

PHP

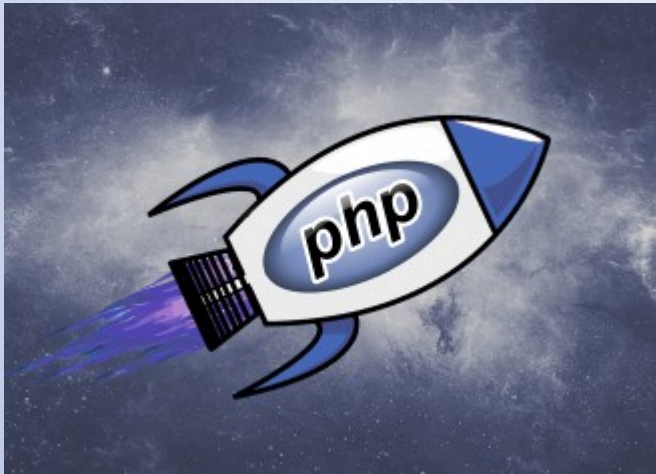


What's new in Php 7 ?
PHP 5.6 vs PHP 7



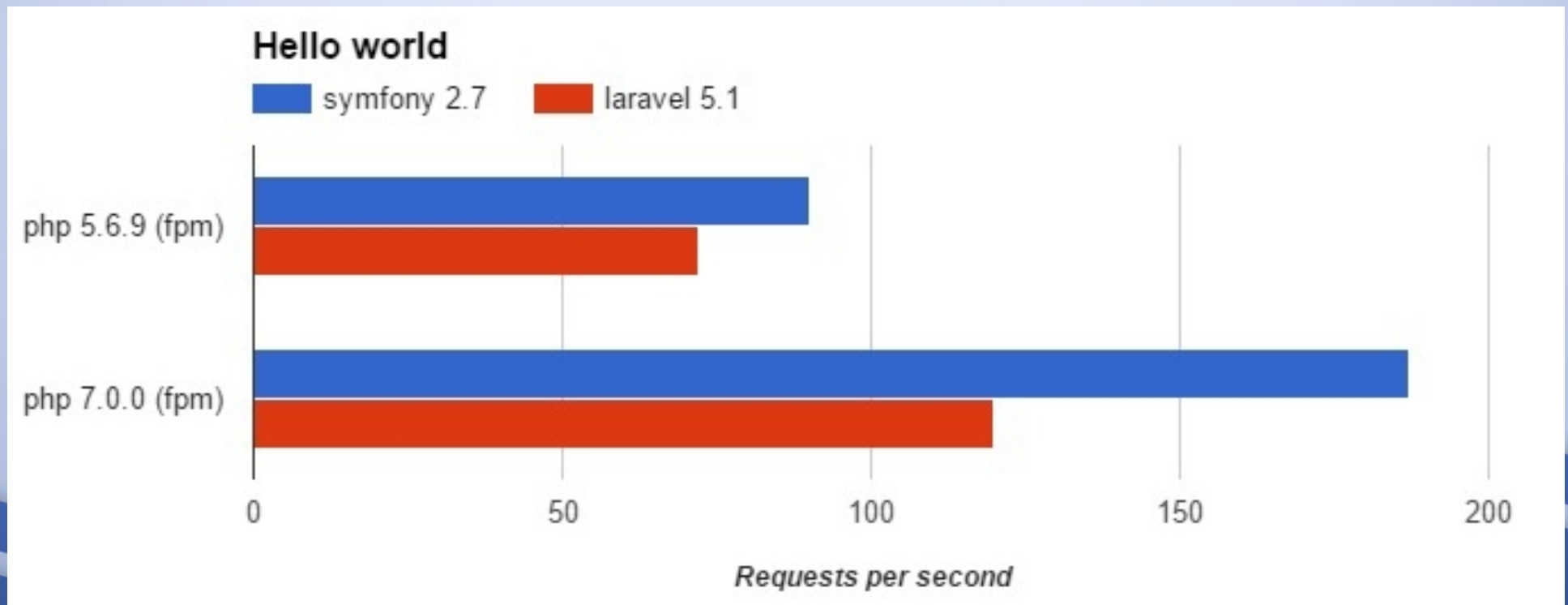
PERFORMANCE

- Php 7 is two times faster than php 5.6
- Php 7 uses significantly less memory than php 5.6



