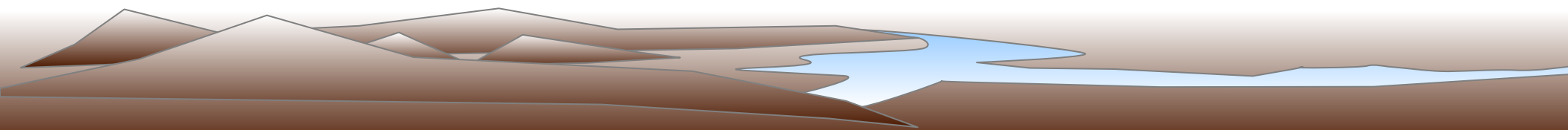


Введение в РНР



Введение в PHP

Базовый синтаксис

PHP — это рекурсивный акроним аббревиатуры PHP Hypertext Preprocessor. Команды на языке PHP обрамляются специальными дескрипторами — тэгами языка PHP. Все, что находится вне этих тегов, игнорируется интерпретатором. Поддерживаются следующие стили написания тэгов:

- XML-стиль (рекомендуемый);

`<?php код на PHP ?>`

- HTML-стиль;

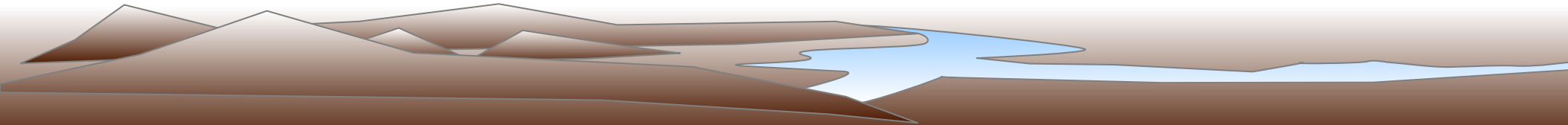
`<script language="php"> код на PHP </script>`

- Краткий стиль;

`<? код на PHP ?>`

- ASP-стиль.

`<% код на PHP %>`



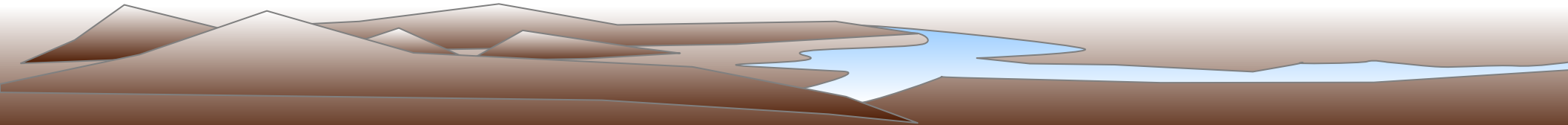
Введение в PHP

Базовый синтаксис

Существует ряд требований, которые необходимо соблюдать при программировании на PHP:

- Каждая команда заканчивается точкой с запятой (;);
- Одну команду можно записывать в несколько строк или несколько команд в одну строку;
- PHP чувствителен к регистру символов в именах переменных и функций;

```
<?php
    $index = 10;
    print($Index);           // Ошибка
?>
```



Введение в PHP

Базовый синтаксис

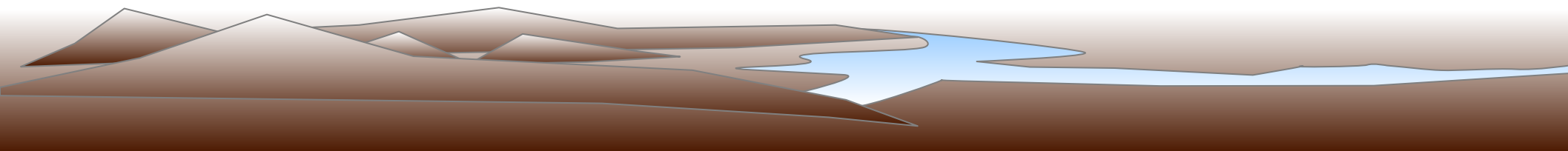
- PHP нечувствителен в отношении ключевых слов, к пробелам, переводам строки, знакам табуляции.

