

Math Functions

Disclaimer: вы смотрите просто запись лекции,
это НЕ специально подготовленный видеокурс!



Intro

While PHP was never considered to be a “math operations language”, it still has a lot of math functions, and a lot of operations require those functions.

Full list: <https://www.php.net/manual/en/ref.math.php>

Rounding numbers

To round a number in PHP we may use **round**, **ceil**, **floor** functions. Unlike in many other languages, **round** function in PHP is really powerful.

```
<?php

echo round(3.4) . "\n"; // 3
echo round(3.5) . "\n"; // 4
echo round(3.6) . "\n"; // 4
echo round(3.6, 0) . "\n"; // 4
echo round(5.045, 2) . "\n"; // 5.05
echo round(5.055, 2) . "\n"; // 5.06
echo round(345, -2) . "\n"; // 300
echo round(345, -3) . "\n"; // 0
echo round(678, -2) . "\n"; // 700
echo round(678, -3) . "\n"; // 1000

echo ceil(4.3) . "\n"; // 5
echo ceil(9.999) . "\n"; // 10
echo ceil(-3.14) . "\n"; // -3

echo floor(4.3) . "\n"; // 4
echo floor(9.999) . "\n"; // 9
echo floor(-3.14) . "\n"; // -4
```

Generating random numbers

And again: unlike many other languages PHP provides simple and convenient way to generate a random number within defined range.

```
<?php

// Get random integer
echo mt_rand(1, 999) . "\n"; // e.g., 743

// Get random float
function getRandomFloat(float $min, float $max) : float {
    if ($min >= $max) {
        throw new Exception("The minimum border should be less than a maximum border.");
    }
    return ($min + lcg_value() * (abs($max - $min)));
}

echo getRandomFloat(0.5, 0.7) . "\n"; // e.g., 0.51656109828938
```

Changing base

And again: unlike many other languages PHP provides simple and convenient way to change a base of a number.

```
<?php

echo dechex(255) . "\n"; // ff
echo hexdec('ff') . "\n"; // 255

echo decbin(3) . "\n"; // 11
echo bindec('11') . "\n"; // 3

echo decoct(8) . "\n"; // 10
echo octdec('10') . "\n"; // 8

echo base_convert('AB', 16, 2) . "\n"; // 10101011
```

Detecting minimal and maximal values

And again: unlike many other languages PHP provides simple and convenient way to detect a min/max value.

```
<?php

$data = array(5, 3, 9, 2);

echo max($data) . "\n"; // 9
echo max(9, 12, 6, 8) . "\n"; // 12

echo min($data) . "\n"; // 2
echo min(9, 12, 6, 8) . "\n"; // 6

$data = array('Test', 'ABC', 'ZZZ', 'Just a string');

echo max($data) . "\n"; // ZZZ
echo min($data) . "\n"; // ABC
```

Other potentially useful functions

These functions may also be useful in some arithmetical operations. Just a quick overview:

```
<?php

echo abs(-4.2) . "\n"; // 4.2
echo abs(5) . "\n";    // 5
echo abs(-5) . "\n";   // 5

echo pi() . "\n";; // 3.1415926535898
echo M_PI . "\n";; // 3.1415926535898

echo pow(2, 8) . "\n";    // 256
echo pow(-1, 20) . "\n";  // 1
echo pow(0, 0) . "\n";    // 1
echo pow(10, -1) . "\n";  // 0.1
echo pow(-1, 5.5) . "\n"; // NAN
```

Detecting incorrect operation results

While dealing with numbers there is always a potential risk to get overflow or “not a number” at all.

```
<?php

$a = 1;
$b = 10;
$c = 0;
$d = -5;

if (is_finite($a + $b)) {
    echo ($a + $b) . "\n";
}

if (is_infinite(log($c))) {
    echo "Sorry, it is impossible to calculate log(" . $c . ")\n";
} else {
    echo log($c) . "\n";
}

if (is_nan(sqrt($d))) {
    echo "Sorry, it is impossible to calculate sqrt(" . $d . ") in real numbers.\n";
} else {
    echo sqrt($d) . "\n";
}
```


Math Functions

Disclaimer: вы смотрите просто запись лекции,
это НЕ специально подготовленный видеокурс!

