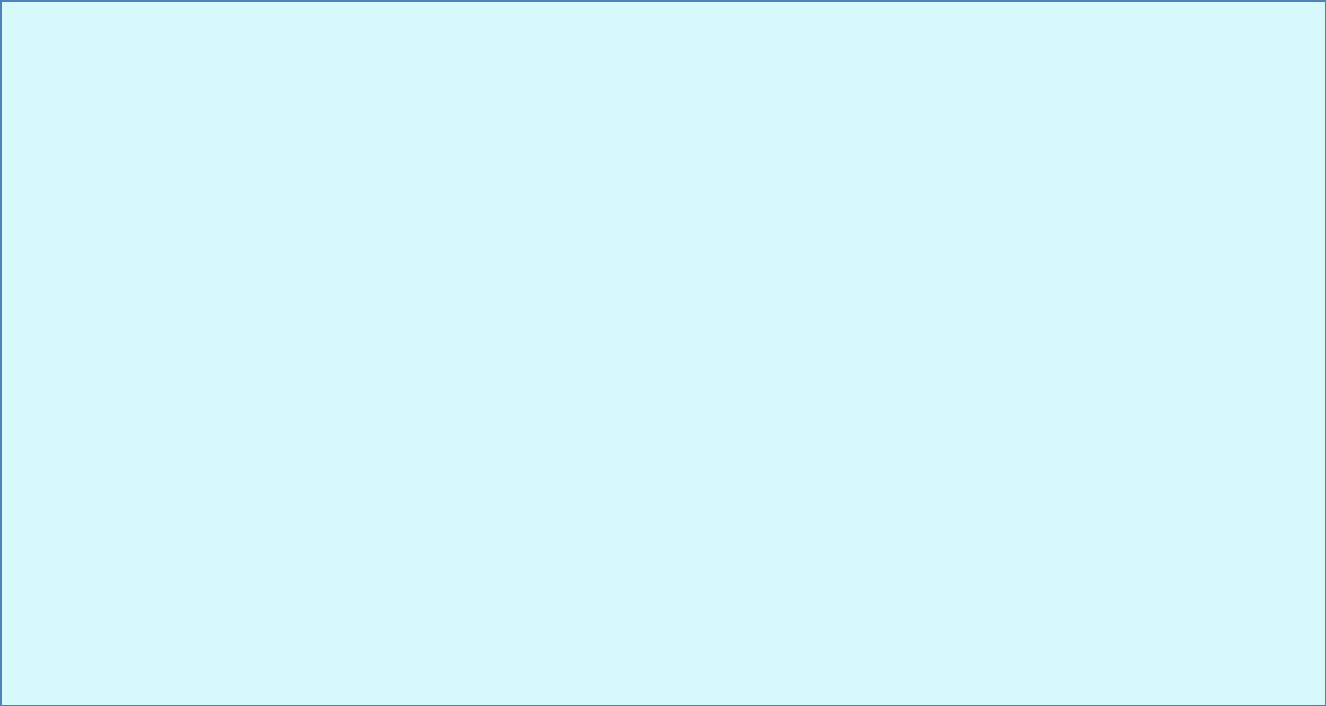
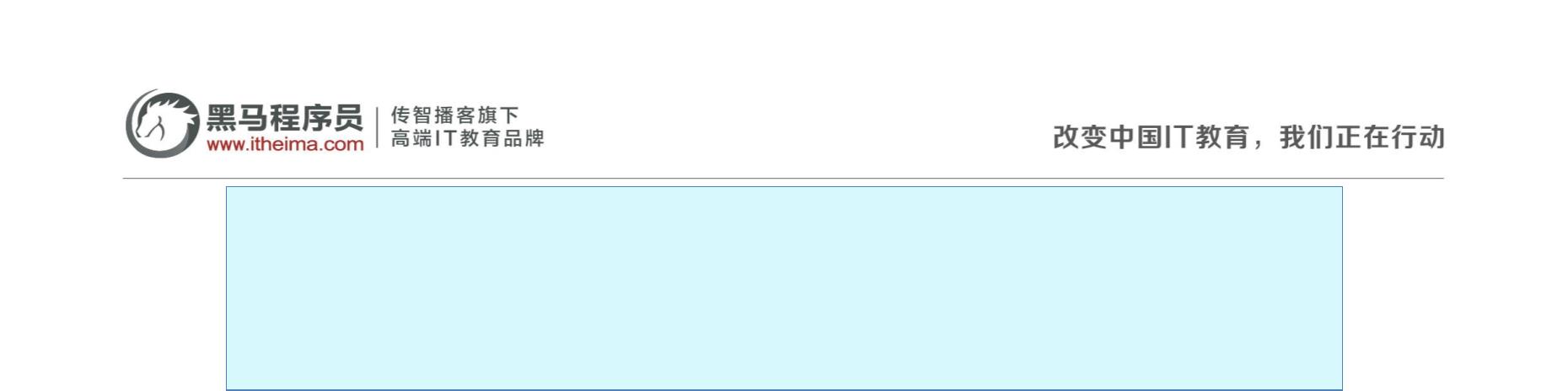
120.is\_readable(): 判断给定文件是否可读

$filename = 'test.txt';

if (is\_readable($filename)) {

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



echo '可读';

} else {

echo '不可读';

}

输出结果: 如果由 filename 指定的文件或目录存在并且可读则返回 TRUE

121.is\_writable(): 判断给定文件是否可写

$filename = 'test.txt';

if (is\_writable($filename)) {

echo '可写';

} else {

echo '不可写';

}

输出结果: 如果文件存在并且可写则返回 TRUE。

122.is\_executable(): 判断给定文件是否可执行

$file = 'setup.exe';

if (is\_executable($file)) {

echo '可执行';

} else {

echo '不可执行';