Этот код полностью корректен:

```
<?php
```

```
$index = 10;  
$index = 10 + 20;  
$index = 10+10;  
$index =  
10  
+  
10;
```

```
?>
```



Введение в PHP

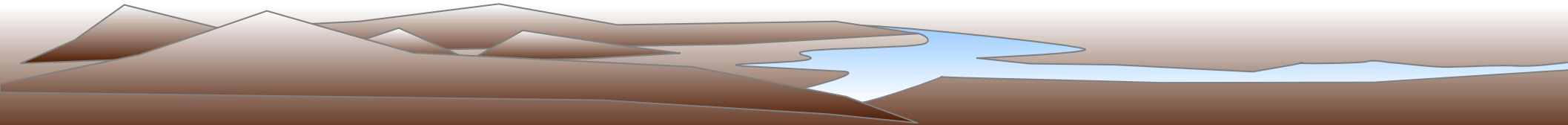
Комментарии

PHP поддерживает три вида комментариев: один многострочный и два однострочных. PHP-парсер никак не анализирует комментарии, они просто игнорируются.

```
<?php
/*
    Первый
    вид
    комментария
*/

// Второй

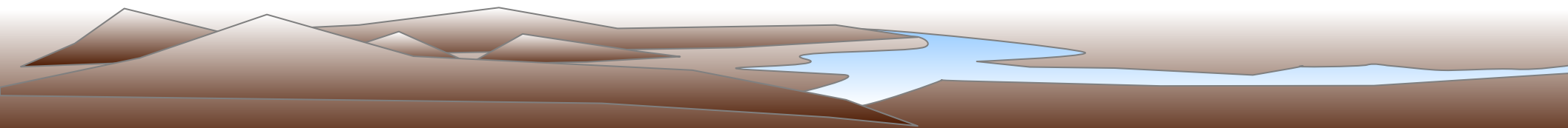
# Третий
?>
```



Введение в RНР

Переменные

- Все имена переменных должны начинаться со знака доллара (\$);
- Объявления не являются обязательными. Переменная начинает существовать с момента присвоения ей значения или с момента первого использования. Если использование начинается раньше присвоения, то переменная будет содержать значение по умолчанию;
- Переменной не назначается определенный тип. Тип определяется хранящимся значением и текущей операцией.

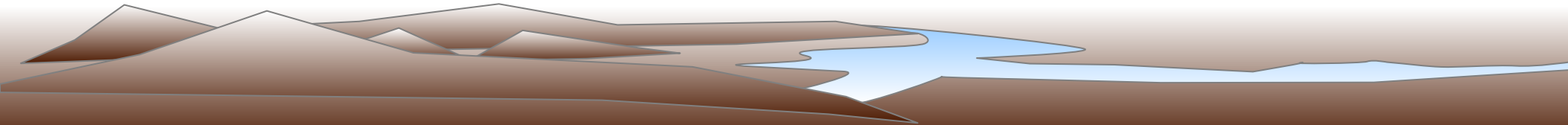


Введение в PHP

Переменные

Первым символом после \$ должна быть буква или символ подчеркивания. Далее в имени переменной могут присутствовать буквы, цифры и символ подчеркивания.

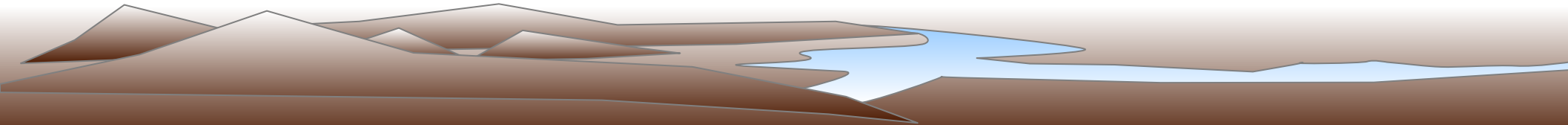
```
<?php
$I;           // Допустимо
$1;           // Недопустимо
$_1;          // Допустимо
$firstName;   // Допустимо
$7Lucky;      // Недопустимо
$~password;   // Недопустимо
$Last!Visit;  // Недопустимо
$Compute-Mean; // Недопустимо
?>
```