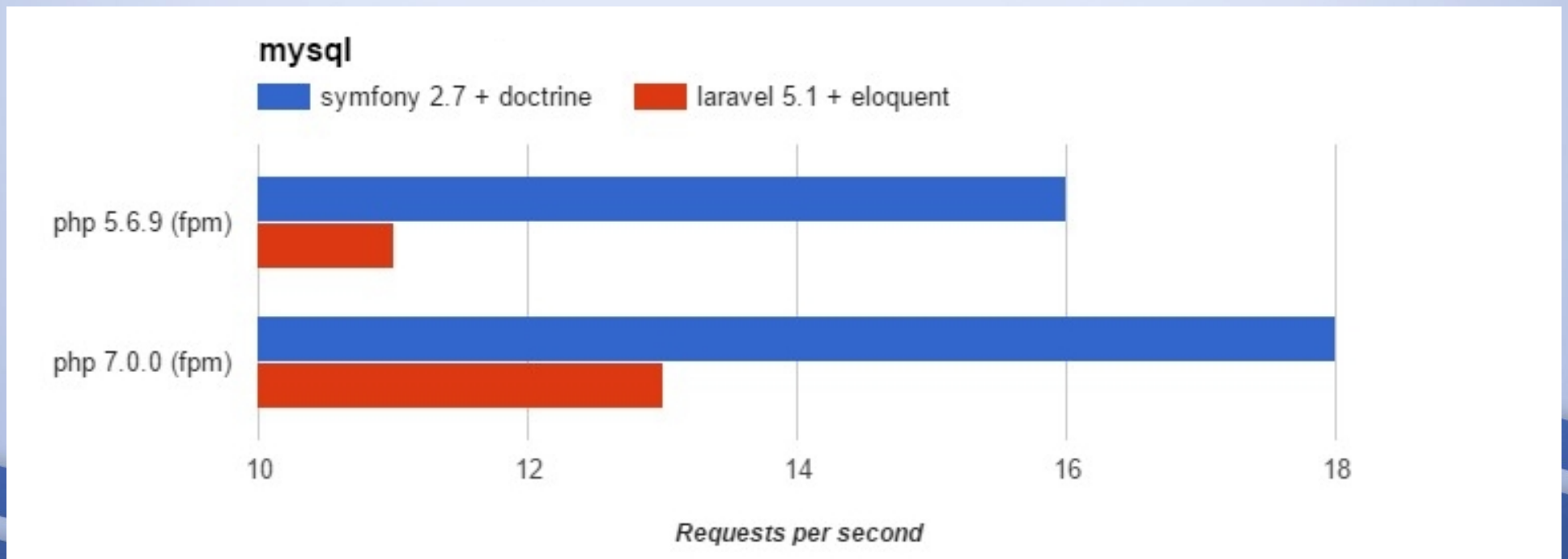
PERFORMANCE

- Hello world – one action, one controller and one view. It was without any additional optimization.