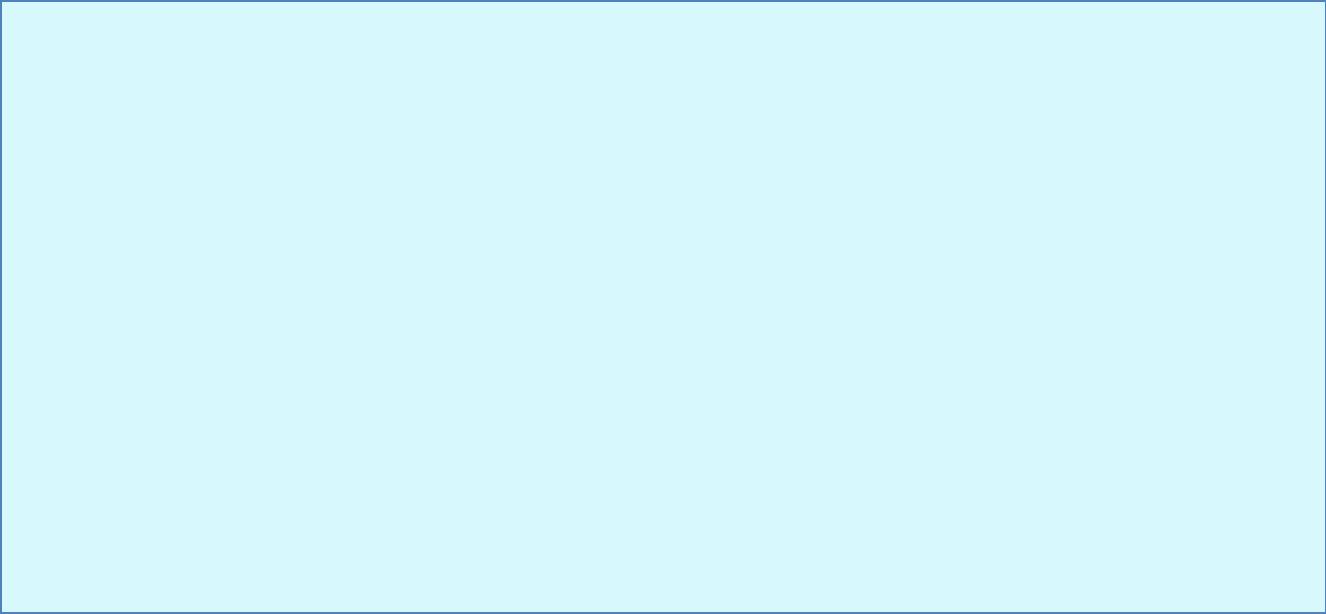
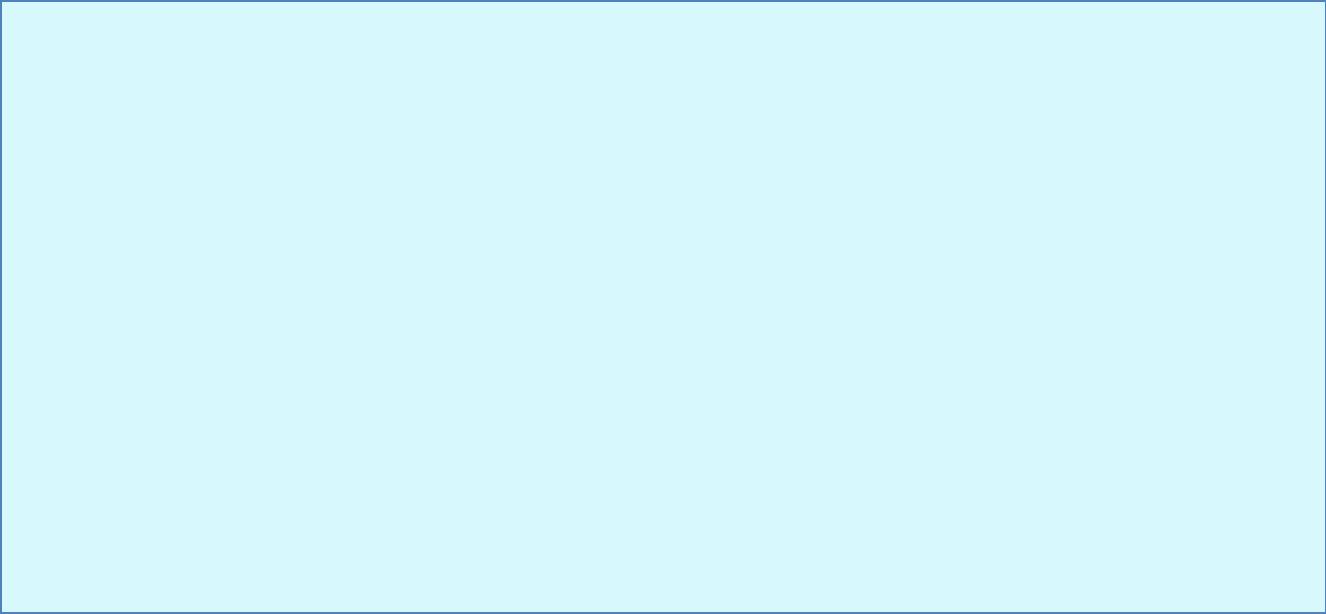
}

输出结果: 如果文件存在且可执行则返回 TRUE

123.filectime(): 获取文件的创建时间

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



$filename = 'somefile.txt';

echo filectime($filename);

输出结果: 时间以 Unix 时间戳的方式返回，如果出错则返回 FALSE

124.filemtime(): 获取文件的修改时间

$filename = 'somefile.txt';

echo filemtime($filename);

输出: 返回文件上次被修改的时间，出错时返回 FALSE。时间以 Unix 时间戳的方式返回

125.fileatime(): 获取文件的上次访问时间

$filename = 'somefile.txt';

echo fileatime($filename);

输出结果: 返回文件上次被访问的时间,如果出错则返回 FALSE.时间以 Unix 时间戳的方

式返回.

126.stat(): 获取文件大部分属性值

$filename = 'somefile.txt';

var\_dump(fileatime($filename));

输出: 返回由 filename 指定的文件的统计信息

二十、文件操作

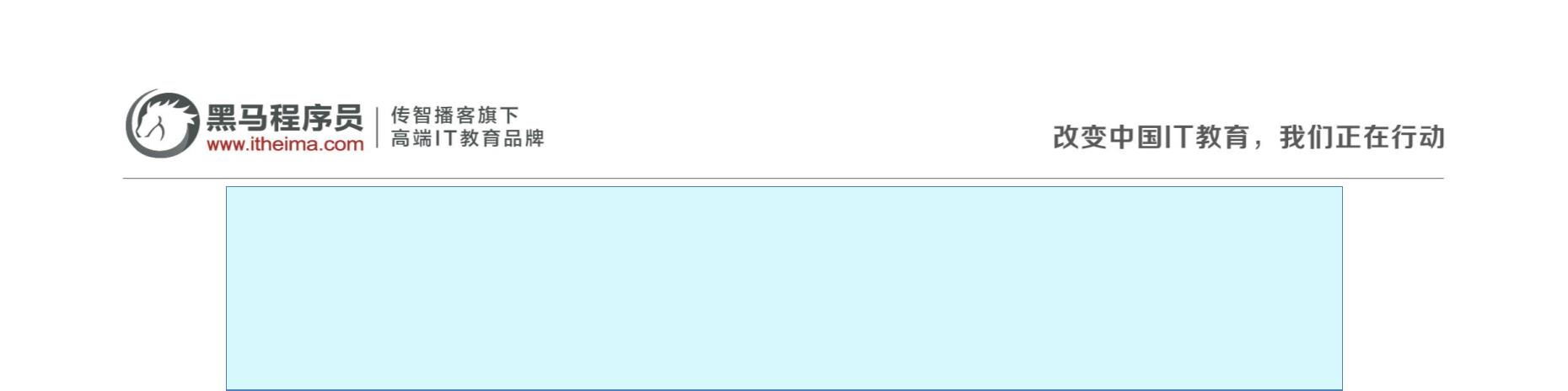
127.fwrite(): 写入文件

$filename = 'test.txt';

$somecontent = "添加这些文字到文件\n";

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



$handle = fopen($filename, 'a');

fwrite($handle, $somecontent);

fclose($handle);

输出结果: 把 string 的内容写入文件指针 handle 处。如果指定了 length,当写入了 length

个字节或者写完了 string 以后，写入就会停止, 视乎先碰到哪种情况

128.fputs(): 同上，是 fwrite 的别名

129.fread(): 读取文件

$filename = "/usr/local/something.txt";

$handle = fopen($filename, "r");

$contents = fread($handle, filesize($filename));

fclose($handle);

输出结果：从文件指针 handle，读取最多 length 个字节

130.feof(): 检测文件指针是否到了文件结束的位置

$file = @fopen("no\_such\_file", "r");

while (!feof($file)) {

}

fclose($file);

