

PHP: ПРОФЕССИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ

УРОК 7. «НОВЫЕ» ВОЗМОЖНОСТИ PHP

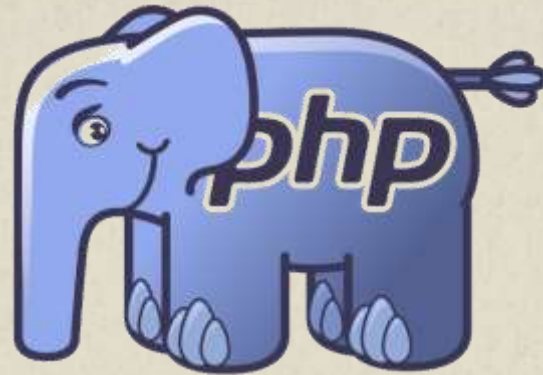
ОБРАЩЕНИЕ К «СКЛАДЧИКАМ»

Я считаю складчины – полной ерундой. Невозможно научиться чему-то, просматривая видео. Без домашних заданий, без общения с преподавателями и коллегами. Покупая в складчину видеозаписи курсов, вы вредите прежде всего самим себе, создавая иллюзию «обучения». И поддерживаете каких-то мутных личностей-«организаторов», имеющих свой процент.

Впрочем, дело ваше.

Однако, если вы хотите по-настоящему учиться – приходите. Адрес есть на слайдах. Напишите в поддержку, мол «я складчик, но я хочу учиться». Скидку гарантирую 😊





АНОНИМНЫЕ ФУНКЦИИ И ЗАМЫКАНИЯ

Анонимная функция – это функция, которая в отличие от обычной, не имеет имени

```
function ($x) {  
    return $x*2;  
}
```

// Это функция, возвращающая свой аргумент, умноженный на два

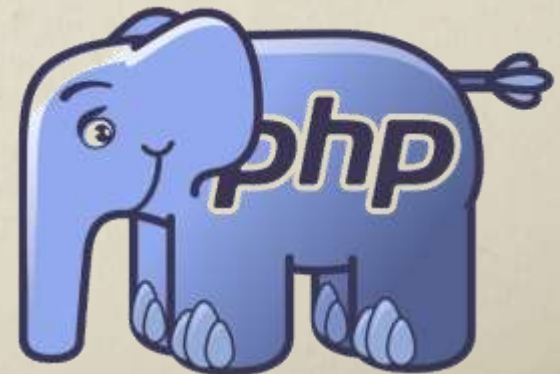
- Анонимную функцию можно вызвать сразу же, «на месте»:

```
echo (function ($x) {  
    return $x*2;  
}))(2);
```

- Можно присвоить переменной и так вызвать:

```
$f = function ($x) {  
    return $x*2;  
};
```

```
echo $f(2);
```



Анонимная функция – это «объект первого класса»

1. Она создается в рантайме (во время исполнения)
2. Анонимную функцию можно присвоить переменной
3. Ее можно вернуть из функции:

```
function getFilter() {  
    return function ($s) {  
        return str_replace(' ', '', $s);  
    }  
}
```

```
$filter = getFilter();  
echo $filter('Hello, world!');
```

4. Ее можно передать в функцию!

```
function apply($str, callable $filter) {  
    return $filter($str);  
}
```

```
echo apply(  
    'Hello, world!',  
    function ($s) {  
        return str_replace(' ', '', $s);  
    }  
);
```

Академия программирования "ProflT"



<http://pr-of-it.ru>

В PHP действует правило “Uniform Syntax”. Все конструкции расшифровываются слева направо!

Объясните, что значит:

- 1. `getFilter()('Foo');`**
- 2. `$filters[1]('Foo')`**
- 3. `getFilters()[1]('Foo');`**



Академия программирования “Profit”

**ЧТО ЭТО
ЗНАЧИТ?**



<http://pr-of-it.ru>

Замыкание – это способ передачи функции контекста

- Замыкание позволяет «замкнуть» на анонимную функцию какое-либо значение из родительского контекста:

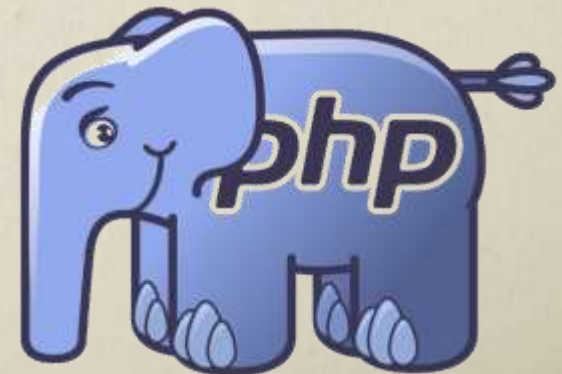
```
$hello = 'Hello';  
$h = function ($s) use ($hello) {  
    return $hello . ' ' . $s;  
}
```

