

ОБРАЗОВАТЕЛЬНЫЙ ОНЛАЙН-КУРС

PHP-разработчик с нуля до PRO

КУРС 1. МОДУЛЬ 3.

■ Структура управления данными. Базовые концепции PHP7

Встроенные (предопределенные) константы

`_FILE_`

В данную константу интерпретатор помещает путь к файлу исполняемого скрипта от корня сайта.
Этот путь содержит как директории, так и само название файла (включая расширение)

`_LINE_`

Константа `_LINE_` содержит в себе номер строки в исполняемом скрипте, на котором она объявлена

`_DIR_`

Эта константа содержит в себе полный путь до директории, в которой лежит исполняемый скрипт

`_FUNCTION_`

При выполнении скрипта интерпретатор положит в данную константу название функции, в которой была использована данная константа. Если мы попытаемся использовать эту константу вне функции, то ее значение будет равно пустой строке

`_CLASS_`

Хранит в себе название класса, в котором была использована данная константа. Подробнее мы поговорим в следующих модулях, посвященным классам

`_METHOD_`

В данной константе хранится название метода класса

`_NAMESPACE_`

Хранит название пространства имен класса, в котором была использована

Встроенные суперглобальные массивы

1. **`$_GET`**

Переменная `$_GET` содержит в себе данные, переданные на сервер через адресную строку браузера

2. **`$_POST`**

Переменная `$_POST` содержит в себе данные, переданные на сервер методом POST, например через форму

3. **`$_FILES`**

Суперглобальный массив `$_FILES` используется в случае загрузки файлов на сервер

4. **`$_COOKIE`**

В массиве `$_COOKIE` содержаться данные, которые были сохранены в браузере пользователя

5. **`$_REQUEST`**

Эта суперглобальная переменная совмещает в себе суперглобальные массивы `$_GET`, `$_POST` и `$_COOKIE`