Введение в PHP

Переменные. Пример

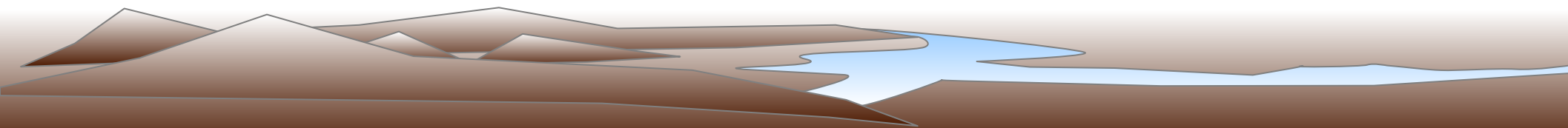
```
<?php
    $foo = 'Bob';           // Присваивает $foo значение 'Bob'
    $foo = "My name is Mike"; // Изменение $foo
    $bar = 25;              // Присваивает $bar значение 25
    $bar = 2 + 2;           // Присваивает $bar 4
    $tmp = $foo;            // Присваивает $tmp значение $foo
    $tmp = &$foo;           // Ссылка на $foo через $tmp
    $foo = "John";          // Изменение $foo
    echo $tmp;              // Выведет на экран "John"
    $foo = "Mike";          // Изменяем значение $foo
    unset($foo);            // Удаляем переменную $foo
    echo $tmp;              // Выведет на экран "Mike"
?>
```



Введение в PHP

Предопределенные переменные

- `$GLOBALS`** — Массив, содержащий все глобальные переменные.
- `$_ENV`** — Массив переменных окружения.
- `$_COOKIE`** — Массив файлов cookie, отправленных на сервер.
- `$_GET`** — Массив переменных, отправленных методом GET.
- `$_POST`** — Массив переменных, отправленных методом POST.
- `$_FILES`** — Массив, содержащий информацию о загруженных файлах.
- `$_REQUEST`** — Массив, содержащий `$_GET`, `$_POST`, `$_FILES`, `$_COOKIE`.
- `$_SESSION`** — Массив переменных, размещенных в сессиях PHP.
- `$_SERVER`** — Массив, содержащий информацию о сервере.



Введение в PHP

Типы данных

PHP поддерживает восемь типов данных.

Четыре скалярных типа:

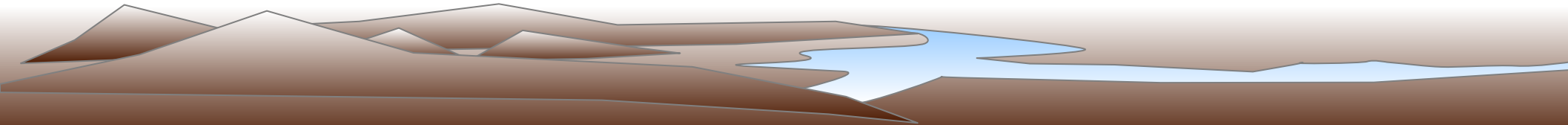
- **boolean** — логический;
- **integer** — целое число;
- **float (double)** — число с плавающей точкой;
- **string** — строка.

Два смешанных типа:

- **array** — массив;
- **object** — экземпляр класса.

Два специальных типа:

- **resource** — ссылка на внешний по отношению к скрипту источник данных (файл на диске, изображение в памяти и т.п.);
- **NULL** — отсутствие какого либо значения.