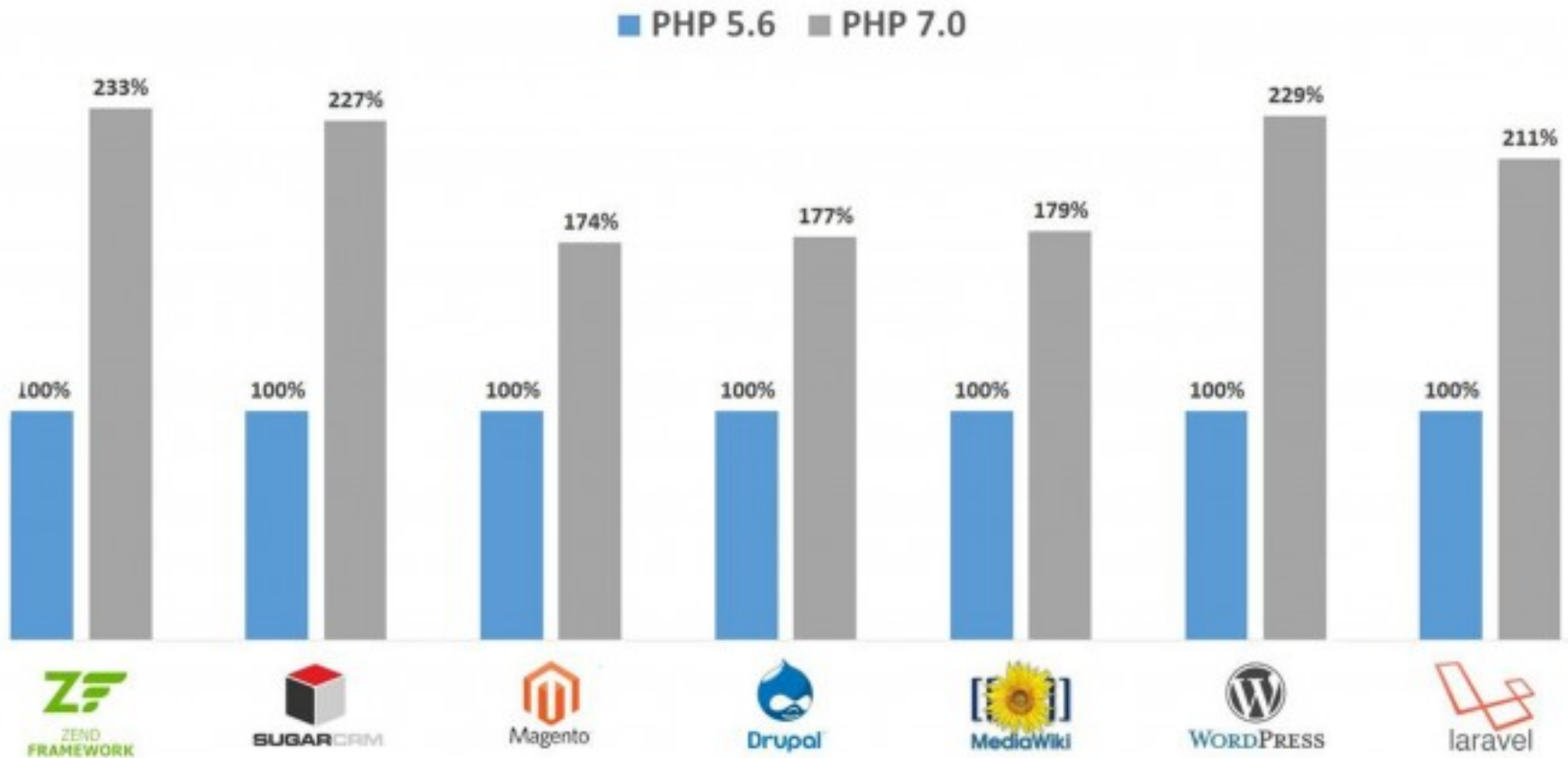


PERFORMANCE

- Mysql - one „INSERT INTO” using a standard tool for handing databases supplied with frameworks.



PERFORMANCE



New features

- Scalar type declarations;
- Return type declarations;
- Null coalescing operator;
- Spaceship operator;
- Constant arrays using `define()`;
- Anonymous classes;
- Filtered `unserialize()`;
- `IntlChar`;
- Expectations;
- Group *use* declarations;
- Generator Return Expressions;
- Generator delegation;
- Integer division with `intdiv()`;
- Session options;

Scalar type declarations

Scalar type declarations come in two flavours: coercive (default) and strict. The following types for parameters can now be enforced (either coercively or strictly): strings (string), integers (int), floating-point numbers (float), and booleans (bool). They augment the other types introduced in PHP 5: class names, interfaces, array and callable.

Scalar type declarations

```
//declare (strict_types=1);
function sumOld($a,$b){
    return $a+$b;
}

function sumNew(int $a, int $b) {
    return $a+$b;
}

echo "1: ".sumOld("10 abc",5)."<br>";
echo "2: ".sumNew("10 abc",5)."<br>";
echo "3: ".sumNew(10,5). " <br>";
echo "4: ".sumNew("10",5). " <br>";
echo "5: ".sumOld("10",5)."<br>";
```

1: 15

Notice: A non well formed numeric value encountered in

2: 15

3: 15

4: 15

5: 15

Scalar type declarations

```
declare (strict_types=1);
```

```
echo "1: ".sumOld("10 abc",5). "<br>";  
echo "2: ".sumNew("10 abc",5). "<br>";  
echo "3: ".sumNew(10,5). "<br>";  
echo "4: ".sumNew("10",5). "<br>";  
echo "5: ".sumOld("10",5). "<br>";
```