6. **`$_SESSION`**

Содержит в себе данные, связанные с пользовательской сессией

7. **`$_SERVER`**

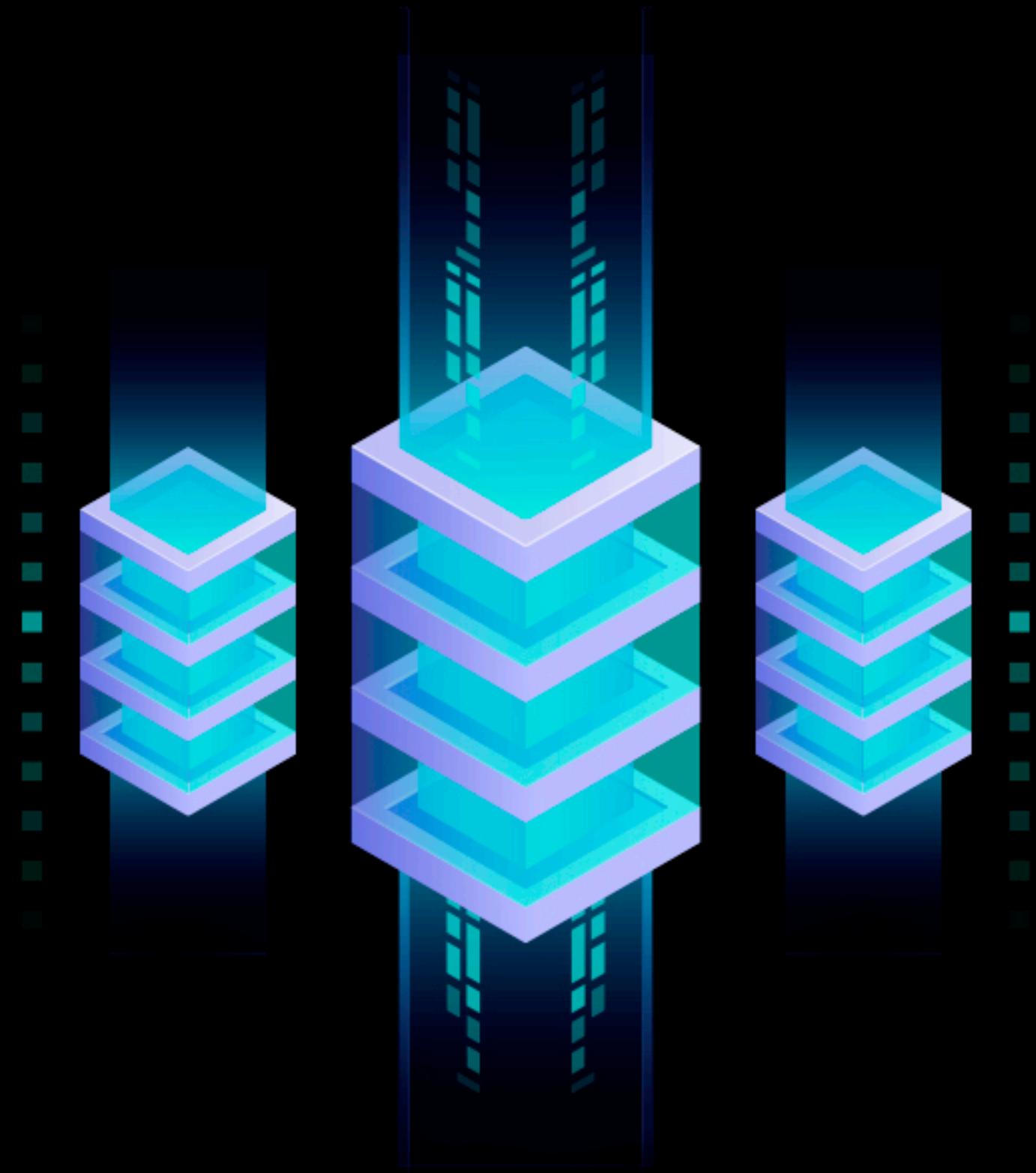
Информация о сервере и среде исполнения

8. **`$_GLOBALS`**

Массив, содержащий значения всех переменных из глобальной области видимости

Выражения

Выражения – это своего рода «кирпичики», из которых строятся PHP-программы. Под выражением в PHP понимается то, что имеет значение, либо это значение можно вычислить, например, вызов функции, использование операторов, или просто обращение к переменной по имени –



■ Операторы —

Оператор присвоения (=)

Для присваивания значений используется оператор присваивания обозначаемый как «=».

Оператор = присваивает значение, расположенное с правой стороны от него, операнду, который располагается с левой стороны. Так же левым operandом оператора = может быть элемент массива или свойство объекта.

Правым же operandом может служить любое значение, любого типа

\$a = \$b

Оператор отрицания (-)

Оператор отрицания обозначается символом «-».

Он применяется к одному операнду, располагаясь перед ним
и возвращает значение операнда с противоположным знаком

\$a = -\$b;

Оператор сложения (+)

Оператор сложения обозначается символом «+» и возвращает сумму двух operandов

\$a = \$b + \$c;

Оператор вычитания (-)

Оператор вычитания используется для определения разности двух operandов и обозначается символом «-». В отличие от оператора отрицания работает с двумя operandами

\$a = \$b - \$c;

Оператор умножения (*)

Оператор умножения «*» возвращает произведение двух operandов

\$a = \$b * \$c;

Оператор деления (/)

