

第9章 PHP常用函数

+ 在编程的过程中，我们常常要做一些同样的操作，因此我们有时候就会常常编写一个执行相同功能的函数。经过长期的收集，PHP就把最常用的函数集成到了系统。这样就可以通过阅读相关文档来了解到这些函数以及使用方法，然后就可以直接调用这些函数了。这为程序编写节省了很多可观的时间。前面我们已经学习过一些系统函数，本章我们接着学习一些常用的系统函数。

9.1 时间和日期处理函数

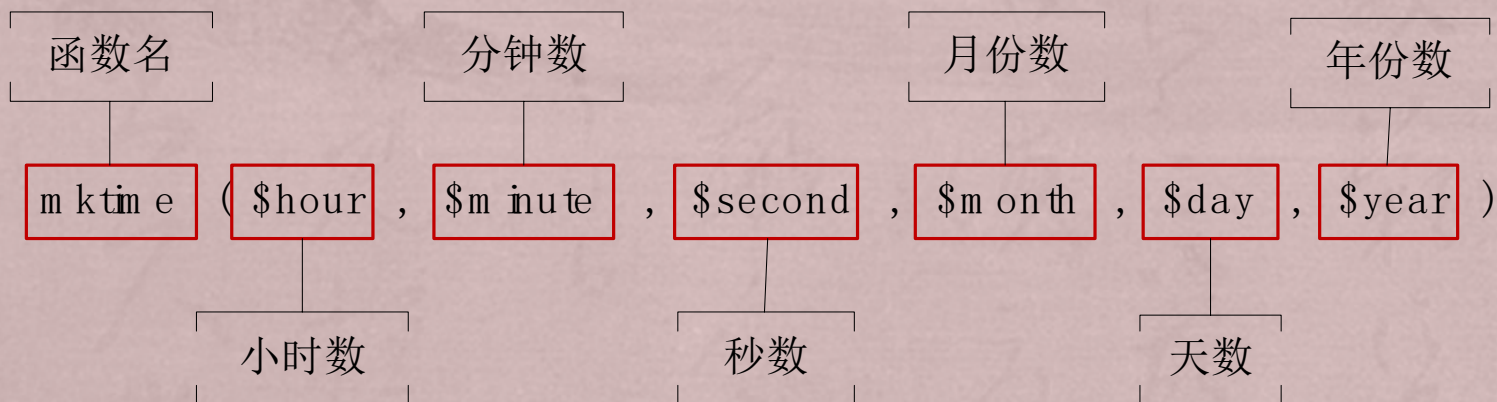
+ 时间是生活中必不可少的，在PHP中也是一样的，时间有时候要存储起来记录某件事情，或者要在浏览器中展示出来，这就需要使用到一些时间和日期的处理函数。本节我们就来常用的时间和日期处理函数。

9.1.1 UNIX时间戳

- + Unix时间戳，或称Unix时间，是一种时间表示方式。它定义为从格林威治时间1970年01月01日00时00分00秒起至现在的总秒数。Unix时间戳不仅被使用在Unix系统，也在许多其他操作系统中被广泛采用。
- + 目前相当一部分操作系统使用32位二进制数字表示时间。此类系统的Unix时间戳最多可以使用到格林威治时间2038年01月19日03时14分07秒（二进制：01111111 11111111 11111111 11111111）。其后一秒，二进制数字会变为10000000 00000000 00000000 00000000，发生溢出错误，造成系统将时间误解为1901年12月13日20时45分52秒。这很可能会引起软件故障，甚至是系统瘫痪。使用64位二进制数字表示时间的系统，最多可以使用到格林威治时间292277026596年12月04日15时30分08秒，则基本不会遇到这类溢出问题。

1.将日期和时间转换为UNIX时间戳

- + 使用mktime()可以将时间转换为UNIX时间戳，它的语法如图所示。
- + 在mktime()中所有的参数都是可以从右向左省略的，被省略的时间会被设置为本地的日期和时间，如果输入的参数非法，函数会返回FALSE。



1.将日期和时间转换为UNIX时间戳

- + (1)演示使用mktime()取得本地时间戳和将时间格式为UNIX时间戳。
- + 在使用mktime()的时候有一个需要注意它会自动计算超出范围的输入的正确值，并不会发生错误。
- + (2)演示mktime()函数会自动计算越界的参数和输入非法参数程序会出现的情况。