输出: 如果文件指针到了 EOF 或者出错时则返回 TRUE，否则返回一个错误（包括

socket 超时），其它情况则返回 FALSE

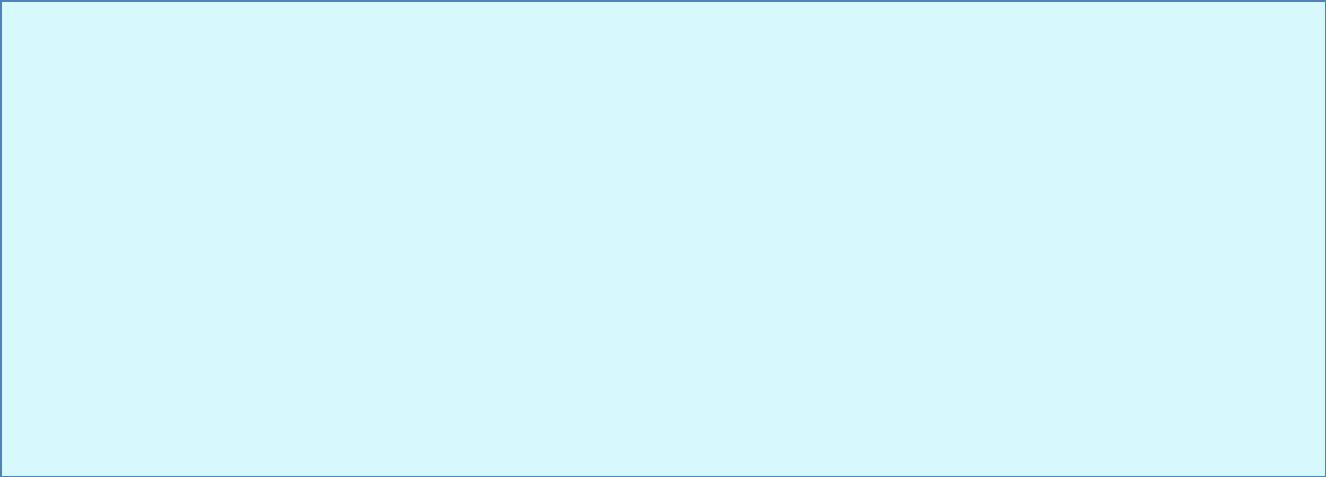
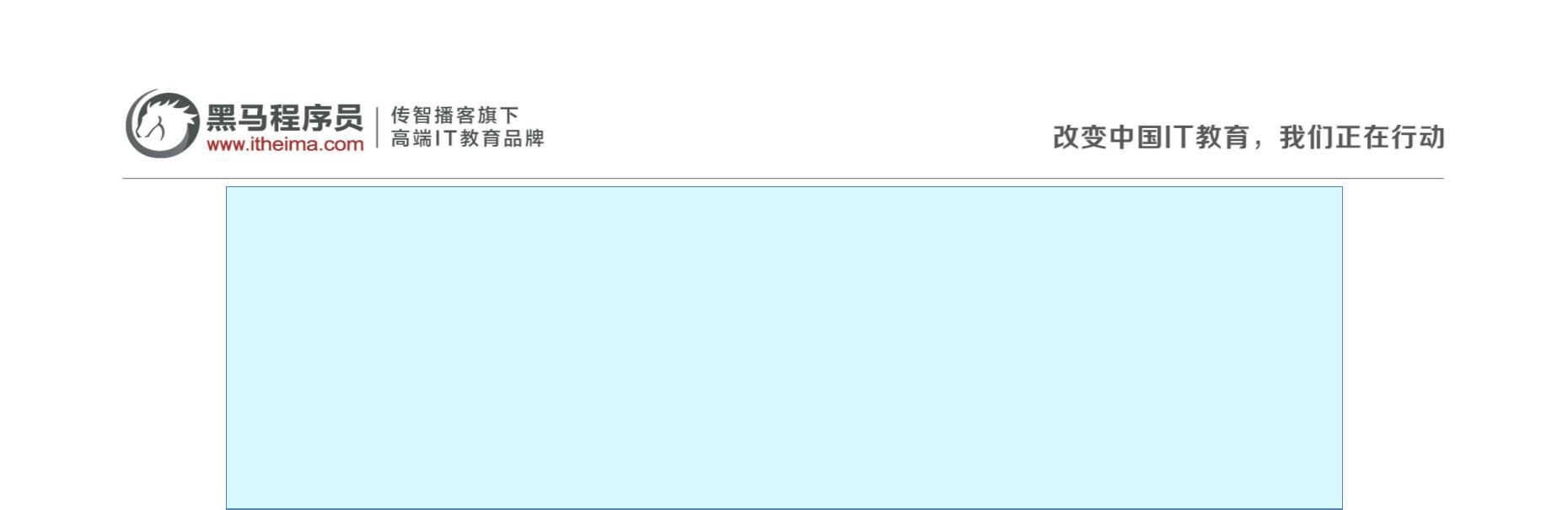
131.fgets(): 从文件指针中读取一行

$handle = @fopen("/tmp/inputfile.txt", "r");

if ($handle) {

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



while (!feof($handle)) {

$buffer = fgets($handle, 4096);

echo $buffer;

}

fclose($handle);

}

输出结果: 从 handle 指向的文件中读取一行并返回长度最多为 length-1 字节的字符串.

碰到换行符（包括在返回值中）、EOF 或者已经读取了 length -1 字节后停止(看先碰到那一

种情况). 如果没有指定 length,则默认为 1K, 或者说 1024 字节.

132.fgetc(): 从文件指针中读取字符

$fp = fopen('somefile.txt', 'r');

if (!$fp) {

echo 'Could not open file somefile.txt';

} while (false !== ($char = fgetc($fp))) {

echo "$char\n";

}

输入: 目标文件中要读取的字符

输出: 返回一个包含有一个字符的字符串，该字符从 handle 指向的文件中得到. 碰到

EOF 则返回 FALSE.