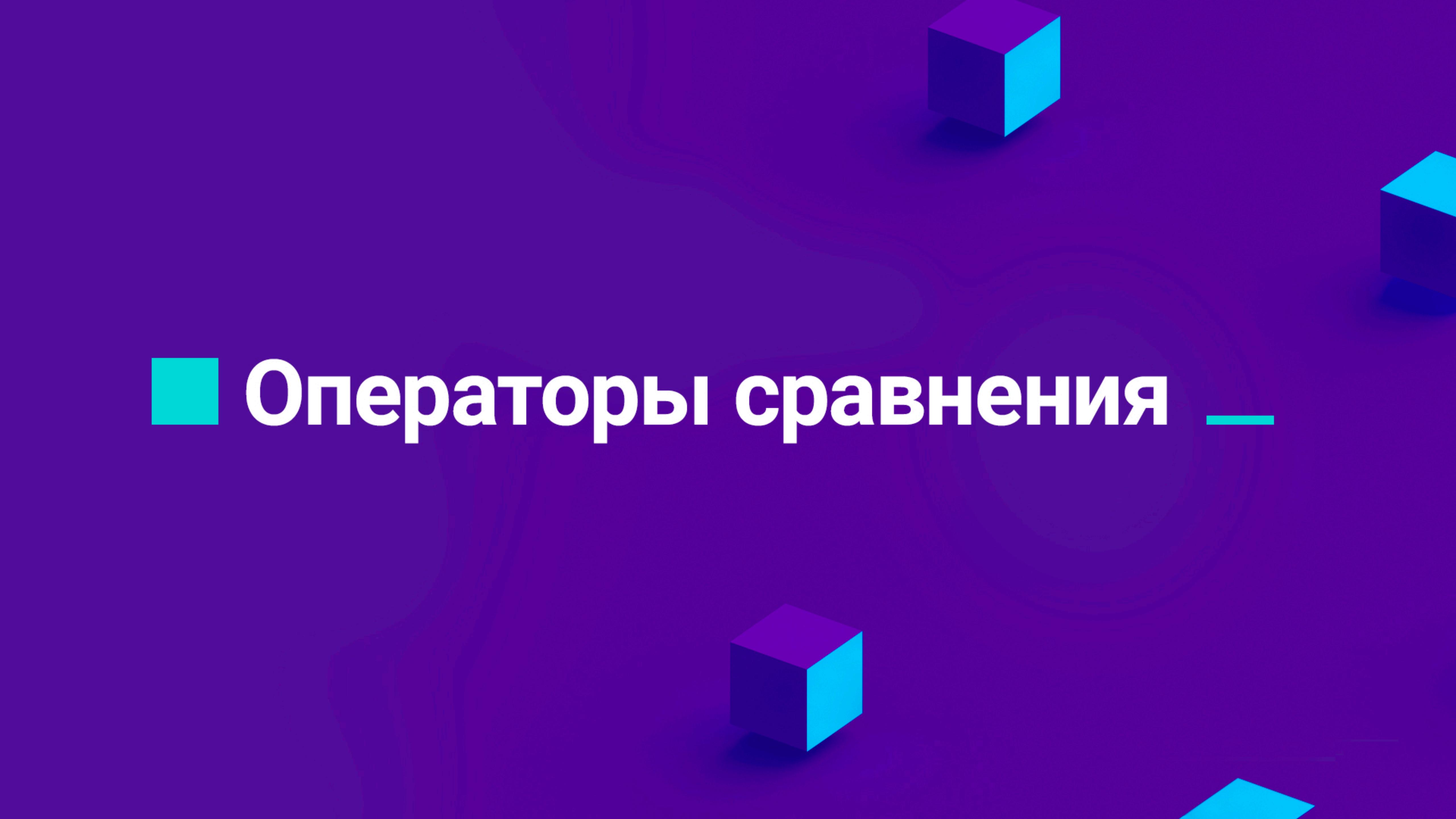
Оператор деления «/» используется когда необходимо вычислить частное двух operandов

\$a = \$b / \$c;

Оператор деления по модулю (%)

Оператор деления по модулю "%" используется, когда необходимо вычислить остаток от деления первого оперенда на второй

\$a = \$b % \$c;



■ Операторы сравнения —

Оператор равенства (==)

Оператор равенства сравнивает между собой значения двух operandов любого типа и возвращает логическое значение true – если operandы равны и false – если не равны. В случае если типы operandов не совпадают, PHP сделает неявное приведение к одному типу данных

`$a == $b;`

Оператор идентичности (==)

Оператор идентичности (оператор строгого равенства), как и оператор равенства, сравнивает два операнда, но при этом должны совпасть не только значения operandов, но и их типы. Т.е. значение true, при строгом сравнении, будет возвращено только тогда, когда оба операнда имеют одинаковые значения и относятся к одному и тому же типу данных.
Во всех остальных случаях будет возвращено значение false

\$a == \$b;