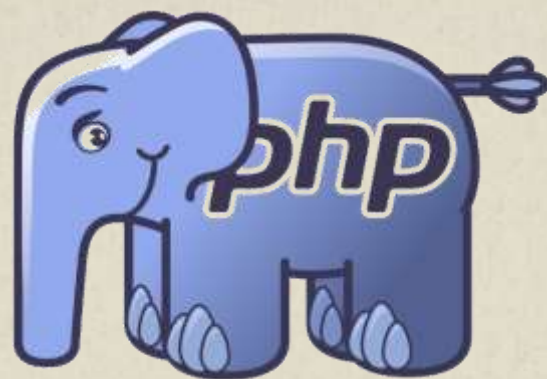
```
...  
echo $h('World');
```

- Замыкание – это не глобальная переменная! Значение замыкания «фиксируется» на момент создания анонимной функции.
- Важно: именно на момент создания, а не на момент вызова функции



ЗАМЫКАНИЕ





ГЕНЕРАТОРЫ

Генераторы – это способ создания итераторов. Вот так ☺

Выглядит генератор, как функция. Но вместо return – yield

```
function generate() {  
    for ($x=1;$x<10;$x++) {  
        yield $x;  
    }  
}
```

Фактически же генератор функцией не является. Это просто «сахар» для создания объекта с интерфейсом Iterator.

Единственное применение генератора, это его использование в контексте **foreach**:

```
foreach (generate() as $num) {  
    echo $num;  
}
```

Главный смысл : использование памяти. Даже случае последовательности из 1000 элементов требуется память для одного!

ГЕНЕРАТОРЫ



Можно генерировать значения с ключами

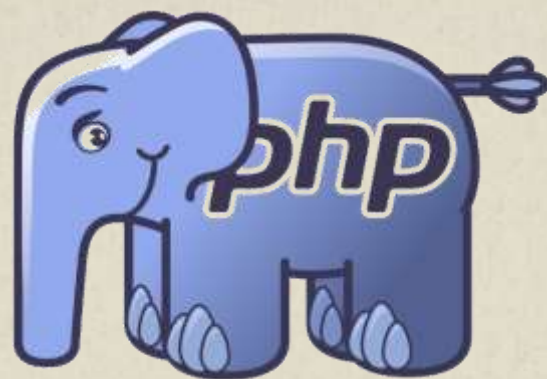
```
function generate() {  
    for ($x=1;$x<10;$x++) {  
        yield $x => $x*100;  
    }  
}  
  
foreach (generate() as $i => $num) {  
    echo $i . '=' . $num;  
}
```

Можно делегировать генерацию:

```
function count_to_ten() {  
    yield 1;  
    yield 2;  
    yield from [3, 4, 5, 6];  
    yield from get78();  
    yield 9;  
    yield 10;  
}
```

ОСОБЕННОСТИ ГЕНЕРАТОРОВ





ПЕРЕМЕННОЕ ЧИСЛО АРГУМЕНТОВ

Просто пример:

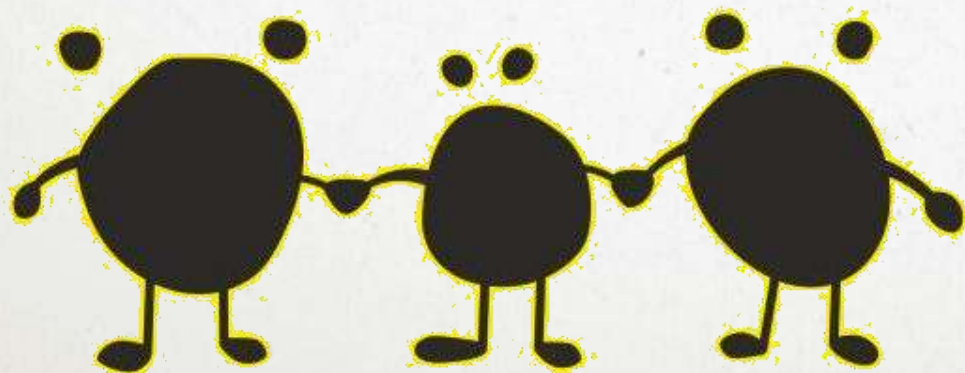
```
function sum(...$nums) {  
    return array_sum($nums);  
}
```

```
echo sum(1, 2, 3);
```

```
echo sum(...[1, 2, 3]);
```

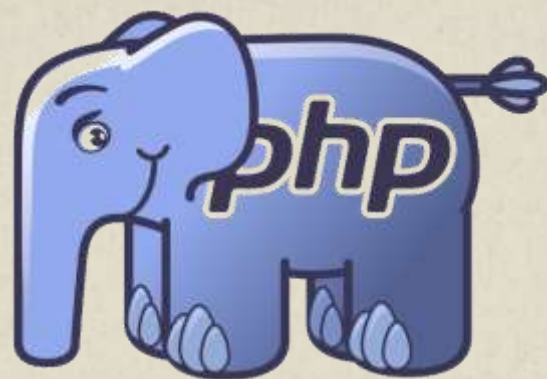
и даже так:

```
function test($a, $b, ...$options)
```



**ПЕРЕМЕННОЕ
ЧИСЛО
АРГУМЕНТОВ**





«НОВЫЕ» ОПЕРАТОРЫ

- **Традиционный ?:**
Тернарный оператор

```
$x ? $a : $b;  
эквивалентно  
if ($x == true) { $a; } else { $b; }
```