133.file(): 把整个文件读入一个数组中

$lines = file('http://www.example.com/'); //在数组中循环，显示 HTML 的源文件并加

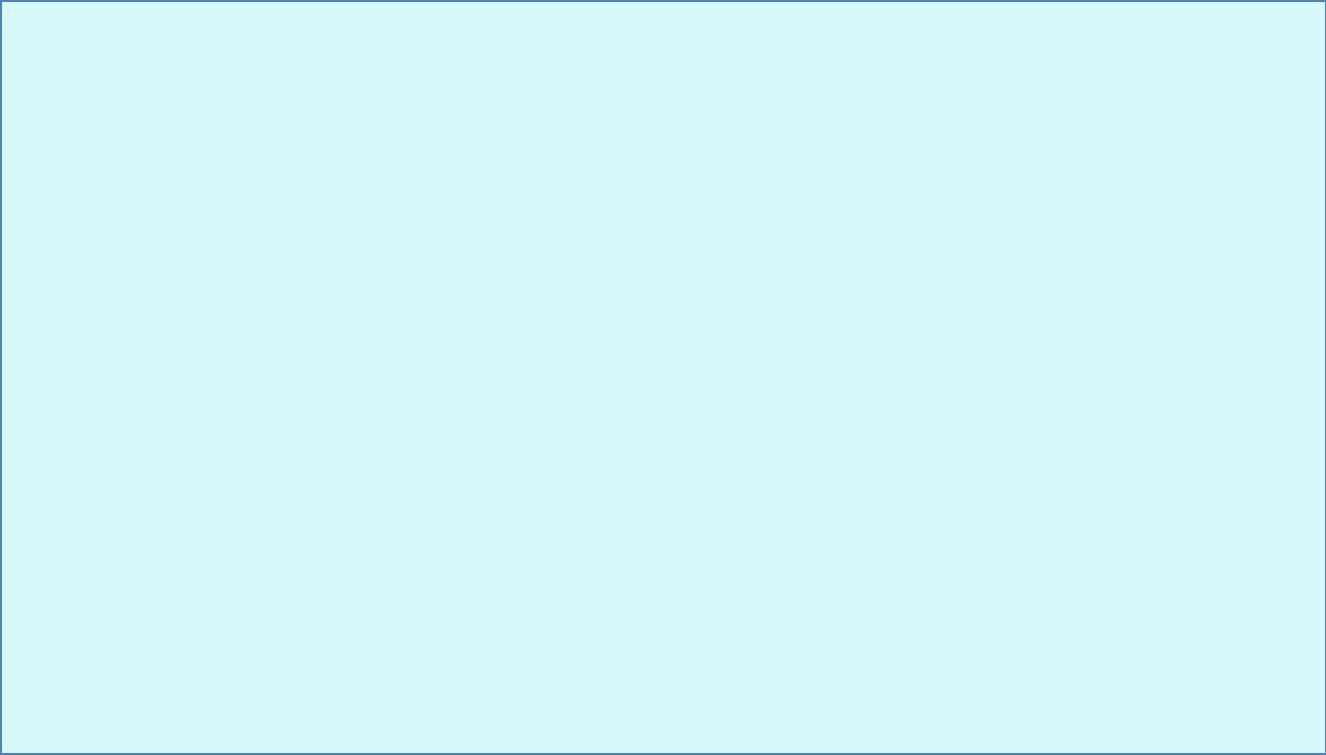
上行号。

foreach ($lines as $line\_num => $line) {

echo "Line #<b>{$line\_num}</b> : " .

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



htmlspecialchars($line) . "<br />\n";

}

//另一个例子将 web 页面读入字符串。参见 file\_get\_contents()。

$html = implode('', file('http://www.example.com/'));

输出: 数组中的每个单元都是文件中相应的一行，包括换行符在内。如果失败 file() 返

回 FALSE

134.readfile(): 输出一个文件

echo readfile("test.txt");

输出结果:There are two lines in this file.

This is the last line.

57（读入一个文件并写入到输出缓冲。返回从文件中读入的字节数。如果出

错返回 FALSE）

135.file\_get\_contents(): 将整个文件读入一个字符串

echo file\_get\_contents('http://www.baidu.com');

输出结果：百度的页面

136.file\_put\_contents():将一个字符串写入文件

echo file\_put\_contents("test.txt","Hello World. Testing!");

输出结果:26（该函数将返回写入到文件内数据的字节数）

137.ftell(): 返回文件指针读/写的位置

$file = fopen("test.txt","r");

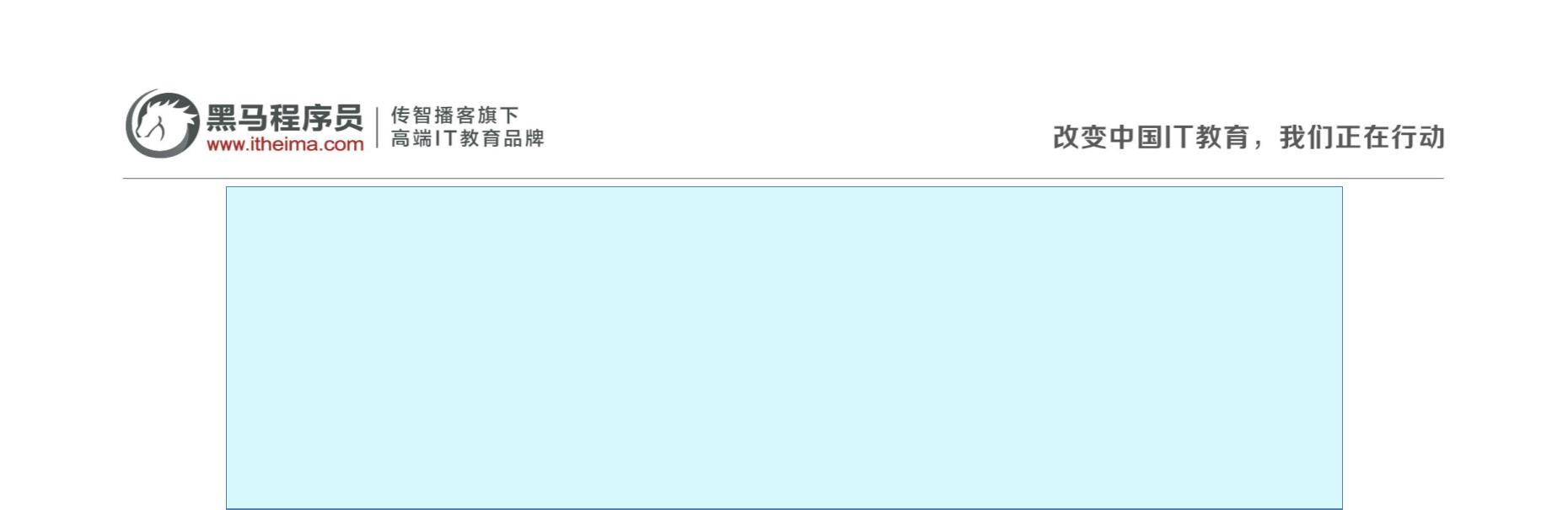
echo ftell($file);

fseek($file,"15");

echo "<br />" . ftell($file);

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



fclose($file);

输出结果:0

15（返回由 handle 指定的文件指针的位置，也就是文件流中的偏移量）

138.fseek(): 在文件指针中定位

$file = fopen("test.txt","r");

// 读取第一行

fgets($file);

// 倒回文件的开头

fseek($file,0);

输出结果: 成功则返回 0；否则返回 -1

139.rewind(): 将文件指针的位置倒回文件的开头

$file = fopen("test.txt","r");

//改变文件指针的位置

fseek($file,"15");

//把文件指针设定为 0

rewind($file);

输出结果：如果成功则返回 TRUE，失败则返回 FALSE

140.flock(): 轻便的执行文件锁定

$fp=fopen('tx.txt','r');

flock($fp, LOCK\_SH);

flock($fp, LOCK\_EX);

flock($fp, LOCK\_NB);