1: 15

Fatal error: Uncaught TypeError: Argument 1 passed to sumNew() must be of the type integer, string given, called in /var/www/html/php7prezentacja/index.php on line 12 and defined in /var/www/html/php7prezentacja/index.php:7 Stack trace: #0 /var/www/html/php7prezentacja/index.php(12): sumNew('10 abc', 5) #1 {main} thrown in /var/www

Scalar type declarations

```
declare(strict_types=1);  
//echo "1: ".sumOld("10 abc",5)."<br>";  
//echo "2: ".sumNew("10 abc",5)."<br>";  
echo "3: ".sumNew(10,5)."<br>";  
echo "4: ".sumNew("10",5)."<br>";  
echo "5: ".sumOld("10",5)."<br>";
```

1: 15

3: 15

Fatal error: Uncaught TypeError: Argument 1 passed to sumNew() must be of the type integer, string given, called in /var/www/html/php7prezentacja/index.php on line 14 and defined in /var/www/html/php7prezentacja/index.php:7 Stack trace: #0 /var/www/html/php7prezentacja/index.php(14): sumNew('10', 5) #1 {main} thrown in /var/www

Return type declarations

PHP 7 adds support for return type declarations. Similarly to argument type declarations, return type declarations specify the type of the value that will be returned from a function. The same types are available for return type declarations as are available for argument type declarations.

Return type declarations

```
public function getName():String {  
    return $this->name;  
}  
  
public function getAge():int{  
    return $this->age;  
}
```

```
$userNew=new ClassTest;  
$userNew->setName( "Radek" );  
function setObject(ClassTest $object): ClassTest{  
    $object->setLikeIt(false);  
    $object->setInWollet(12.56);  
  
    return $object;  
}  
  
$aaa=setObject($userNew);  
  
var_dump($aaa);
```

```
object(ClassTest)#1 (4) { ["name":"ClassTest:private"]=> string(5) "Radek"  
["age":"ClassTest:private"]=> int(22) ["likeIt":"ClassTest:private"]=> bool(false)  
["cashInYourWallet":"ClassTest:private"]=> float(12.56) }
```

Return type declarations

```
$userNew=new ClassTest;
$userNew->setName("Radek");
function setObject(ClassTest $object): ClassTest{
    $object->setLikeIt(false);
    $object->setInWollet(12.56);

    return "I";
}

$aaa=setObject($userNew);
|
var_dump($aaa);
```

Fatal error: Uncaught TypeError: Return value of setObject() must be an instance of ClassTest, string returned in /var/www/html/php7prezentacja/testCla.php:20 Stack trace: #0 /var/www/html/php7prezentacja/testCla.php(23): setObject(Object(ClassTest)) #1 {main} thrown in /var/www/html/php7prezentacja/testCla.php on line 20

Changed functions

- `debug_zval_dump()` now prints "int" instead of "long", and "float" instead of "double"
- `dirname()` now optionally takes a second parameter, depth, to get the name of the directory depth levels up from the current directory.
- `getrusage()` is now supported on Windows.
- `mktime()` and `gmmktime()` functions no longer accept `is_dst` parameter.
- `preg_replace()` function no longer supports "\e" (PREG_REPLACE_EVAL). `preg_replace_callback()` should be used instead.

Changed functions

- `setlocale()` function no longer accepts category passed as string. `LC_*` constants must be used instead.
- `exec()`, `system()` and `passthru()` functions have NULL byte protection now.
- `shmop_open()` now returns a resource instead of an int, which has to be passed to `shmop_size()`, `shmop_write()`, `shmop_read()`, `shmop_close()` and `shmop_delete()`.
- `substr()` and `iconv_substr()` now return an empty string, if string is equal to start characters long.
- `xml_set_object()` now requires to manually unset the `$parser` when finished, to avoid memory leaks.

Removed Extensions

- ereg
- mssql
- mysql
- sybase_ct

Removed SAPIs

- aolserver
 - apache
 - apache_hooks
 - apache2filter
 - caudum
 - continuity
 - isapi
 - milter
 - nsapi
 - phttpd
 - pi3web
 - roxen
 - thttpd
 - tux
 - webjames
- 