2. 计算时间差

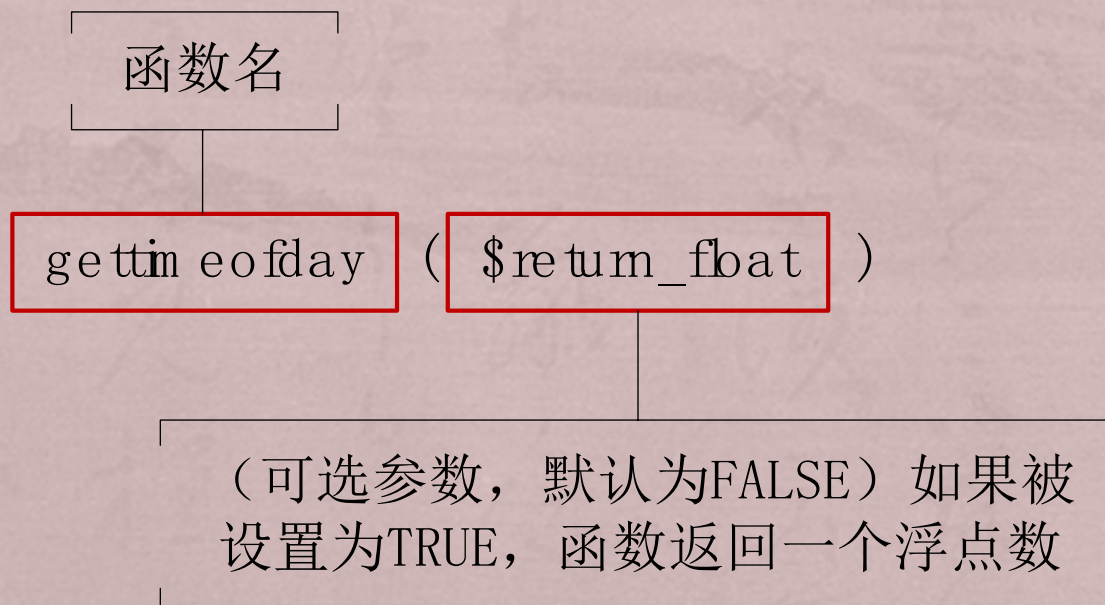
- + UNIX 时间戳的优势就在于通用性，不会因为国家或者地区的不同而出现不兼容的情况。而由于它是表示秒数，虽然看起来很头疼但是，用于计算时间差就非常方便。
- + (1) 演示一个人从出生到现在一共过了多长时间。

9.1.2 获取时间

- + PHP提供了多种获取日期和时间的函数，除了我们前面认识的mktime()和time()之外。还可以使用getdate()获取当前的时间信息，使用gettimeofday()获取某一天中的具体时间。

1.GETTIMEOFDAY()

- + gettimeofday() 可以获取某一天中的具体时间，它可以接受一个可选参数，语法如图所示。



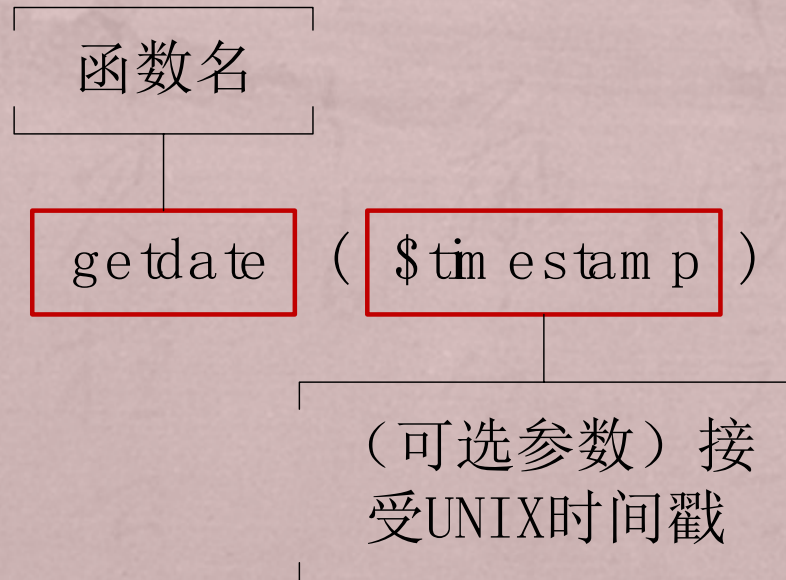
1.GETTIMEOFDAY()

- + gettimeofday()默认返回的是一个包含四个元素的数组，如果可选参数被设置为TRUE，函数会返回一个浮点数。gettimeofday()返回的数组元素名及对应的描述如表所示。
- + (1)演示使用gettimeofday()获取当前时间并输出返回的数组信息。

数组元素名	描述
sec	自Unix纪元起的秒数(UNIX时间戳)
usec	此时微秒数，1000000微秒=1秒
minuteswest	与格林威治时间相差的分钟数
dstime	夏令时修正类型

2.GETDATE()

- + getdate()的作用同样是取得日期和时间信息，不同的是它可以返回更加多的项目，而且不仅可以取得当前时间信息，而且可以取得一个UNIX时间戳的时间信息，它的语法如图所示。