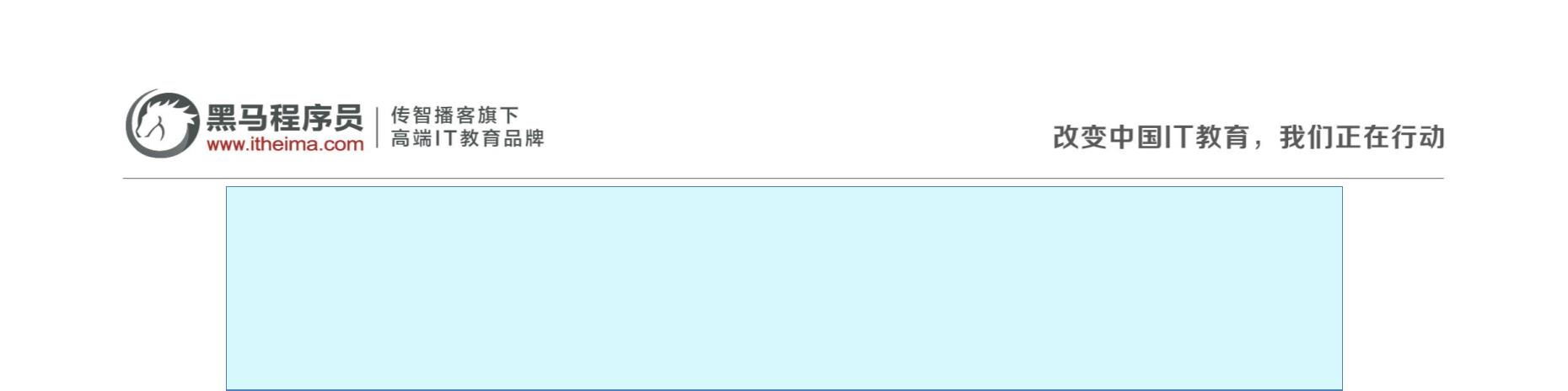
//共享锁

//独立锁，写文件时用它打开

//附加锁

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



flock($fp, LOCK\_UN); //释放锁

fclose($fp);

输出: 如果成功则返回 TRUE，失败则返回 FALSE

二十一、目录

141.basename(): 返回路径中的文件名部分

$path = "/testweb/home.php";

//显示带有文件扩展名的文件名

echo basename($path);

//显示不带有文件扩展名的文件名

echo basename($path,".php");

输出结果: home.php

home

142.dirname(): 返回路径中的目录部分

echo dirname("c:/testweb/home.php");

echo dirname("/testweb/home.php");

输出结果: c:/testweb

/testweb

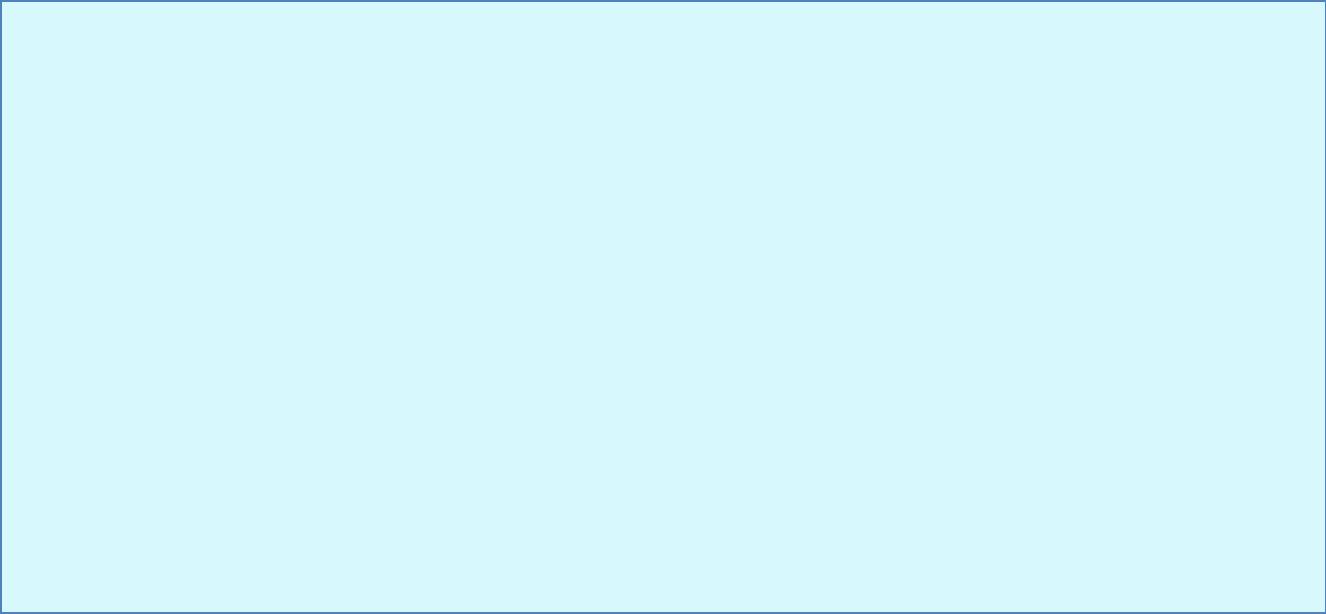
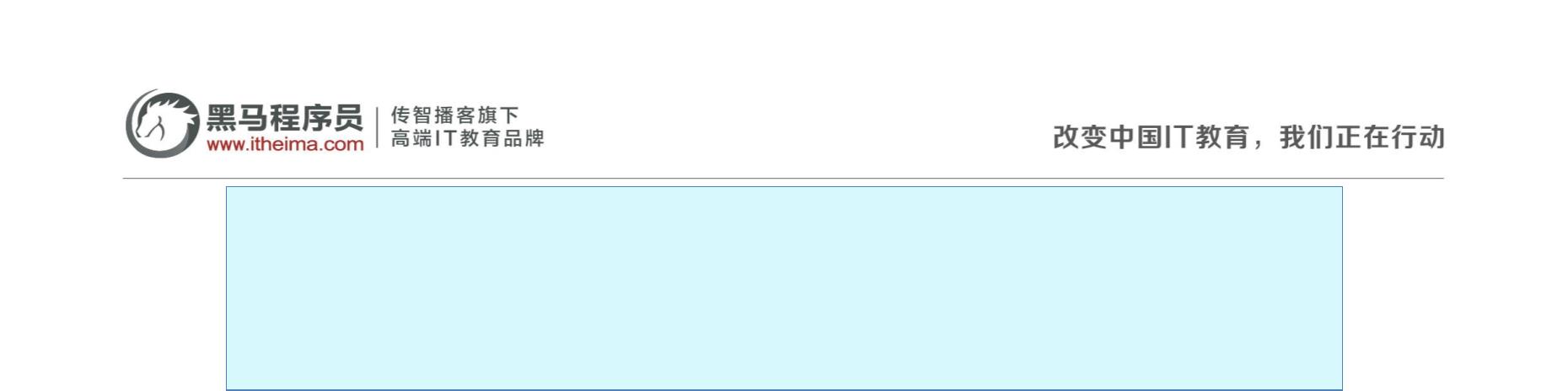
143.pathinfo(): 返回文件路径的信息

print\_r(pathinfo("/testweb/test.txt"));

输出结果：

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



Array

(

[dirname] => /testweb

[basename] => test.txt

[extension] => txt

) （返回一个关联数组包含有 path 的信息）

144.opendir(): 打开目录句柄（打开一个目录，读取它的内容，然后关闭）

$fp=opendir('/imges/');

echo readdir($fp);

closedir($fp);

输出结果：cat.gif（如果成功则返回目录句柄的 resource，失败则返回 FALSE）

145.readdir(): 从目录句柄中读取条目（返回目录中下一个文件的文件名）

$fp=opendir('/imges/');

echo readdir($fp);

closedir($fp);

输出结果：cat.gif（如果成功则返回目录句柄的 resource，失败则返回 FALSE）

146.closedir(): 关闭目录句柄

$fp=opendir('/imges/');

echo readdir($fp);

closedir($fp);

输出结果：cat.gif（如果成功则返回目录句柄的 resource，失败则返回 FALSE）

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



147.rewinddir() : 倒回目录句柄

$fp=opendir('/images/');

echo readdir($fp).'<br />';

rewinddir($fp);

echo readdir($fp).'<br />';

closedir($fp);

输出结果: 指定的目录流重置到目录的开头

148.mkdir(): 新建目录

mkdir('123');