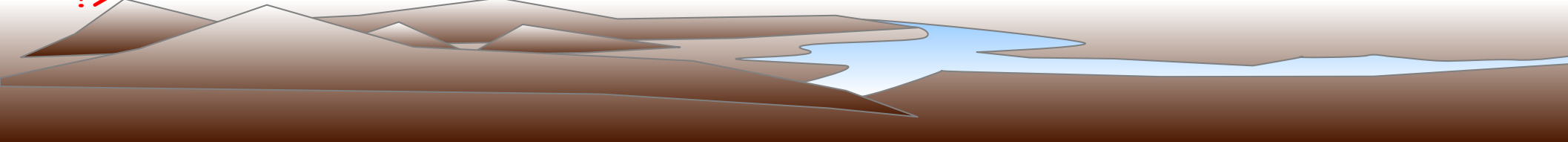


Введение в PHP

Типы данных. Пример

```
<?php
$foo = TRUE;           // Логический
$int = 1234;           // Целое число
$flt = 1.234;          // Число с плавающей точкой
// Это простая строка
echo "Это простая строка";
// Это вставит: новую строку
echo "Это вставит: \n новую строку";
// Переменная ОК вставилась в текст
$a = "ОК"; echo "Переменная $a вставилась в текст";
// Это простая строка
echo 'Это простая строка';
// Он сказал "I'll be back"
echo 'Он сказал: "I\'ll be back"';
// Это не вставит: \n новую строку
echo 'Это не вставит: \n новую строку';
//Переменная $a не подставляется
$a = "ОК"; echo 'Переменная $a не подставляется';
```

?>



Введение в PHP

Типы данных. Полезные функции

isset (имя_переменной)
переменная.

- сообщает, существует ли

unset (имя_переменной)

- уничтожает указанную переменную

empty (имя_переменной)
какое-либо значение.

- сообщает, присвоено ли переменной

gettype(имя_переменной)
переменной

- возвращает тип указанной

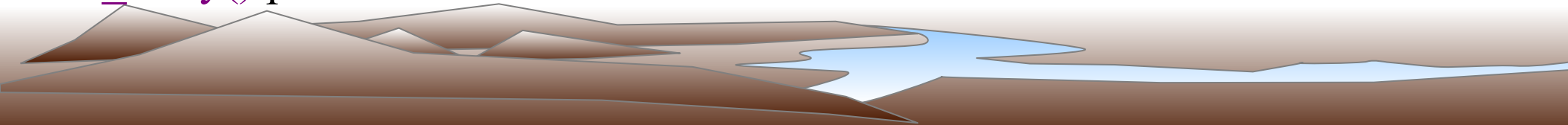
settype(имя_переменной, тип)
тип.

- конвертирует переменную в другой

is_bool(имя_переменной)
переменной логическим.

- проверяет является ли тип

Функции **is_numeric()**, **is_float()**, **is_int()**, **is_string()**, **is_object()**, **is_array()** работают по аналогии.

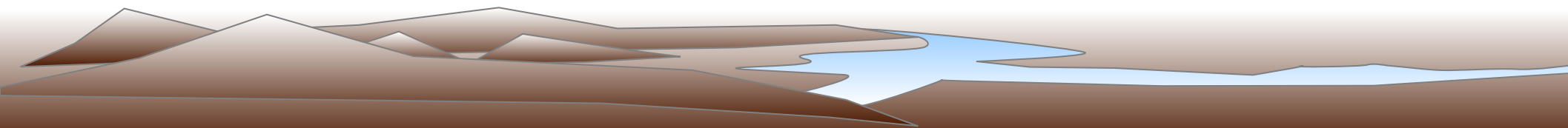


Введение в PHP

Константы

Для задания значений, которые не будут меняться в ходе выполнения сценария можно использовать константы. Так же как и переменные, константы могут быть определены и доступны в любом месте сценария, но у них есть и ряд особенностей:

- У констант нет префикса в виде знака доллара;
- Константам нельзя присваивать значения, их можно определить вызовом функции `define()`;
- Константы не могут быть определены или аннулированы после первоначального объявления.



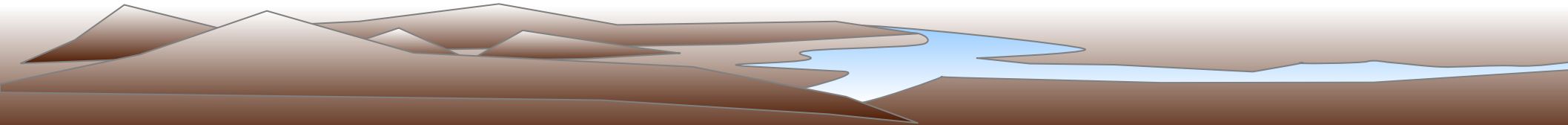
Введение в PHP

Константы. Пример

```
<?php
```

```
define ('PI', 3.14);  
$index = 10 * PI;           // Верно  
PI = 10 * 3.14;             // Ошибка!  
  
define("CONSTANT", "Здравствуй, мир.");  
echo CONSTANT;              // Выведет "Здравствуй, мир."  
echo Constant;              // Выведет "Constant" и предупреждение
```

```
?>
```



Введение в PHP

Предопределенные константы

__LINE__

- Номер текущей строки.