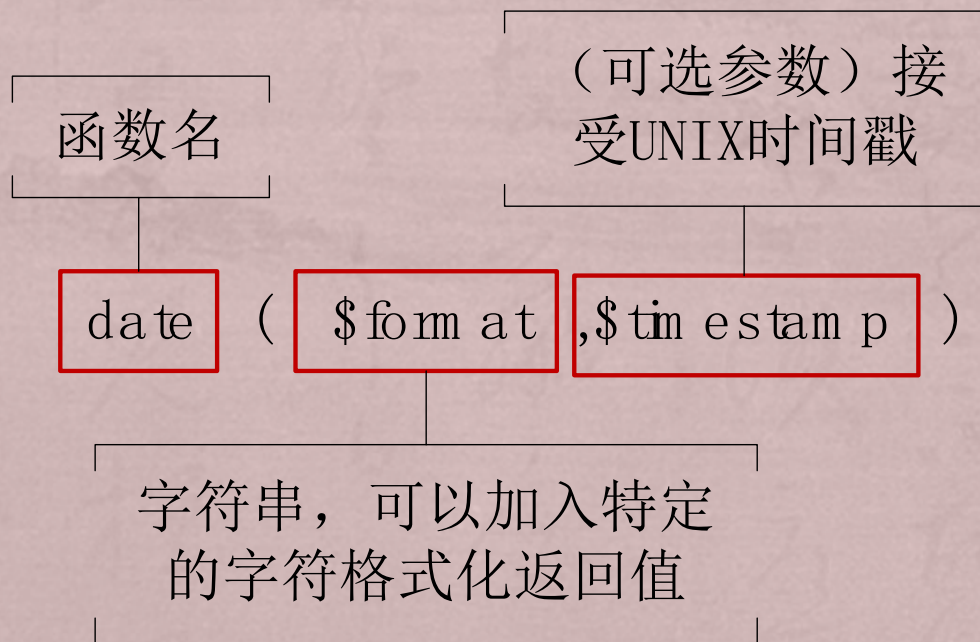
2.GETDATE()

- + getdate()在没有参数的情况下返回当前时间信息的数组，在接受参数的情况下返回UNIX时间戳的详细信息数组。数组的元素名称及对应描述如表所示。
- + (1)演示getdate()获取当前时间日期返回的数组信息和被赋值UNIX时间戳时返回的数组信息。
- + (2)演示一个使用getdate()取得时间信息并以符合我们习惯的时间格式输出。

元素名称	描述
seconds	秒的数字表示
minutes	分钟的数字表示
hours	小时的数字表示
mday	月份中第几天的数字表示
wday	星期中第几天的数字表示
mon	月份的数字表示
year	4位数字表示的完整年份
yday	一年中第几天的数字表示
weekday	星期几的完整文本表示
month	月份的完整文本表示
0	自从Unix纪元开始至今的秒数

9.1.3 日期和时间格式输出

- + 当日期和时间保存在计算机的时候可以使用UNIX时间戳的形式，但是UNIX时间戳的可读性很差。虽然我们可以通过一些计算来使的结果更加容易阅读，但这不是最简便的方法。在PHP中我们可以使用date()非常容易地格式化一个日期时间信息，它的语法如图所示。



9.1.3 日期和时间格式输出

+ date()将UNIX时间戳的
按照给定格式字符串的
形式返回。如果没有本
给出时间戳则使用本
地当前时间。date()的
格式化是使用特定的
字符来约束的, 这些
字符特定于所
表中的就列出了所有
表定的格式化了符, 指
者不就要见这大一个
表就害怕了。这是完
全没有必要的。在于
个表有目的在让读
者有个参考, 常用
者只有不多几个, 在
使用后就记住。

特定字符	功能描述
日	
d	月份中的第几天, 有前导零的2位数字
D	星期中的第几天, 文本表示, 3个字母
j	月份中的第几天, 没有前导零
l	星期几, 完整的文本格式
N	ISO-8601格式数字表示的星期中的第几天
S	每月天数后面的英文后缀, 2个字符
w	星期中的第几天, 数字表示
z	年份中的第几天
星期	
W	ISO-8601格式年份中的第几周, 每周从星期一开始
月	
F	月份, 完整的文本格式
m	数字表示的月份, 有前导零
M	三个字母缩写表示的月份
n	数字表示的月份, 没有前导零
t	给定月份所应有的天数
年	
L	是否为闰年, 闰年为1, 否则为0
o	ISO-8601格式年份数字。这和Y的值相同, 如果星期数 (W) 属于前一年或下一年, 则用那一年
Y	4位数字完整表示的年份
y	2位数字表示的年份
时间	
a	小写的上午和下午值
A	大写的上午和下午值
B	Swatch Internet 标准时
g	小时, 12小时格式, 没有前导零
G	小时, 24小时格式, 没有前导零
h	小时, 12小时格式, 有前导零
H	小时, 24小时格式, 有前导零
i	有前导零的分钟数
s	秒数, 有前导零

9.1.3 日期和时间格式输出

- + (1) 演示使用 `date()` 格式化当前时间并输出。
- + (2) 演示使用 `date()` 判断年份是不是闰年并使用判断语句输出提示。
- + (3) 使用 `date()` 和 `mktime()` 得到未来的日期。