输出结果: 新建一个指定的目录

149.rmdir(): 删除目录

rmdir('123');

输出结果: 删除指定的目录。目录必须是空的，而且要有相应的权限。如果成功则返回

TRUE，失败则返回 FALSE

150.unlink(): 删除文件

unlink('123/1.txt');

rmdir('123');

输出结果: 删除 1.txt

151.copy(): 拷贝文件

echo copy("source.txt","target.txt");

输出结果: 1

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



152.rename(): 重命名一个文件或目录

rename("images","pictures");

输出: 如果成功则返回 TRUE，失败则返回 FALSE

二十二、文件的上传与下载

153.is\_uploaded\_file():判断文件是否是通过 HTTP POST 上传的

$file = "test.txt";

if(is\_uploaded\_file($file)){ echo ("$file is uploaded via HTTP POST");

}

else{ echo ("$file is not uploaded via HTTP POST");

}

输出结果：test.txt is not uploaded via HTTP POST

154.move\_uploaded\_file(): 将上传的文件移动到新位置

if(is\_uploaded\_file($\_FILES['bus']['tmp\_name'])){

if( move\_uploaded\_file($\_FILES['bus']['tmp\_name'], $NewPath) ){

echo '上传成功<br /><img src="'.$NewPath.'">';

}else{ exit('失败');

}

}else{ exit('不是上传文件');

}

二十三、时间函数

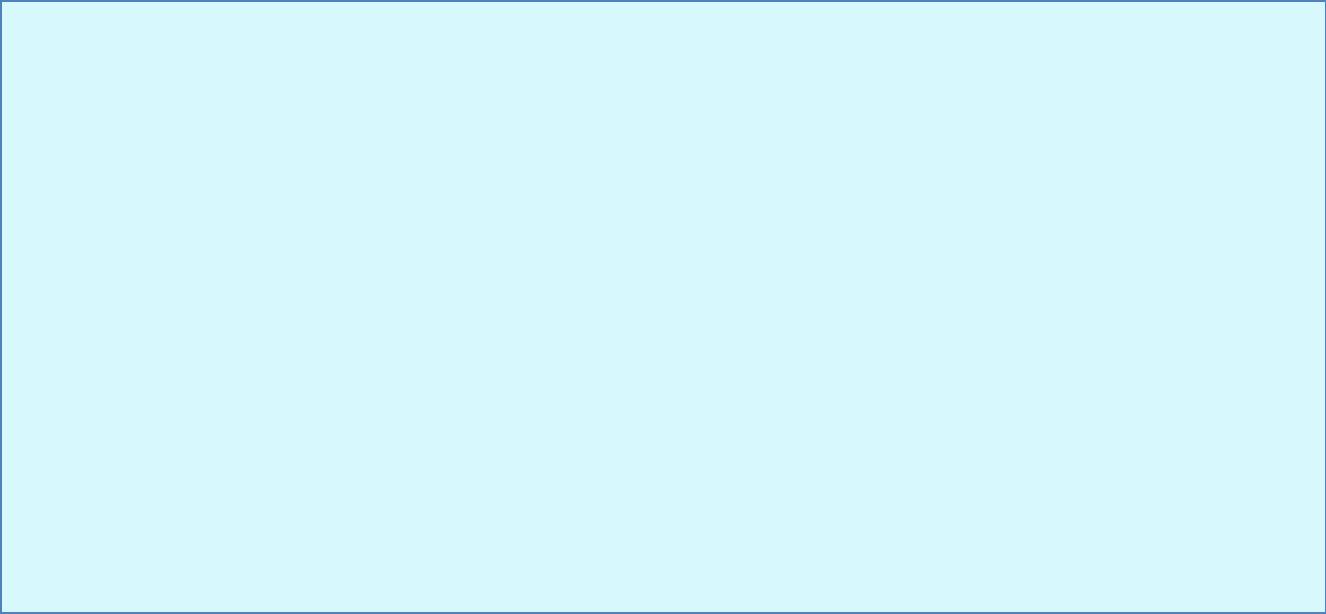
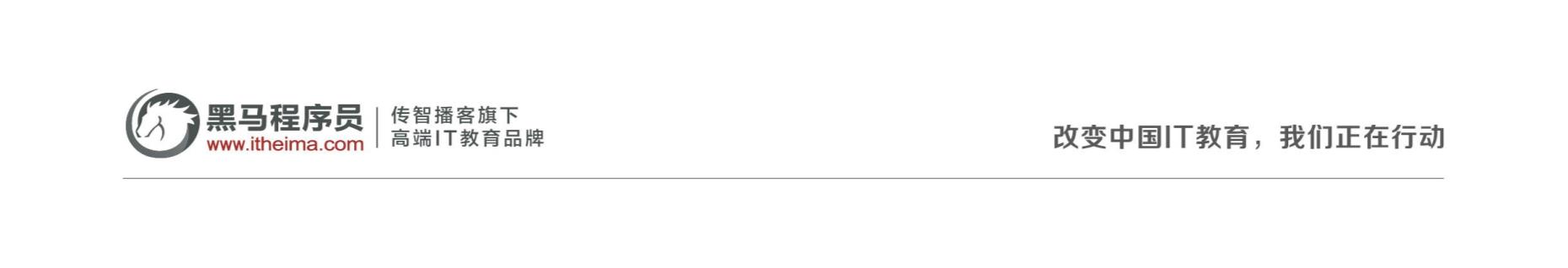
155.time(): 返回当前的 Unix 时间戳 time();

$t=time();

echo($t . "<br>");

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



echo(date("Y-m-d",$t));

输出结果: 1494396195

2017-05-10（返回自从 Unix 纪元（格林威治时间 1970 年 1 月 1 日

00:00:00）到当前时间的秒数）

156.mktime(): 取得一个日期的 Unix 时间戳，使用它来查找该日期的天

echo "Oct 3, 1975 was on a ".date("l", mktime(0,0,0,10,3,1975));

输出结果：Oct 3, 1975 was on a Friday

156.date(): 格式化一个本地时间／日期

echo date("Y-m-d h:i:s ");

输出: 2017-05-10 06:11:06

157.checkdate(): 验证一个格里高里日期

var\_dump(checkdate(12,31,-400));

echo "<br>";

var\_dump(checkdate(2,29,2003));

echo "<br>";

var\_dump(checkdate(2,29,2004));

输出结果：bool(false)

bool(false)

