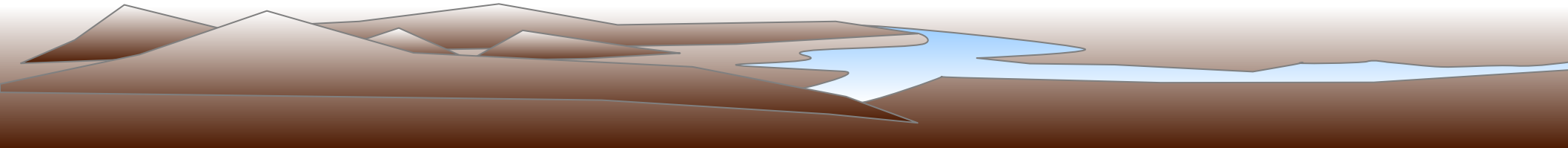


# Функции и Code Reuse

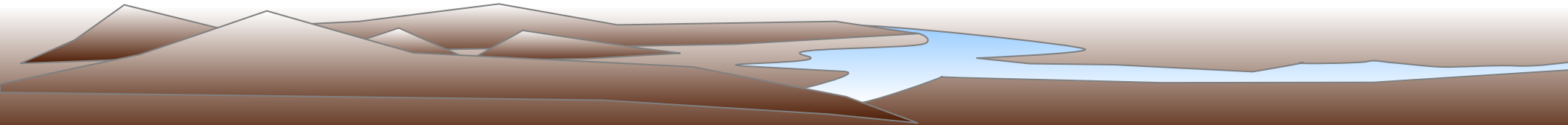


# Функции и Code Reuse

Функция — это часть программы, обозначенная определенным именем, выполняющая определенную задачу и необязательно возвращающая результат.

```
<?php
    function PrintName ($name) {
        echo $name;
    }

    $boy = 'Jack';
    PrintName($boy); // Выведет: «Jack»
    PrintName('Sally'); // Выведет: «Sally»
?>
```



# Функции и Code Reuse

## Области видимости

```
<?php
    $a = 1;           // Глобальная область видимости
    function Test() {
        echo $a;     // Локальная область видимости
    }
    Test();          // Не выведет ничего
?>
```

Получить доступ к глобальным переменным из локальной области видимости можно следующими способами:

```
<?php
    $a = 1; $b = 2;
    function Sum () {
        global $a, $b;
        return $b += $a;
    }
    echo Sum();

<?php // Рекомендуемый вариант
    $a = 1; $b = 2;
    function Sum () {
        return $GLOBALS['b'] += $GLOBALS['a'];
    }
    echo Sum();
?>
```

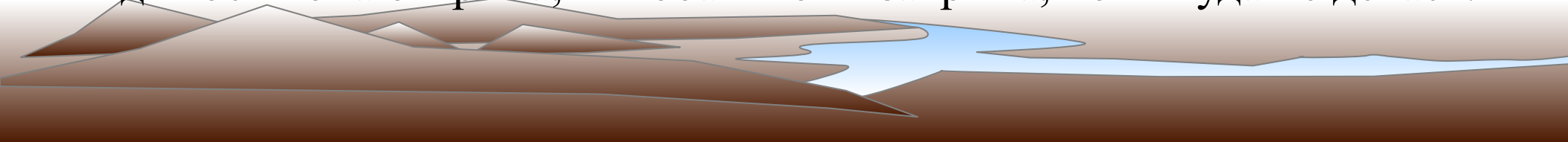
# Функции и Code Reuse

## Статические переменные

Есть другой способ определить переменную так, чтобы ее значение хранилось вне функции, но было доступно также и внутри нее. Способ заключается в использовании статических переменных.

```
<?php
function Test() {
    static $a = 0;
    echo $a;
    $a++;
}
Test();           // 0
echo $a;         // Ошибка:$a нет в глобальной области видимости
Test();          // 1
Test();          // 2; Переменная $a сохранила свое значение
?>
```

Статическая переменная как коробочка: в локальной области видимости она открыта, в глобальной - закрыта, но никуда не делась.



# Функции и Code Reuse

## Математические функции

**abs** – Возвращает абсолютное значение числа.

**acos** – Возвращает значение арккосинуса значения.

**cos** – Возвращает косинус числа в радианах.

**base\_convert** – Преобразует число из одного основания в другое.

**decbin** – Возвращает двоичное представление целого числа.

**log** - Возвращает натуральный логарифм значения.

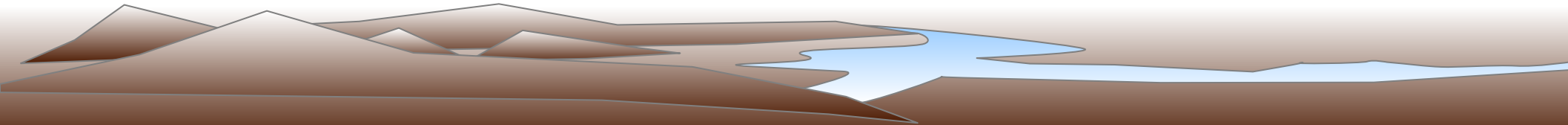
**pi** – Возвращает приближительное число пи.