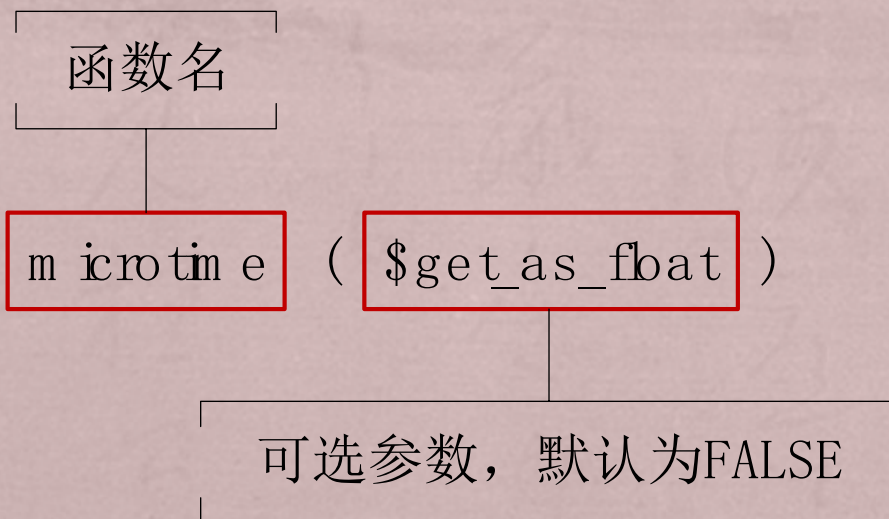
9.1.4 使用

DATE_DEFAULT_TIMEZONE_SET() 设置默认时区

- + 我们知道不同的国家和地区都有自己的本地时间，PHP的默认时区是通用协调时即UTC时间。而北京正好位于时区的东八区，因此时间领先于UTC时间8小时。因此我们使用date()等函数取得的日期会和北京时间相差8小时。这时我们就可以使用date_default_timezone_set()来更改默认时区。
- + date_default_timezone_set()只可以接受一个代表默认时区的字符串代码，字符串“PRC”表示的就是中华人民共和国。
- + (1)演示使用date_default_timezone_set()设置默认时区。

9.1.5 使用MICROTIME()计算程序执行时间

- + 在程序世界里，通常判断一个程序的级别可以从执行相同的操作花费的时间来比较。而这些差别都是微秒级的，因此检测出这微小的差别我们就要学习一个新的日期时间函数microtime()，它的语法如图所示。
- + microtime()默认返回一个由微秒数和UNIX时间戳组成的数组，如果可选参数被设置为TRUE，函数将返回一个UNIX时间戳的浮点数。



9.1.5 使用MICROTIME()计算程序执行时间

- + (1)演示microtime()在无参数和参数被设置为“TRUE”时的输出。
- + (2)演示使用microtime()计算程序执行时间。

9.2 数学函数

+ 数学计算在程序的运用也是非常多的，例如找出一组数字中的最大和最小的数、正弦、正切、余弦以及数字的四舍五入、产生随机数这些操作，这些操作PHP都提供了相应的函数，本节中我们就来学习几个常用的数学函数。

9.2.1 进制间的转换

+ 我们知道计算机使用的是进制数，有的时候我们需要把一个数转换为其他进制的数，这就需要用到转换进制的函数，常用的进制间转换的函数及功能描述如表所示。

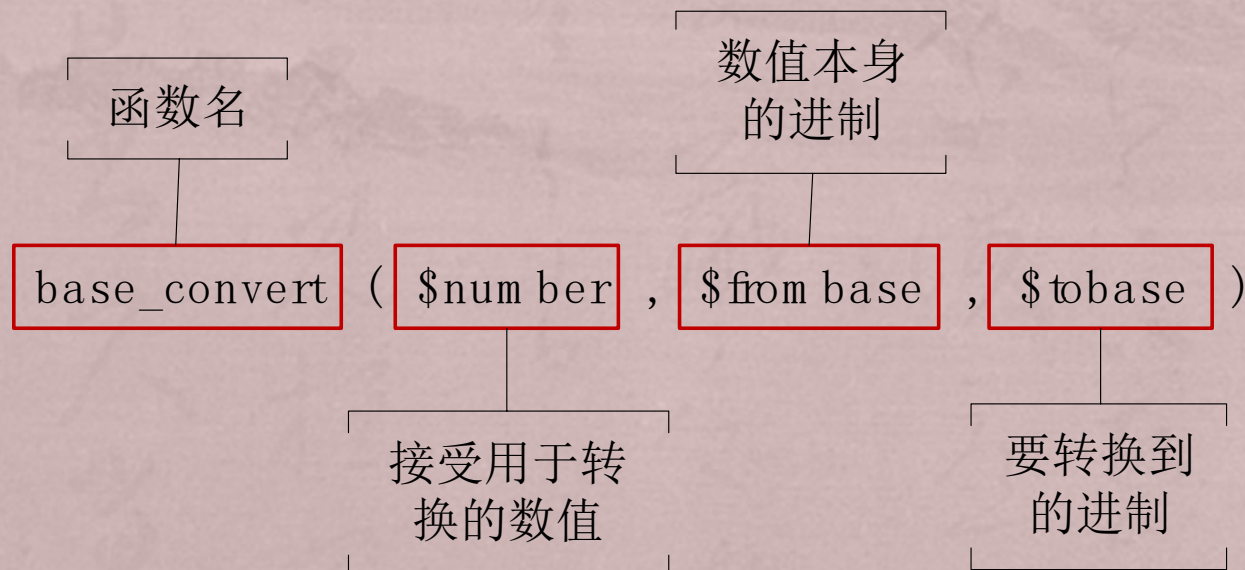
函数名	功能描述
decbin()	将十进制数转换为二进制数
decoct()	将十进制数转换为八进制数
dechex()	将十进制数转换为十六进制数
bindec()	将二进制数转换为十进制数
hexdec()	将十六进制数转换为十进制数
base_convert()	任意进制间数字的转换

