



Information Technology





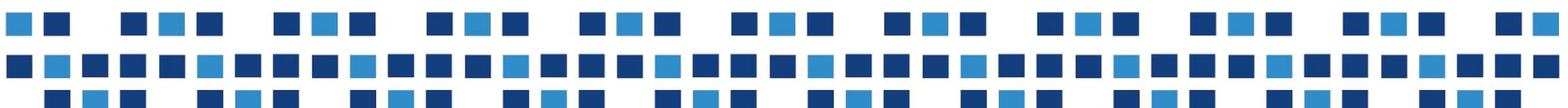
Строки. Массивы. Функции

Типы данных



- Строчный string
- Целочисельный integer
- Число с плавающей точкой float
- Логический boolean
- Массив array
- Объект object
- Ресурс resource
- Без значения / значение неизвестно null

RНР является слабо типизированным языком, нам не нужно указывать тип данных при объявлении переменной. Интерпретатор сам определяет ее тип, но также существует возможность явного приведения типов



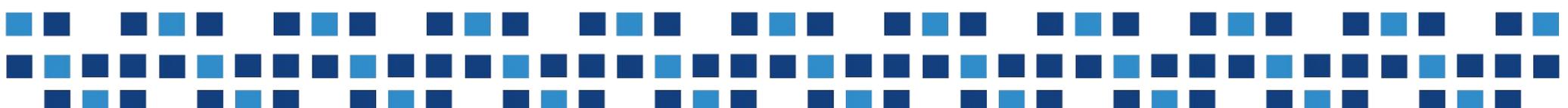
String



Значения строчного типа данных должны
быть заключены

в

```
"Hello, world!", 'car', 'There is 5 cats in the room'
```



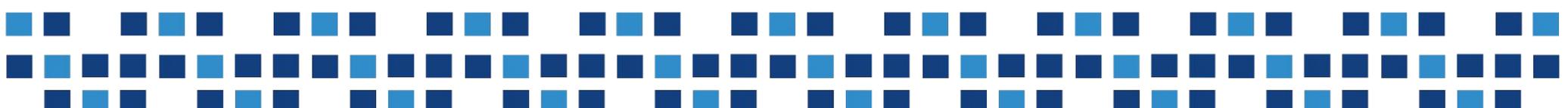
- **substr** - получить часть строки

```
string substr ( string $string , int $start [, int $length ] )
```

Возвращает подстроку строки *string*, начинающейся с *start* символа по счету и длиной *length* символов.

- **strlen()** - Возвращает длину строки

```
int strlen ( string $string )
```

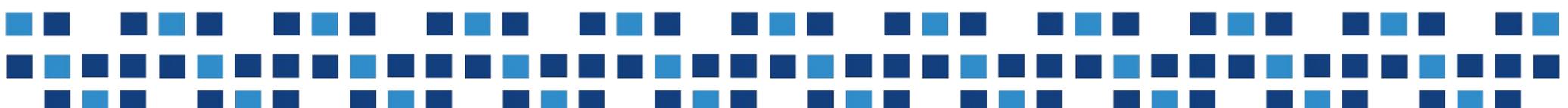


- **trim** - убрать лишние пробелы по краям строки

```
string trim ( string $str [, string $character_mask = " \t\n\r\0\x0B" ] )
```

- **strpos** - осуществляет поиск в строке

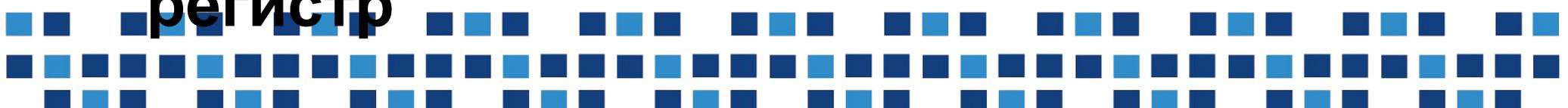
```
mixed strpos ( string $haystack , mixed $needle [, int $offset = 0 ] )
```