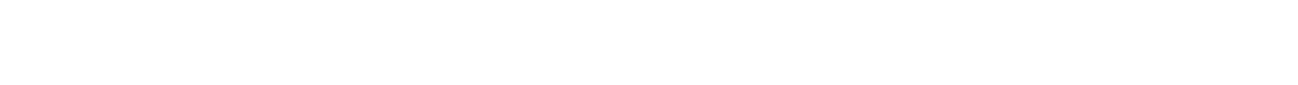
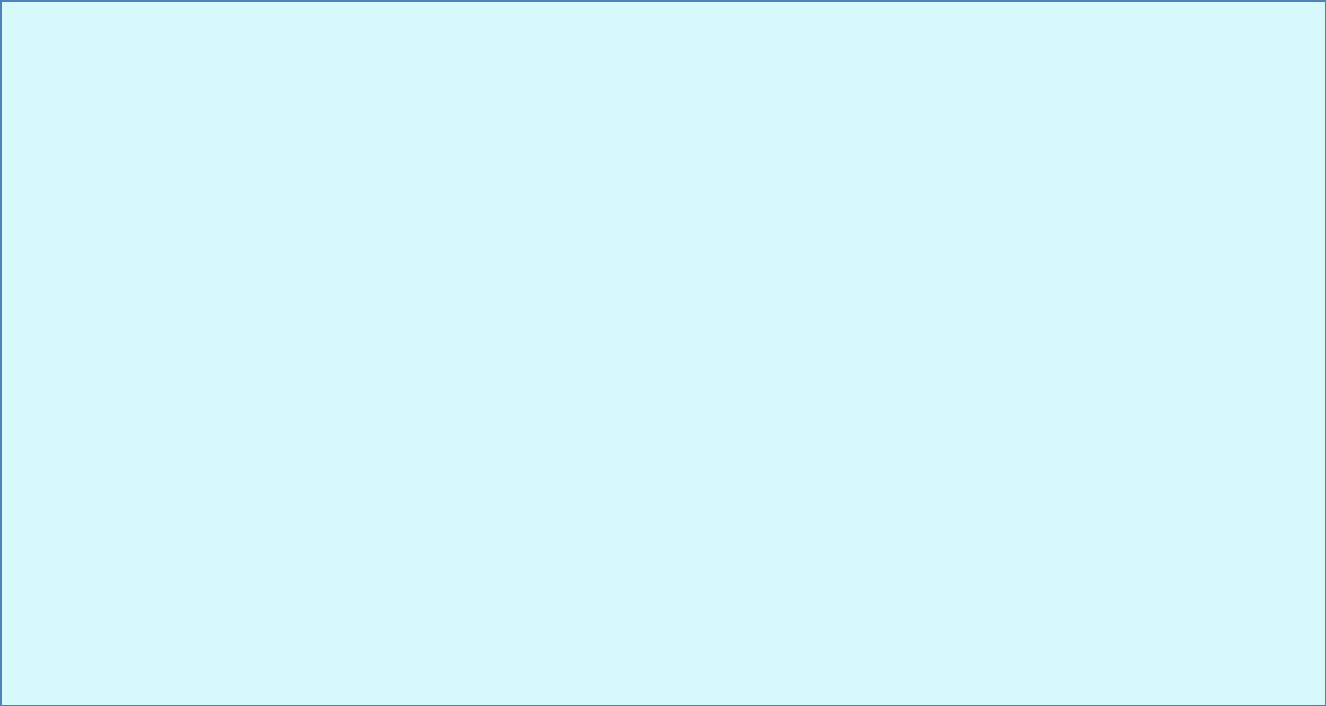
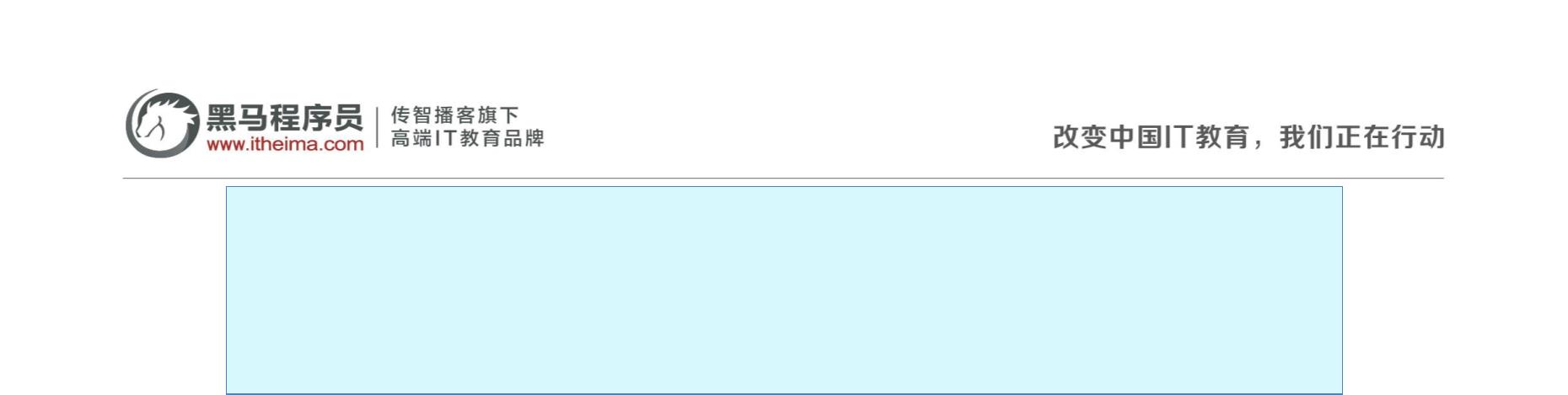
bool(true)

158.date\_default\_timezone\_set(): 设定用于一个脚本中所有日期时间函数的默认时区

date\_default\_timezone\_set('PRC');

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



159.getdate(): 取得日期／时间信息

print\_r(getdate());

输出结果: Array ( [seconds] => 42 [minutes] => 18 [hours] => 6 [mday]

=> 10 [wday] => 3 [mon] => 5 [year] => 2017 [yday] => 129 [weekday] =>

Wednesday [month] => May [0] => 1494397122 ) （返回一个根据 timestamp 得出

的包含有日期信息的关联数组。如果没有给出时间戳则认为是当前本地时间）

160.strtotime(): 将任何英文文本的日期时间描述解析为 Unix 时间戳

echo strtotime("now");

输出结果：1494397277

161.microtime(): 返回当前 Unix 时间戳和微秒数

echo(microtime());

输出结果：0.00806100

1494398062

二十四、其他常用:

162.intval(): 获取变量的整数值

echo intval(0.99);

变量转成整数类型

输出结果：0（向下取整）

163.PDO 类的相关函数 prepare() execute() fetch()

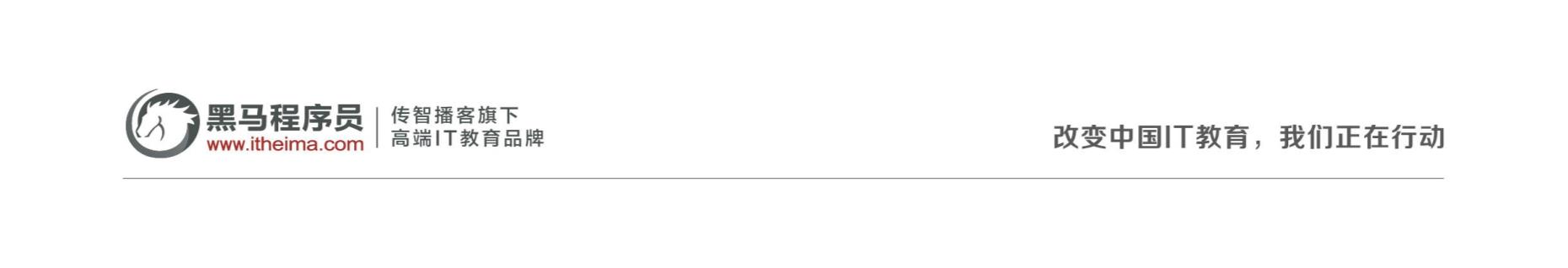
<?php$driver = 'mysql';

$database = "dbname=CODINGGROUND";

$dsn = "$driver:host=localhost;unix\_socket=/home/cg/mysql/mysql.sock;$database";

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



$username = 'root';

$password = 'root';try {

$conn = new PDO($dsn, $username, $password);

echo "<h2>Database CODINGGROUND Connected<h2>";