9.2.1 进制间的转换

- + (1) 演示 `decbin()`、`decoct()`、`dechex()`、`bindec()`、`hexdec()` 的使用方法以及输出结果。

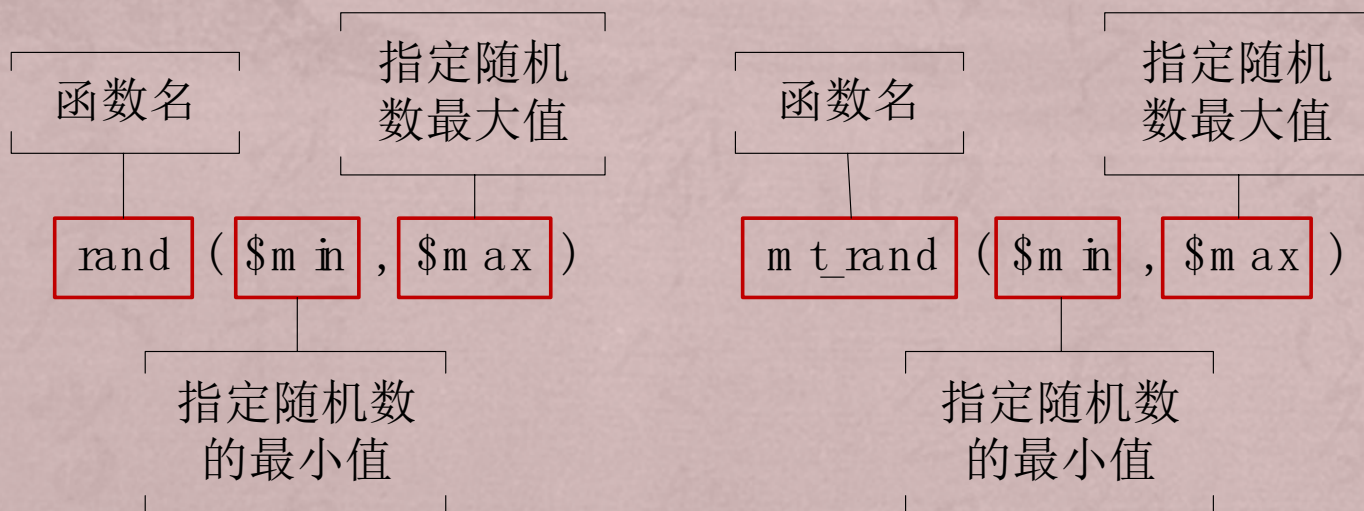
9.2.1 进制间的转换

- + 函数 `base_convert()` 可以进行任意进制间数值的转换，它的语法如图所示。
- + (1) 演示使用 `base_convert()` 来转换对应的数值。



9.2.2 生成随机数

- + 随机数在实际应用也是一个不可忽略的部分，就像我们最常见的验证码就是使用的随机数，还有概率类游戏，例如丢骰子游戏也需要使用到随机数。在PHP中我们可以使用rand()和mt_rand()生产一个随机整数。它们的语法很类似，可以没有参数或者接受两个参数，如图所示。
- + rand()和mt_rand()的最主要的不同点是它们使用的产生随机数的算法是不同，可以生成的随机数范围不同。我们可以使用getrandmax()和mt_getrandmax()取得对应函数可以生成的最大随机数。



9.2.2 生成随机数

- + (1) 演示使用 `rand()` 和 `me_rand()` 生成随机数，以及使用 `getrandmax()` 和 `mt_getrandmax()` 获取函数可以生成随机数的最大值。
- + (2) 利用生成的随机数判断该做什么事。

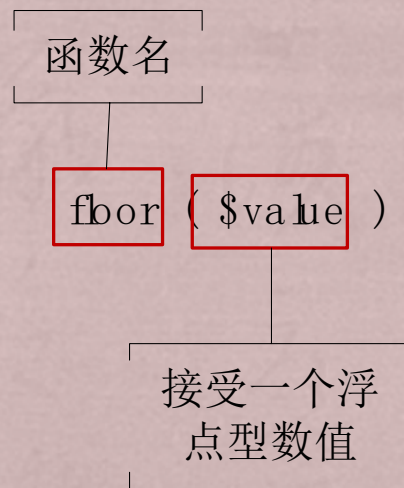
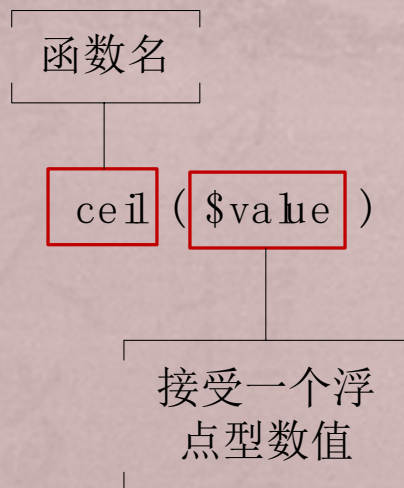
9.2.3 近似数处理函数