__FILE__

- Полный путь и имя текущего файла.

__FUNCTION__

- Имя текущей функции.

__CLASS__

- Имя текущего класса.

PHP_EXTENSION_DIR

- Каталог расширений PHP

PHP_OS

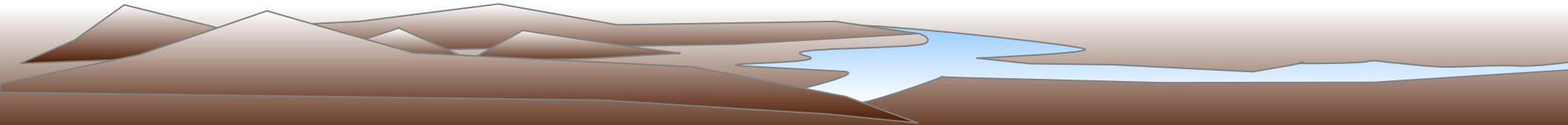
- Операционная система

PHP_VERSION

- Версия PHP

PHP_CONFIG_FILE_PATH

- Каталог размещения php.ini

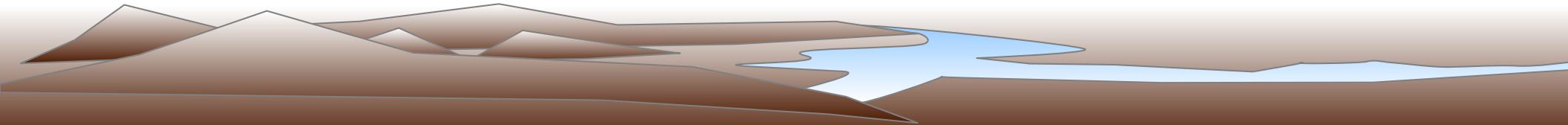


Введение в РНР

Операторы

Операторы бывают трех видов:

1. унарные, те, которые работают только с одним аргументом.
2. бинарные — с двумя.
3. тернарный оператор ?:..



Введение в PHP

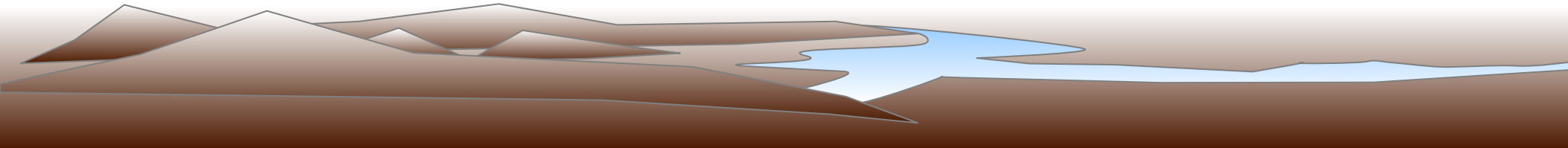
Арифметические операции

<code>-\$a</code>	// Смена знака
<code>\$a + \$b</code>	// Сумма
<code>\$a - \$b</code>	// Разность
<code>\$a * \$b</code>	// Произведение
<code>\$a / \$b</code>	// Частное
<code>\$a % \$b</code>	// Остаток от деления
<code>\$a += \$b</code>	// Аналогично <code>\$a = \$a + \$b</code>
<code>\$a -= \$b</code>	// Аналогично <code>\$a = \$a - \$b</code>
<code>\$a *= \$b</code>	// Аналогично <code>\$a = \$a * \$b</code>
<code>\$a /= \$b</code>	// Аналогично <code>\$a = \$a / \$b</code>
<code>\$a %= \$b</code>	// Аналогично <code>\$a = \$a % \$b</code>