**rad2deg** – Возвращает в градусах значение аргумента в радианах.

**round** – Возвращает округленное значение аргумента.

**sqrt** – Извлечение квадратного корня из числа.

**rand** – Возвращает число, лежащее между двумя необязательными аргументами включительно.



# Функции и Code Reuse

## Функции работы с массивами

`array_chunk` - Разбить массив на части.

`array_merge` - Слить два или большее количество массивов.

`array_pop` - Извлечь последний элемент массива.

`array_push` - Добавить один или несколько элементов в конец массива.

`array_rand` - Выбрать одно или несколько случайных значений.

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

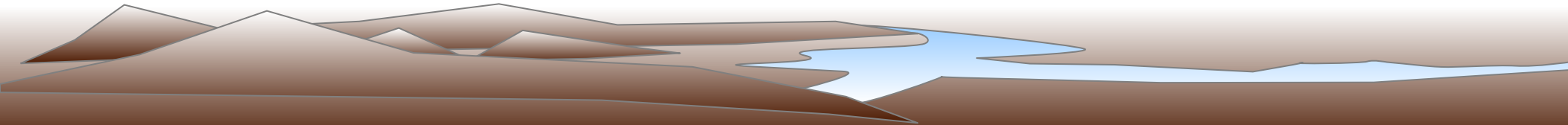
`array_shift` - Извлечь первый элемент массива.

`array_splice` - Удалить последовательность элементов массива и заменить её другой последовательностью.

`count` - Посчитать количество элементов массива.

`in_array` - Проверить, присутствует ли в массиве значение.

`sort` - Отсортировать массив.



# Функции и Code Reuse

## Функции работы со строками

**addslashes** - Экранирует спецсимволы в строке.

**chr** - Возвращает символ по его коду.

**count\_chars** - Возвращает информацию о символах строки.

**explode** - Разбивает строку на подстроки.

**htmlentities** - Преобразует спецсимволы в их HTML-аналоги.

**join** - Объединяет элементы массива в строку.

**trim** - Удаляет пробелы из начала и конца строки.

**str\_replace** - Заменяет строку поиска на строку замены.

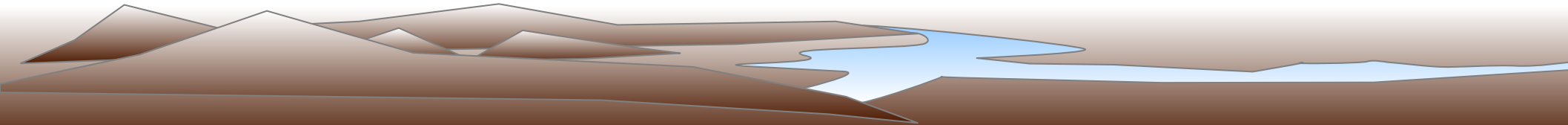
**str\_split** - Преобразует строку в массив.

**strip\_tags** - Удаляет HTML и PHP тэги из строки.

**strlen** - Возвращает длину строки.

**strpos** - Возвращает позицию первого вхождения подстроки.

**strstr** - Находит первое вхождение подстроки.



# Функции и Code Reuse

## Функции работы с датой и временем

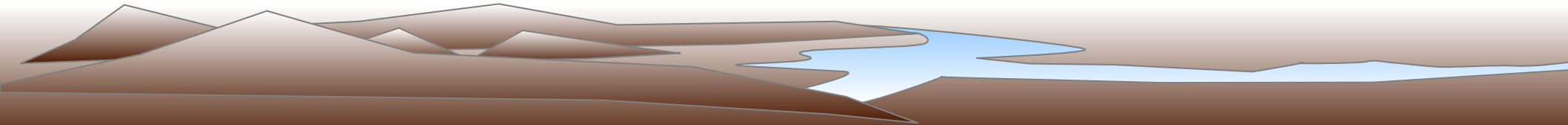
**date** - Возвращает время, отформатированное в соответствии с аргументом `format`, используя метку времени, заданную аргументом `timestamp` или текущее системное время, если `timestamp` не задан.