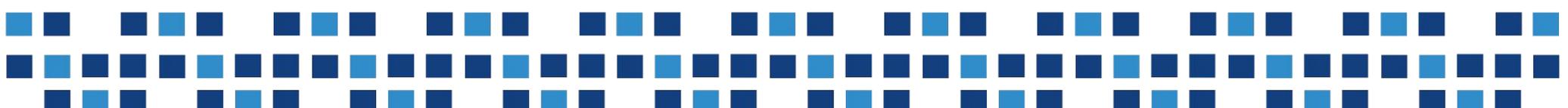
- **strip_tags** - удаляет тэги HTML и PHP из строки

```
string strip_tags ( string $str [, string $allowable_tags ] )
```

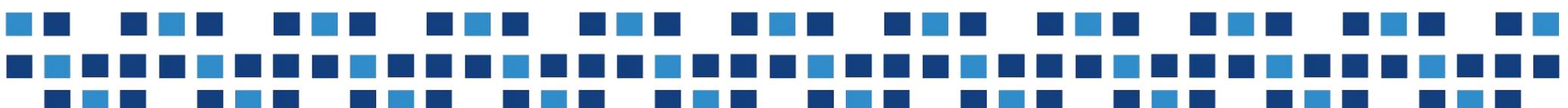
- **strtolower** - переводит строку в нижний регистр
- **strtoupper** - переводит строку в верхний регистр



- `md5` — Возвращает MD5-хэш строки
Возвращает хэш в виде 32-символьного шестнадцатеричного числа.



- `md5` — Возвращает MD5-хэш строки
Возвращает хэш в виде 32-символьного шестнадцатеричного числа.

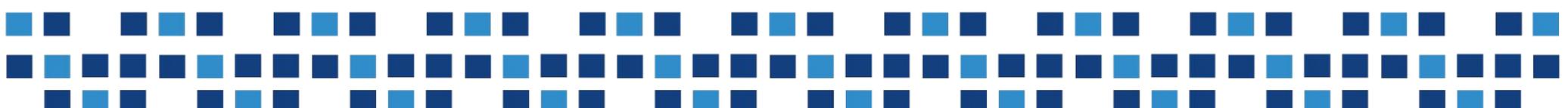


- **array_chunk** Разбивает массив на несколько меньших массивов заданного

```
array array_chunk ( array $array , int $size [, bool $preserve_keys = false ] )
```

- **array_flip** — Меняет местами ключи с их значениями в массиве

```
array array_flip ( array $array )
```

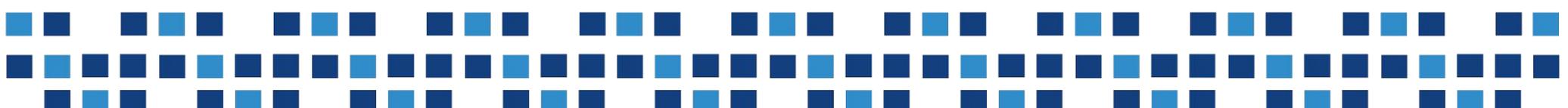


- `array_key_exists` — Проверяет, присутствует ли в массиве указанный ключ или индекс

```
bool array_key_exists ( mixed $key , array $array )
```

- `array_keys` — Возвращает все или некоторое подмножество ключей массива

```
array array_keys ( array $array [, mixed $search_value = null [, bool $strict = false ]] )
```

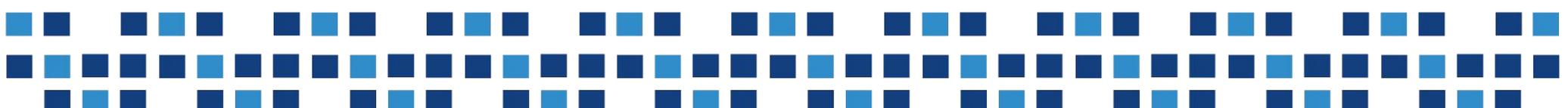


- `array_map` — Применяет callback-функцию ко всем элементам указанных массивов

```
array array_map ( callable $callback , array $array1 [ , array $... ] )
```