+ 近似数处理函数通常就是执行四舍五入或者舍去法求整的操作，PHP中常用的近似数处理函数如表所示。

函数名	功能描述
abs()	取得数值的绝对值
ceil()	进一法取整
floor()	舍去法取整
fmod()	返回除法的浮点数余数
round()	对浮点数四舍五入法取整

1.CEIL()、FLOOR()

- + ceil()和floor()的语法比较类似，如图所示。
- + ceil()会返回一个不小于传入参数的浮点型数值，floor()会返回一个不大于传入参数的浮点型数值。也就是说ceil()的参数只要有小数位就会给整数位加1，floor()会把小数位舍去。

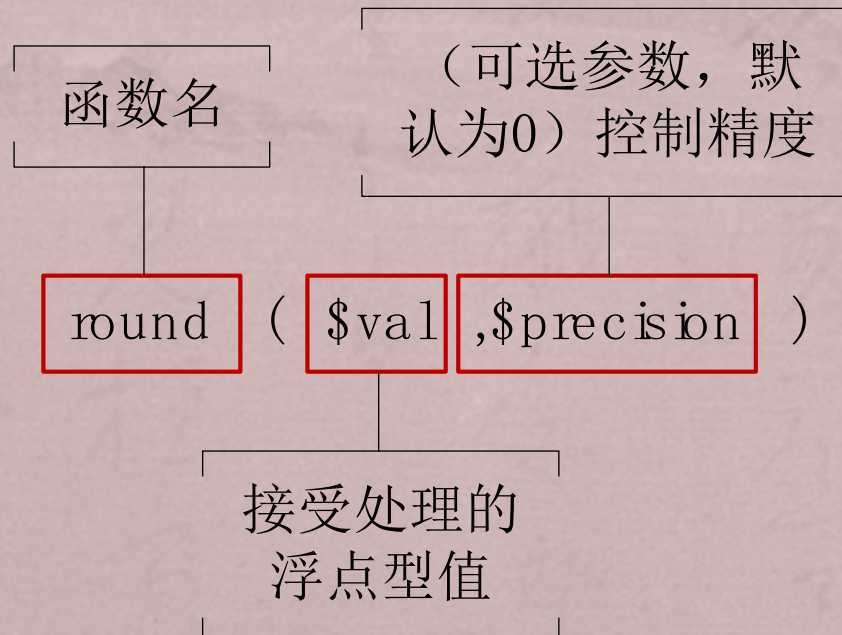


1.CEIL()、FLOOR()

+ (1)演示ceil()和floor()的用法以及返回的值。

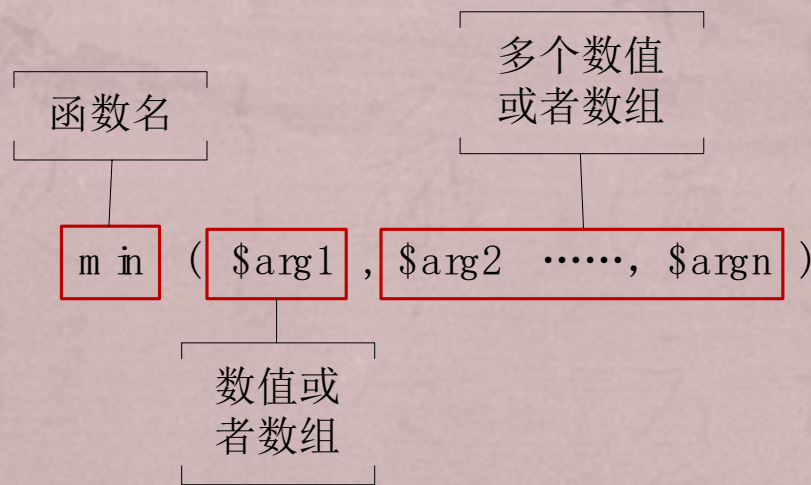
2.ROUND()

- + round() 使用四舍五入法对浮点数取整，语法如图所示。
- + round() 接受一个浮点型数值，返回一个由\$precision控制的精度的浮点型数值，\$precision可以为正数负数或者0。
- + (1) 演示round()的各种使用形式对结果的影响。



9.2.4 查找最大值和最小值

- + 在PHP中找出最大值和最小值，可以自己创建一个函数，我们也做过比较两个数大小的函数。在系统中提供了min()和max()来判断多个参数的最大值和最小值，它们的语法类似，都可以接受一个或者多个参数返回其中最大或者最小的值，min()如图所示。
- + min()会返回参数中最小的值，如果仅有一个参数且为数组，min()返回该数组中最小的值。如果给出了两个或更多参数，min()会返回这些值中最小的一个。