Введение в PHP

Операции сравнения

<code>\$a == \$b</code>	// TRUE если \$a равно \$b.
<code>\$a === \$b</code>	// TRUE если \$a равно \$b И имеет тот же тип
<code>\$a != \$b</code>	// TRUE если \$a не равно \$b.
<code>\$a !== \$b</code>	// TRUE если \$a не равно \$b ИЛИ у них разные типы.
<code>\$a < \$b</code>	// TRUE если \$a строго меньше \$b.
<code>\$a > \$b</code>	// TRUE если \$a строго больше \$b.
<code>\$a <= \$b</code>	// TRUE если \$a меньше или равно \$b.
<code>\$a >= \$b</code>	// TRUE если \$a больше или равно \$b.



Введение в PHP

Логические операции

`$a and $b` // TRUE если и `$a`, и `$b` TRUE.

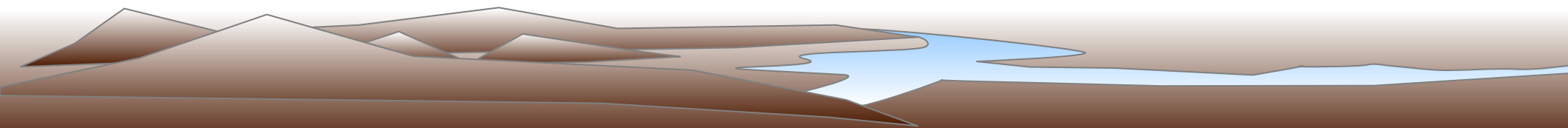
`$a or $b` // TRUE если или `$a`, или `$b` TRUE.

`$a xor $b` // TRUE если `$a`, или `$b` TRUE, но не оба.

`!$a` // TRUE если `$a` не TRUE.

`$a && $b` // TRUE если и `$a`, и `$b` TRUE.

`$a || $b` // TRUE если или `$a`, или `$b` TRUE.



Введение в RНР

Побитовые операции

$\$a \ \& \ \b // Побитовое И

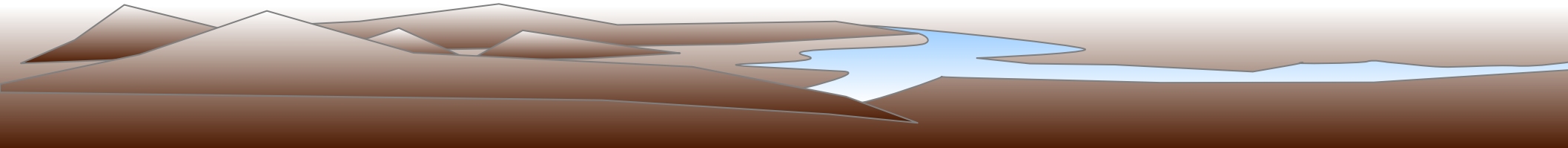
$\$a \ | \ \b // Побитовое ИЛИ

$\$a \ \wedge \ \b // Исключающее или

$\sim \$a$ // Отрицание

$\$a \ \ll \ \b // Побитовый сдвиг влево

$\$a \ \gg \ \b // Побитовый сдвиг вправо



Введение в PHP

Специфичные операции

Конкатенация

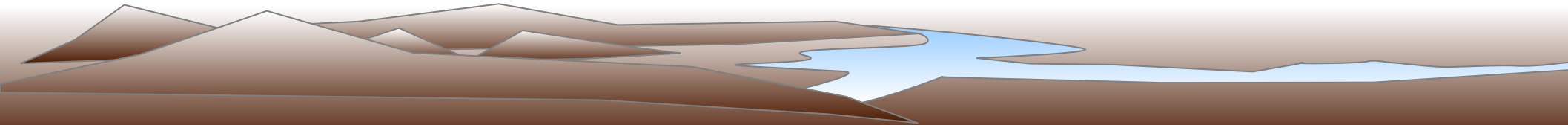
```
$a = "Hello ";
```

```
$b = $a . "World!"; // $b содержит строку "Hello World!"
```

```
$a .= "World!";      // $a содержит строку "Hello World!"
```

Подавление ошибки

```
@$a = 1 / 0;          // Ошибка не будет сгенерирована
```