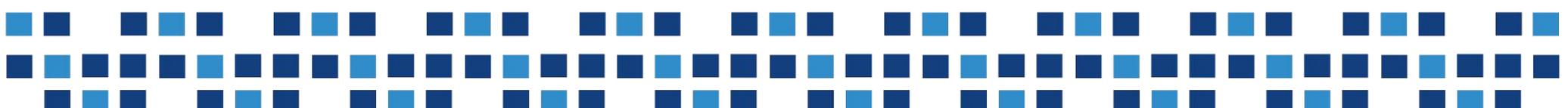
- `array_rand` — Выбирает одно или несколько случайных значений из массива

```
mixed array_rand ( array $array [ , int $num = 1 ] )
```



- `array_reverse` — Возвращает массив с элементами в обратном порядке

```
array array_reverse ( array $array [, bool $preserve_keys = false ] )
```

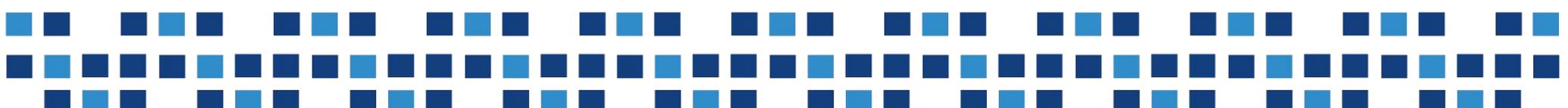


- `array_search` — Осуществляет поиск данного значения в массиве и возвращает ключ первого найденного элемента в случае удачи

```
mixed array_search ( mixed $needle , array $haystack [, bool $strict = false ] )
```

- `in_array` — Проверяет, присутствует ли в массиве значение

```
bool in_array ( mixed $needle , array $haystack [, bool $strict = FALSE ] )
```

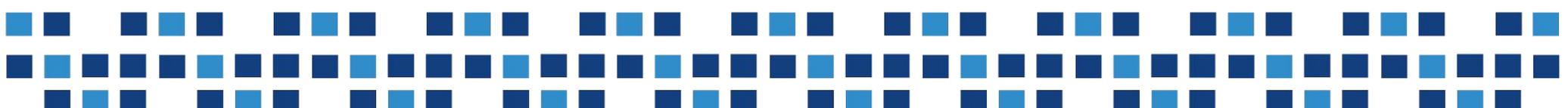


- `array_sum` — Вычисляет сумму значений массива

```
number array_sum ( array $array )
```

- `array_unique` — Убирает повторяющиеся значения из массива

```
array array_unique ( array $array [, int $sort_flags = SORT_STRING ] )
```



Функции по работе с массивами



- `sort` — Сортирует массив

```
bool sort ( array &$array [, int $sort_flags = SORT_REGULAR ] )
```

Флаги сортировки:

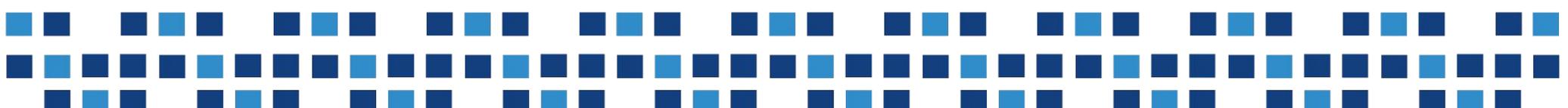
`SORT_REGULAR` - обычное сравнение элементов (без изменения типов)

`SORT_NUMERIC` - числовое сравнение элементов
`SORT_STRING` - строковое сравнение элементов

`SORT_LOCALE_STRING` - сравнивает элементы как строки с учетом текущей локали. Используется локаль, которую можно изменять с помощью функции `setlocale()`

`SORT_NATURAL` - сравнение элементов как строк, используя естественное упорядочение, аналогичное упорядочению в функции `natsort()`

`SORT_FLAG_CASE` - может быть объединен (побитовое ИЛИ) с константами `SORT_STRING` или `SORT_NATURAL` для сортировки строк без учета регистра.