}catch(PDOException $e){

echo "<h1>" . $e->getMessage() . "</h1>";

}

$sql = 'SELECT \* FROM users';

$stmt = $conn->prepare($sql);

$stmt->execute();

echo "<table style='width:100%'>";

while($row = $stmt->fetch(PDO::FETCH\_ASSOC)){

echo "<tr>";

foreach($row as $value){

echo sprintf("<td>%s</td>", $value);

} echo "</tr>";

}echo "</table>";?>

164.正则表达式-元字符

元字符及其匹配范围

\d 匹配任意一个十进制数字,等价于: [0-9]

\D 匹配除十进制数字以外的任意数字,等价于: [^0-9]

\s:匹配空白字符,等价于: [\n\f\r\t\v]

\S: 匹配除空白字符以外的任意一个字符, 等价于[^\n\f\r\t\v]

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



\w 匹配任意一个数字、字母和下划线,等价于: [0-9a-zA-Z\_]

\W 匹配除字母、数字和下划线以外的任意字符, 等价于: [^0-9a-zA-Z\_]

[]：1)用来表示范围。2)匹配任意一个中括号中定义的原子

[^]: 中括号里面的^(抑扬符)：表示匹配任意一个除中括号里面定义的原子

限定次数

\* 匹配 0 次、1 次或多次其前的原子, 等价于: {0,}

+ 匹配 1 次或多次其前的原子, 等价于: {1,}

? 匹配 0 次或 1 次其前的原子, 等价于: {0,1}

{n} 表示其前的原子正好出现 n 次, 等价于: {n,} 表示其前的原子至少出现 n 次，最多不限制

{m,n} 表示其前的原子最少出现 m 次，最多出现 n 次

其它

. 匹配除换行符(\n)以外的任意字符【windows 下还匹配\f\r】 | 两个或多个分支选择【优先

级最低】

^ 匹配输入字符的开始位置

$ 匹配输入字符的结束位置

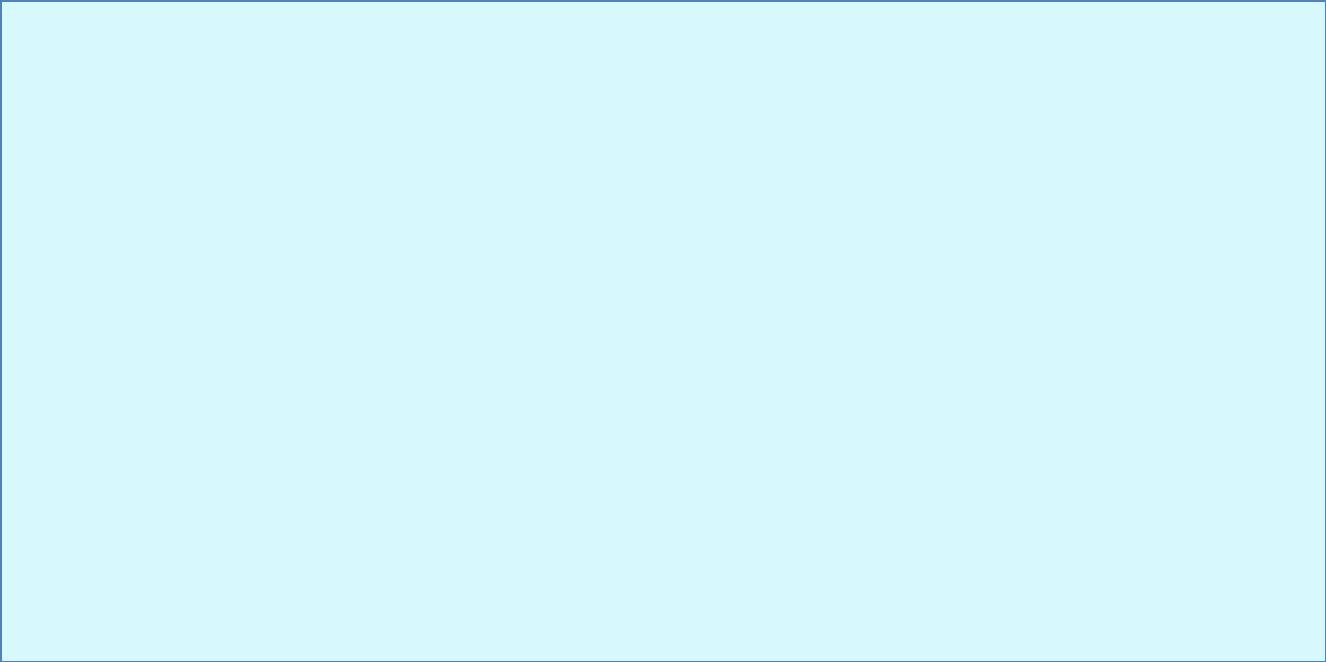
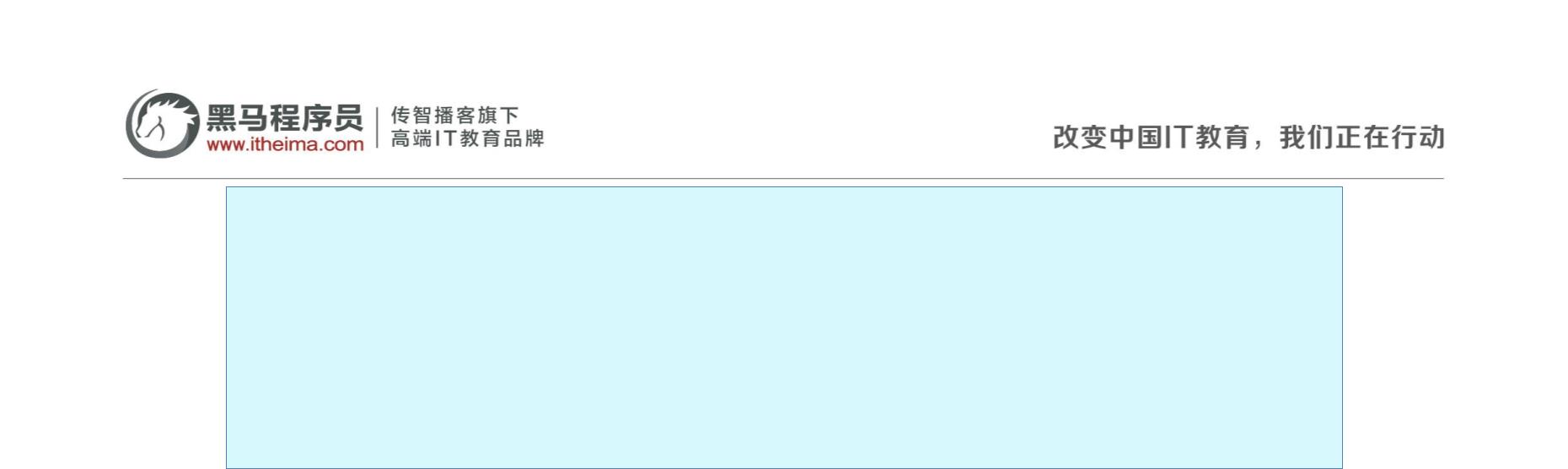
\b 匹配词边界

\B 匹配非词边界

() 1)模式单元，把多个小原子组成一个大原子。2)可以改变优先级

北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090



北京市昌平区建材城西路金燕龙办公楼一层

电话：400-618-9090