Введение в PHP

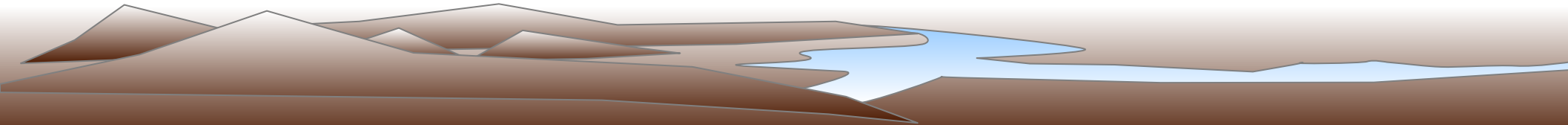
Инкремент / Декремент

```
++$a    // Увеличивает $a на единицу и возвращает значение $a
$a++    // Возвращает значение $a, а затем увеличивает $a на единицу
--$a    // Уменьшает $a на единицу и возвращает значение $a
$a--    // Возвращает значение $a, а затем уменьшает $a на единицу
```

<?php

```
$a = 5;
echo "Должно быть 5: " . $a++ . "<br>";
echo "Должно быть 6: " . $a . "<br>";
$a = 5;
echo "Должно быть 6: " . ++$a . "<br>";
echo "Должно быть 6: " . $a . "<br>";
```

?>



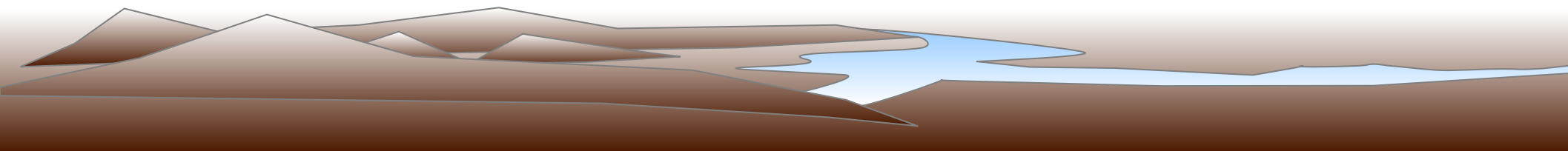
Введение в PHP

Тернарная операция

По сути является аналогом условной конструкции **if...else**
Она записывается следующим образом:

условие ? значение, если условие истинно : значение, если ложно

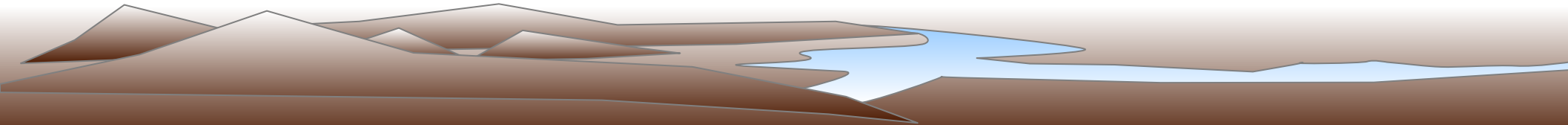
```
<?php
    $grade = 3;
    $result = ($grade > 2 ? 'Сдал' : 'Не сдал');
    echo $result;
?>
```



Введение в PHP

Демонстрация

- Использование echo и print;
- Использование переменных и констант;
- Использование специальных функций;
- Использование арифметических операторов;
- Использование операторов сравнения;
- Использование логических операторов.



Введение в PHP

Практическая работа

1. Создайте файл 1-1.php, содержащий 5 разных переменных, присвойте переменным значения разного типа. Используя `gettype()` выведите тип каждой переменной.
2. Создайте файл 1-2.php, содержащий 2 переменные числового типа. Произведите над переменными произвольное арифметическое действие и выведите его результат.
3. Создайте файл 1-3.php, содержащий 2 переменные строкового типа. Инициализируйте переменные произвольным текстом. С помощью конкатенации объедините содержимое переменных и выведите результат.
4. Создайте файл 1-4.php, содержащий 2 переменные с одинаковым типом значений. Используя тернарный оператор сравнения проведите исследование на возвращаемые результаты.