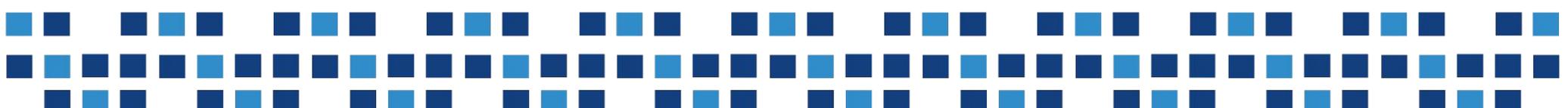


- `arsort` — Сортирует массив в обратном порядке, сохраняя ключи

```
bool arsort ( array &$array [, int $sort_flags = SORT_REGULAR ] )
```

- `asort` — Сортирует массив, сохраняя ключи

```
bool asort ( array &$array [, int $sort_flags = SORT_REGULAR ] )
```

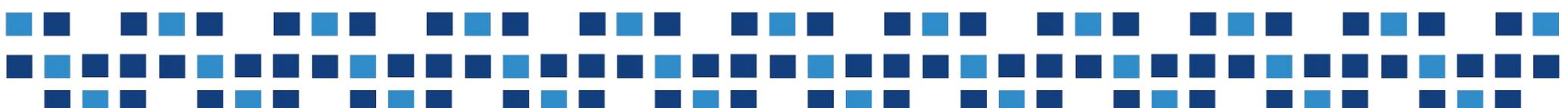


- `krsort` — Сортирует массив по ключам в обратном порядке

```
bool krsort ( array &$array [, int $sort_flags = SORT_REGULAR ] )
```

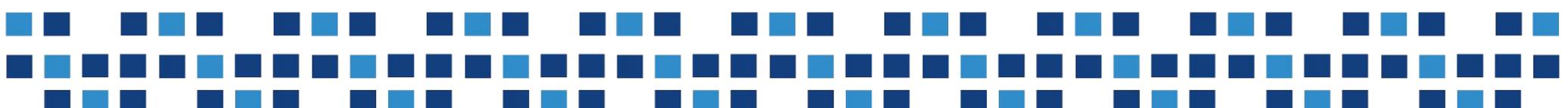
- `ksort` — Сортирует массив по ключам

```
bool ksort ( array &$array [, int $sort_flags = SORT_REGULAR ] )
```



Для **создания функции** необходимо определить следующие составляющие:

- **Название функции.**
- **Аргументы (параметры) функции**
- **Тело функции**
- **Возвращаемые значения**

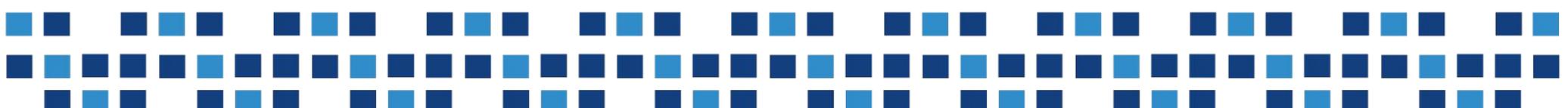


Самописные функции



Базовый синтаксис функции

```
function someName ([params]) {  
  [body]  
}
```



Функции по работе со строками

- <http://php.net/manual/ru/ref.mbstring.php>
- <http://php.net/manual/ru/book.strings.php>

Функции по работе с массивами

<http://php.net/manual/ru/book.array.php>

Справочник php функций

<http://php.net/>