Оператор неравенства **(!= или <>)**

Оператор неравенства является противоположностью оператора равенства. При использовании он вернет значение true только в том случае, если значения сравниваемых operandов не равны

\$a != \$b; или \$a <> \$b;

Оператор неидентичности `(!=)`

Оператор неидентичности является противоположностью оператора идентичности. Т.е. значение `true` будет возвращено только тогда, когда оба операнда не равны друг другу или не относятся к одному и тому же типу данных

`$a != $b`

Оператор «строго больше» (>)

Данный оператор сравнивает два операнда и возвращает true, если значение операнда слева от оператора больше значения операнда справа, и false, если значение левого операнда меньше правого или значения равны

\$a > \$b;

Оператор «строго меньше» (<)

Оператор «строго меньше» сравнивает два операнда и возвращает true, когда значение слева от него меньше значения, находящегося справа, и false, если значение левого операнда больше правого или значения равны

\$a < \$b;

Оператор «больше или равно» (\geq)

Как понятно из названия, данный оператор добавляет возможность равенства к оператору «строго больше», рассматриваемому выше.

То есть, чтобы было возвращено значение `true`, значение слева от оператора должно быть больше или равно значению справа от него. Значение `false` возвращается, только если значение слева от операнда меньше, чем значение справа

`$a \geq $b;`

Оператор «меньше или равно» (`<=`)

Оператор «меньше или равно», так же как и оператор «больше или равно», расширяет рассматриваемый выше оператор «строго меньше».

При использовании данный оператор вернет значение `true`, если значение слева от него меньше или равно значению справа от него. Значение `false` будет возвращено только тогда, когда значение операнда слева больше, чем значение операнда справа

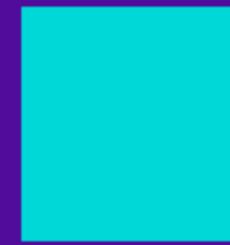
`$a <= $b;`

Оператор космический корабль (<=>)(php7)

Этот оператор предназначен для сравнения двух выражений.

Он возвращает -1, 0 или 1 если \$a, соответственно, меньше, равно или больше чем \$b. Сравнение производится в соответствии с правилами сравнения типов PHP

\$a < = > \$b;



Логические операторы —

Логическое И (&& или AND)

Сравнивает между собой два операнда и возвращает значение true тогда и только тогда, когда оба операнда имеют значение true. В случае, если хотя бы один или оба операнда возвращают false, то и оператор возвращает false

\$a	\$b	\$a && \$b
false	false	false
false	true	false
true	false	false
true	true	true

Логическое ИЛИ (|| или OR)

Является противоположностью логического «И» и возвращает значение true если один или оба операнда имеют истинное значение. В случае, если оба операнда возвращают false, то и оператор возвращает false

\$a	\$b	\$a \$b
false	false	false
false	true	true
true	false	true
true	true	true

Исключающее или (XOR)

Оператор возвращает значение true, если один и только один из operandов имеет значение true. В случае, когда оба операнда возвращают логическое true, оператор вернет значение false

\$a	\$b	\$a xor \$b
false	false	false
false	true	true
true	false	true
true	true	false

Логическое НЕ (!)

Оператор применяется для инверсии логического значения своего операнда и всегда возвращает true или false. Т.е. он возвращает противоположное логическое значение

\$a	! \$a
true	false
false	true

Побитовые операторы

- Побитовое И (&)
- Побитовое ИЛИ (|)
- Исключающее ИЛИ (^)
- Побитовое НЕ (~)



Сдвиг влево (<<)

Сдвигает все биты левого операнда влево на количество позиций, указанное в правом операнде, который должен быть целым числом.

Освободившиеся биты заполняются 0 битами.

Сдвиг значения влево на одну позицию эквивалентен умножению на 2, на две позиции – умножению на 4 и т.д.



Сдвиг вправо (>>)

Сдвигает все биты левого операнда вправо на количество позиций, указанное в правом операнде, который должен быть целым числом. Освободившиеся биты заполняются 0 битами.