- **Сокращенный ?:**
Тернарный оператор без средней части

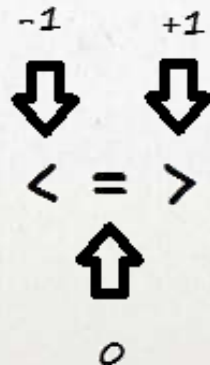
```
$x ?: $b;  
эквивалентно  
if ($x == true) { $x; } else { $b; }
```

- **Умный ??**
Оператор объединения с null

```
$x ?? $b;  
эквивалентно  
if (isset($x)) { $x; } else { $b; }
```

- **Космический <=>**
Оператор сравнения

```
$x <=> $y;  
эквивалентно  
if ($x < $y) { -1 }  
elseif ($x == $y) { 0 }  
elseif ($x > $y) { 1 }
```



«НОВЫЕ» ОПЕРАТОРЫ





КОНСТАНТНЫЕ ВЫРАЖЕНИЯ

Начиная с PHP 5.6 стало ВОЗМОЖНЫМ:

- Использовать в определении констант константные выражения (то есть выражения, которые могут быть вычислены на этапе компиляции):

```
const TWELVE = 6*2;  
const DOZEN = TWELVE . ' eggs';
```

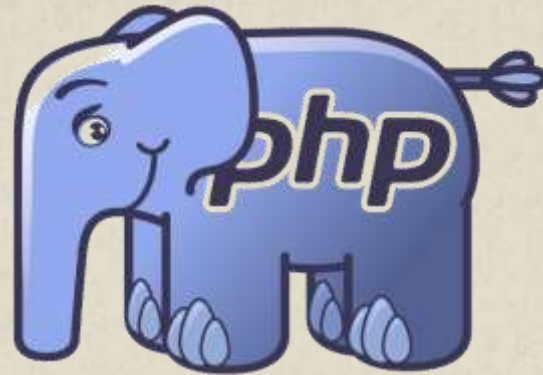
```
const FOO = $bar;  
const BAR = baz();
```

- Создавать константы-массивы:

```
const STATUSES = [  
    'SUCCESS' => 0,  
    'FAIL' => 1,  
];  
  
$this->status = STATUSES['SUCCESS'];
```

КОНСТАНТНЫЕ ВЫРАЖЕНИЯ





ФУНКЦИИ И КОНСТАНТЫ В ПРОСТРАНСТВАХ ИМЕН

Функции и константы тоже могут быть в пространствах имен, как и классы

- Определение констант внутри пространства имен:

```
namespace Math {  
  
    const E = 2.71828;  
  
    function tg($x) {  
        return sin($x) / cos($x);  
    }  
  
}
```

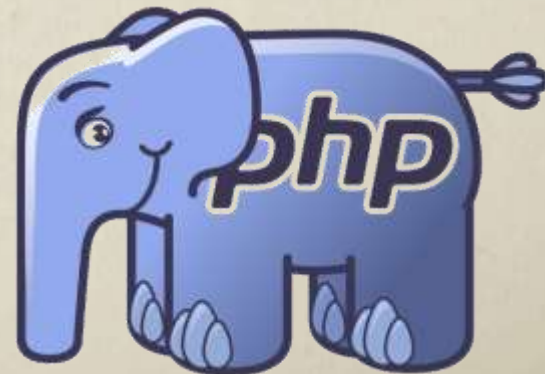
- Обращение по полному имени:

```
echo Math\PI;  
echo Math\tg(1);
```

- Импорт:

```
use function Math\tg;  
use function Math\tg as tangens;  
  
use const Math\PI;
```

NAMESPACES





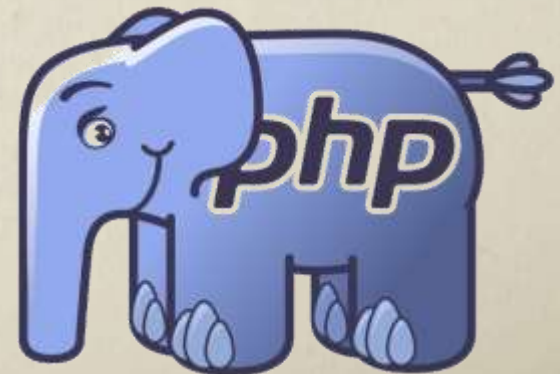
АНОНИМНЫЕ КЛАССЫ

Анонимный класс – это класс, не имеющий формально имени, создающийся «на лету»

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

NB. Честно, пока все примеры на применение анонимных классов несколько искусственные... Может быть вы найдете хороший?

АНОНИМНЫЕ КЛАССЫ



ДО ВСТРЕЧИ НА СЛЕДУЮЩЕМ УРОКЕ!

**ВИДЕОЗАПИСЬ, СЛАЙДЫ, ПРЕЗЕНТАЦИЯ
И ДОМАШНЕЕ ЗАДАНИЕ
БУДУТ ВЫЛОЖЕНЫ ДО 10 УТРА СЛЕДУЮЩЕГО ДНЯ**