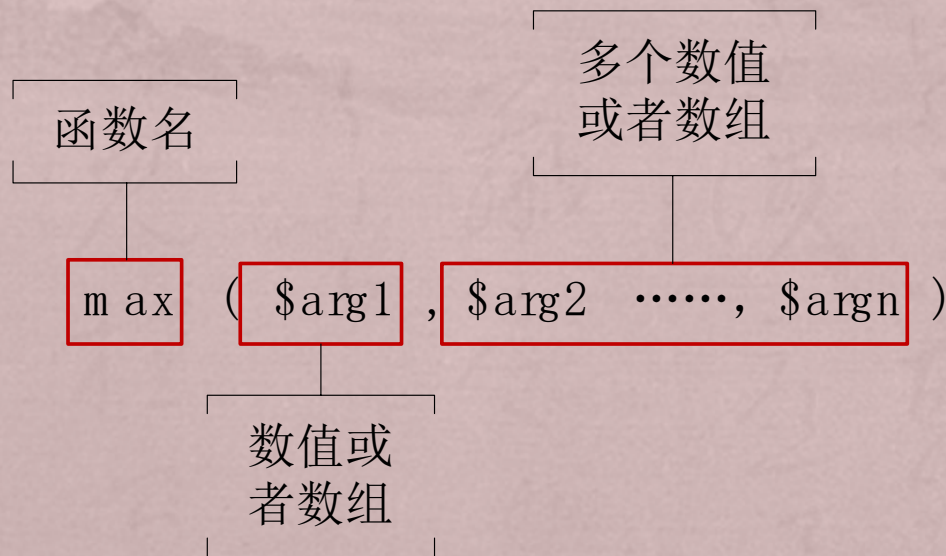


9.2.4 查找最大值和最小值

- + (1)演示min()的使用方法以及比较特殊的值会返回的结果。
- + 结合上面的代码和运行结果我们总结一下min()的运行特点：
- + 如果参数均为数字，则返回最小的数字。
- + 如果参数为一个数组，则返回数组中最小元素的值。
- + 如果参数为一个数组和一个数值比较，则返回数值。
- + 如果参数为多个参数个数相等的数组，则从左向右依次比较各数组对应的元素，返回比较元素最小的数组。
- + 如果参数为两个或者多个长度不同的数组，返回元素个数少的数组。
- + 字符串参数会被认为是0，如果所有参数值的相同返回靠左参数。

9.2.4 查找最大值和最小值

- + `max()`语法如图所示。
- + `max()`会返回参数中最大的值，如果仅有一个参数且为数组，`max()`返回该数组中最大的值。如果给出了两个或更多参数，`max()`会返回这些值中最大的一个。
- + `max()`也有类似`min()`的运行特点：
- + 如果参数均为数字，则返回最大的数字。
- + 如果参数为一个数组，则返回数组中最大元素的值。
- + 如果参数为一个数组和一个数值比较，则返回数组。
- + 如果参数为多个参数个数相等的数组，则从左向右依次比较各数组对应的元素，返回比较元素最大的数组。
- + 如果参数为两个或者多个长度不同的数组，返回元素个数多的数组。
- + 字符串参数会被认为是0，如果所有参数值的相同则返回靠左的参数。
- + (1)演示`max()`的使用方法以及比较特殊的值会返回的结果。