Сдвиг значения вправо на одну позицию эквивалентен делению на 2, на две позиции – делению на 4 и т.д.



Операторы инкремента и декремента

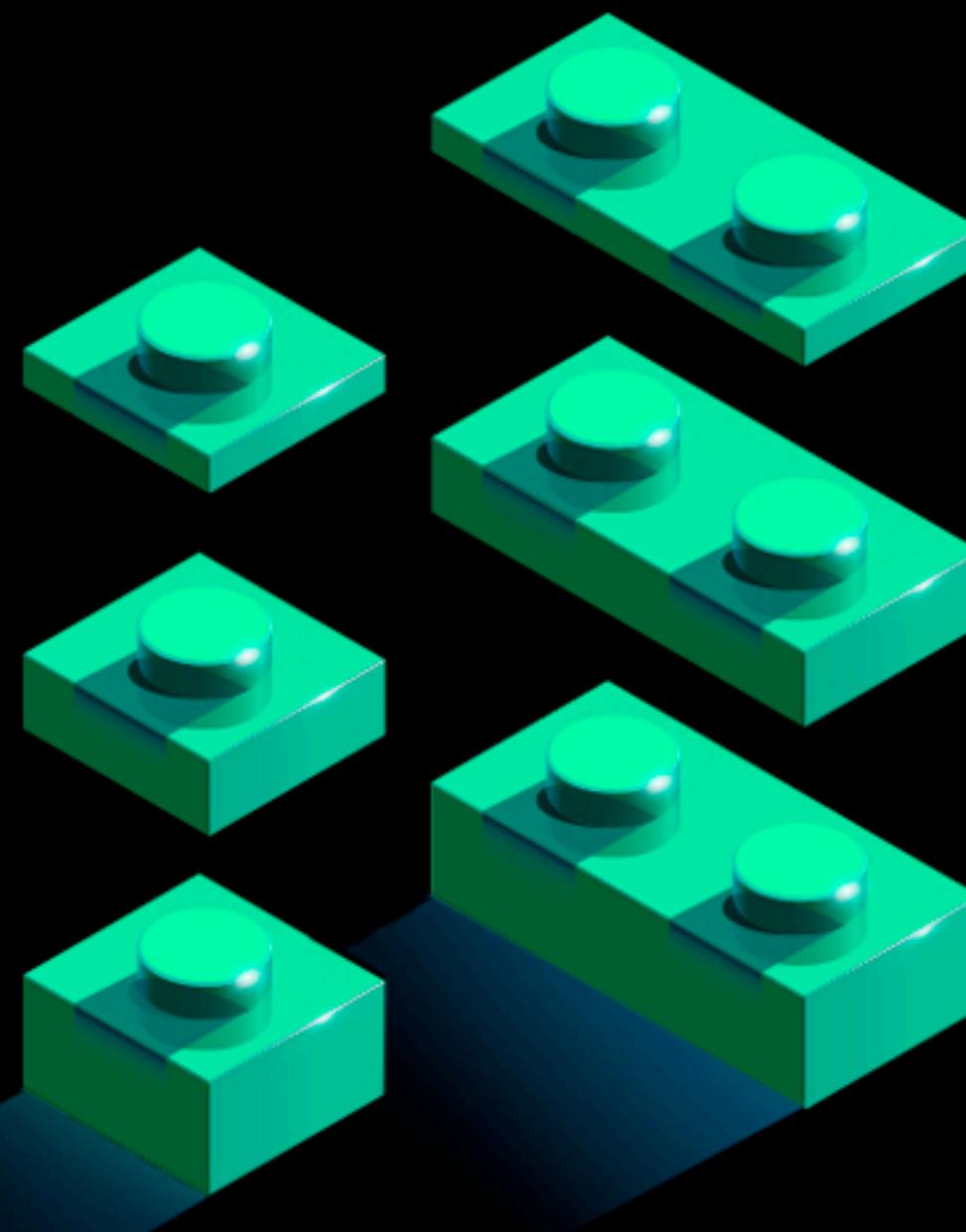
Зачастую при написании кода необходимо многократно увеличивать или уменьшать значение переменной на 1. Примером может служить счетчик итераций в циклах, о которых мы поговорим чуть позже

