



PHP 5.6.x

to

PHP 7.0.x





## New Features:

### Scalar type declarations

```
function sumOfInts(int ...$ints)
{
    return array_sum($ints);
}
```



## New Features:

### Return type declarations

```
function arraysSum(array ...$arrays): array
{
    return array_map(function(array $array): int
    {
        return array_sum($array);
    }, $arrays);
}
```



## New Features:

### Null coalescing operator (??)

```
$username = $_GET['user'] ?? 'nobody';  
// This is equivalent to:  
$username = isset($_GET['user']) ? $_GET['user'] : '  
nobody';
```



## New Features: Spaceship operator ( $\Leftrightarrow$ )

```
echo 1 <=> 1; // 0  
echo 1 <=> 2; // -1  
echo 2 <=> 1; // 1
```



## New Features:

### Constant arrays using define()

```
define('ANIMALS', [  
    'dog',  
    'cat',  
    'bird'  
]);
```



## New Features: Anonymous classes

```
$app = new Application;  
$app->setLogger(new class implements Logger {  
    public function log(string $msg) {  
        echo $msg;  
    }  
});
```





## New Features:

### Unicode codepoint escape syntax

```
echo "\u{aa}"; //a  
echo "\u{0000aa}"; //a  
echo "\u{9999}"; //香
```





## New Features: Closure::call()

```
class A {private $x = 1;}  
$getX = function() {return $this->x;};  
echo $getX->call(new A);
```



## New Features: Filtered unserialize()

```
unserialize ( string $str [, array $options ] )  
: mixed
```



## New Features: Group *use* declarations

```
use some\namespace\{ClassA, ClassB, ClassC as C};  
use function some\namespace\{fn_a, fn_b, fn_c};  
use const some\namespace\{ConstA, ConstB, ConstC};
```



## New Features: Generator Return Expressions

```
$gen = (function() {  
    yield 1;  
    yield 2;  
  
    return 3;  
})();
```

```
foreach ($gen as $val) {  
    echo $val, PHP_EOL;  
}  
  
echo $gen-  
>getReturn(), PHP_EOL;
```



## New Features: Generator delegation

```
function gen()  
{  
    yield 1;  
    yield 2;  
    yield from gen2(); //Where gen2  
                        yields 3 and 4  
}
```



New Features:

Integer division with `intdiv()`

```
var_dump(intdiv(10, 3));
```



## New Features: Session options

```
session_start([  
    'cache_limiter' => 'private',  
    'read_and_close' => true,  
]);
```





## New Features:

`preg_replace_callback_array()`

```
preg_replace_callback_array ( array  
$patterns_and_callbacks , mixed $subject [, int $limit =  
-1 [, int &$count ]] ) : mixed
```

# Example:

## preg\_replace\_callback\_array()

```
<?php
$subject = 'Aaaaaa Bbb';

preg_replace_callback_array(
    [
        '~[a]+~i' => function ($match) {
            echo strlen($match[0]), ' matches for "a" found', PHP_EOL;
        },
        '~[b]+~i' => function ($match) {
            echo strlen($match[0]), ' matches for "b" found', PHP_EOL;
        }
    ],
    $subject
);
?>
```



## New Features: CSPRNG Functions

```
random_bytes ( int $length ) : string  
random_int ( int $min , int $max ) : int
```



thanks!

Any questions?

Georgi Georgiev