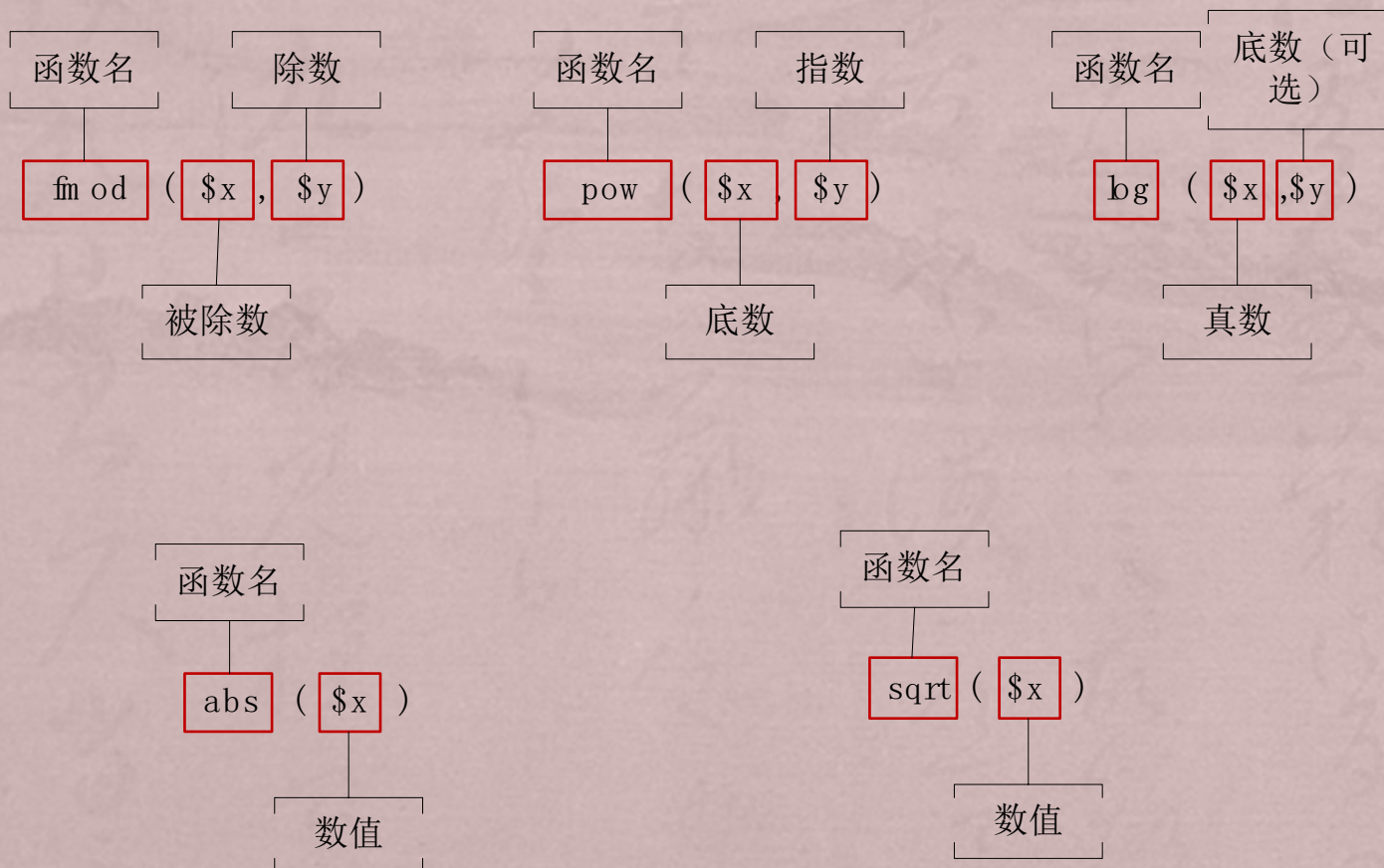
9.2.5 数学计算

+ 在数学中我们常用的计算有求绝对值、求平方根、指数、对数这些计算，PHP都为我们提供了相应的函数，常用的如表所示。

函数名	功能描述
abs()	求绝对值
sqrt()	求平方根
pow()	指数表达式
fmod()	返回除法的浮点余数
log()	自然对数

9.2.5 数学计算

- + 它们的语法如图所示。
- + (1) 演示表中函数的使用及其输出。



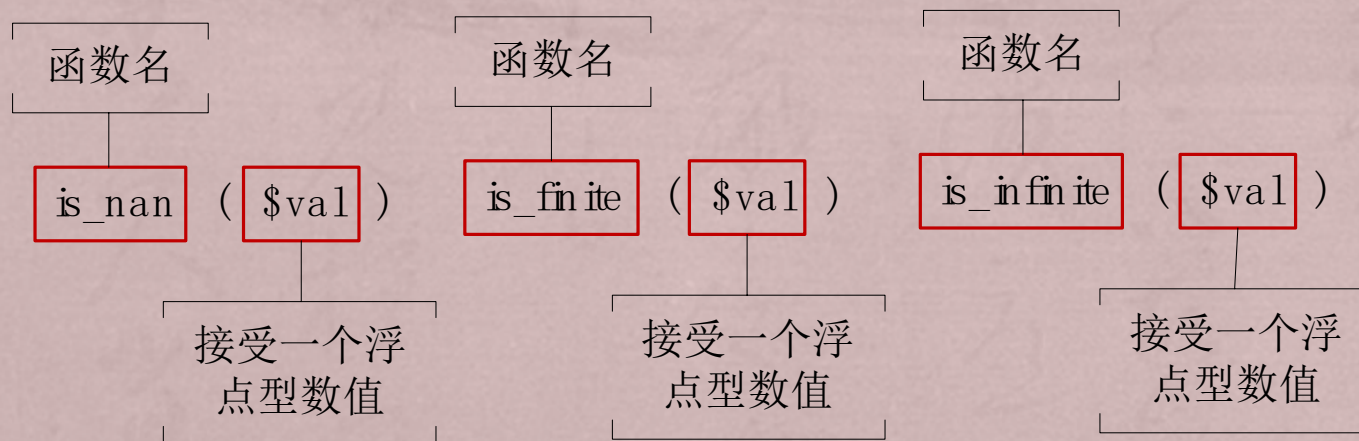
9.2.6 数值判断

- + 在PHP的数学函数中使用is_nan()、is_finite()、is_infinite()来判断数值，它们的描述如表所示。

函数名	功能描述
is_nan()	判断参数是否为一个非数值
is_finite()	判断参数是否为一个有限值
is_infinite()	判断参数是否为一个无限值

9.2.6 数值判断

- + 它们的语法都类似，都接受一个浮点类型的参数，如图所示。
- + (1) 演示联合使用 `is_nan()`、`is_finite()`、`is_infinite()` 判断数值。
- + 由于 `pow(1000,10000)` 的值是一个接近无限大的值因此被显示为“INF”，类似的“NAN”是一个常量，表示不是一个数字（not a number）。



9.3 小结

+ 本章主要学习了系统函数中的时间日期的处理函数和数学相关的处理函数。当然把所有的函数都讲解一遍显然是不现实的，本章中我们挑出最最常用的一些函数做了讲解。虽然使用非常简单，但是需要注意的细节如果不给读者讲解是很容易把读者引入错误中去的。本章希望读者做到的就是在可以使用到这些函数的地方应该想到使用它们而不是去自己写一个函数。