**gmdate** - Эта функция идентична функции `date()` за исключением того, что возвращает время по Гринвичу (GMT)

**getdate** - Возвращает ассоциативный массив, содержащий информацию о дате, представленной меткой времени `timestamp` или текущем системном времени, если `timestamp` не передан.

**time** — Возвращает количество секунд, прошедших с начала эпохи UNIX (1 января 1970, 00:00:00 GMT) до текущего момента.

**mktime** — Возвращает метку времени для заданной даты.



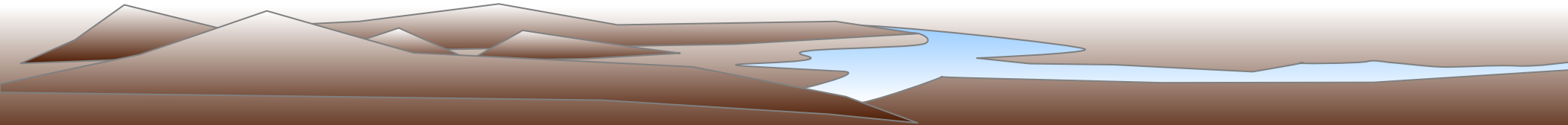


# Функции и Code Reuse

## Повторное использование кода

Чтобы подключить в текущий файл код из стороннего файла, можно использовать следующие конструкции языка PHP:

- `include` — Подключает код из указанного файла в текущий.
- `require` — Работает аналогично `include`. Разница в том, что `require`, не найдя указанного файла, выведет ошибку и заставит сценарий завершиться, тогда как `include` выведет лишь предупреждение, а сценарий продолжит работу.
- `include_once` — Подключит код только, если он еще не подключен, поможет избежать двойного подключения кода и связанных с ним ошибок.
- `require_once` — Работает аналогично `include_once`, но с особенностями `require`.

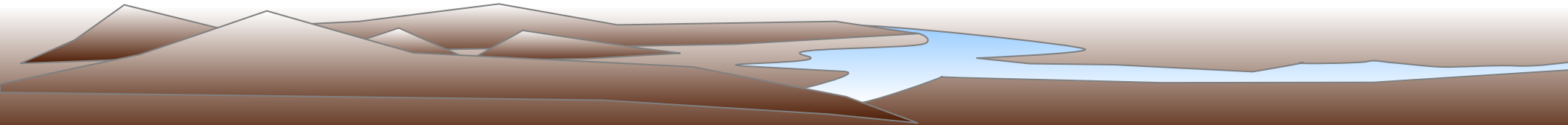


# Функции и Code Reuse

## Повторное использование кода. Пример

```
<?php // файл Sum.function.php
function Sum ($a, $b){
    return $a + $b;
}
?>
```

```
<?php // файл index.php
echo Sum (1,2); // Вызовет ошибку, т.к. Sum не существует
include ('Sum.function.php'); // Подключаем Sum
echo Sum (1,2); // Выведет 3
?>
```



# Функции и Code Reuse

## Демонстрация

Создание и вызов функций;

Области видимости;

Использование статических переменных;

Использование функций даты и времени;

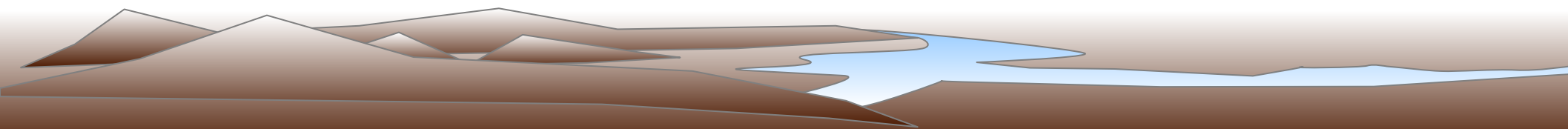
Использование строковых функций;

Использование функции для работы с массивами;

Использование предопределенных констант;

Использование предопределенных переменных;

Подключение файлов.



# Функции и Code Reuse

## Практическая работа

1. Создайте функцию, принимающую три аргумента — имя, фамилию и возраст. Функция должна выводить HTML-таблицу, ячейками которой и будут эти аргументы. Несколько раз вызовите функцию.
2. Создайте массив с несколькими произвольными элементами. Выведите массив с элементами в обратном порядке, а так же количество элементов.
3. Создайте строку «Я не люблю PHP». Используя функции работы со строками, замените «не» на «очень». Подсчитайте и выведите количество символов в строке. Выведите текущие дату и время.
4. Создайте таблицу, которая будет основой одностраничного сайта. Разделите таблицу и вынесите части в отдельные файлы (top.php, menu.php, bottom.php). Подключите части в исходный скрипт.