Для этого в PHP применяются операторы инкремента (увеличения на 1) обозначается как «`++`» и декремента (уменьшения на 1) обозначается как «`--`»

`$a--;` и `$a++;`

Строковые операторы

Единственным строковым оператором в PHP является «.» (точка) – оператор конкатенации. Он применяется, когда необходимо склеить два операнда в одну строку (такая операция называется конкатенацией) –





Операторы приведения типов

(int), (integer) – приведение к целому числу

(bool), (boolean) – приведение к булеву типу

(float), (double), (real) – приведение к числу с плавающей точкой (float)

(string) – приведение к строке

(array) – приведение к массиву

(object) – приведение к объекту

Для приведения типов, в PHP также используются функции:

(int), (integer) – приведение к целому числу

(bool), (boolean) – приведение к булеву типу

(float), (double), (real) – приведение к числу с плавающей точкой (float)

(string) – приведение к строке



Сокращенная форма операторов

- **`+=`** складывает значение переменной слева со значением справа, результат записывается в переменную слева
- **`-=`** вычитает из значения переменной слева значение справа, результат записывается в переменную слева
- **`*=`** умножает значение переменной слева на значение справа, результат записывается в переменную слева
- **`/=`** делит значение переменной слева на значение справа, результат записывается в переменную слева
- **`.=`** конкатенирует значение переменной слева со значением справа, результат приводится к строке и записывается в переменную слева
- **`%=`** вычисляет значение, равное делению по модулю переменной слева на значение справа и записывает результат в переменную слева

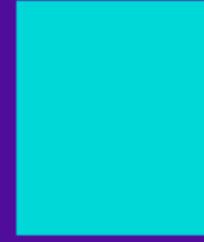


Тернарный оператор

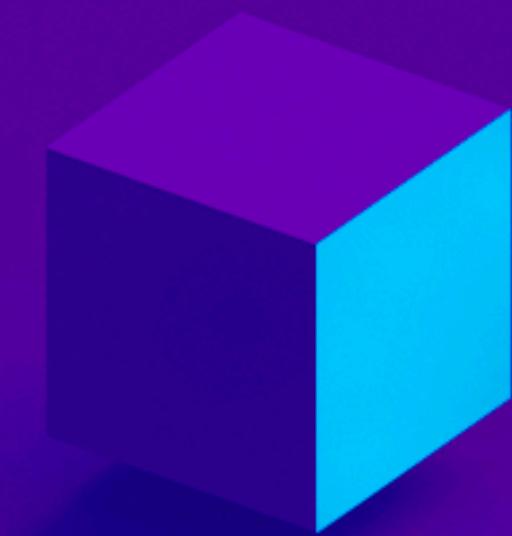
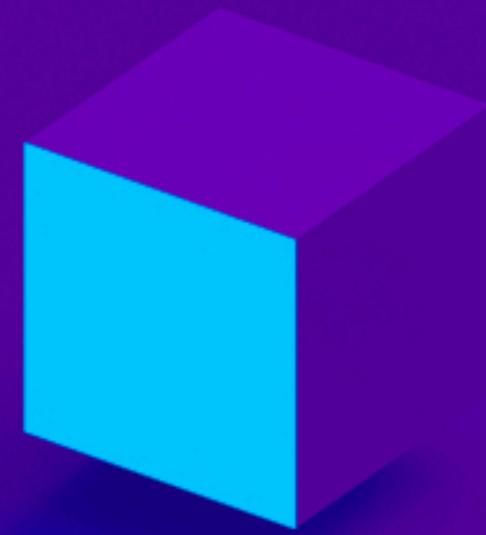
Тернарный оператор или оператор условия позволяет вернуть разные значения в зависимости от истинности или ложности условия.

Для этого сперва вычисляется условие, приводится к логическому значению типа boolean true или false, а затем выполняется одно из выражений. Если условие true, выполнится и вернется значение выражения «истина», а если условие false выполнится и вернется значение выражения «ложь»

\$a ? \$a : \$b;



Управляющие конструкции —

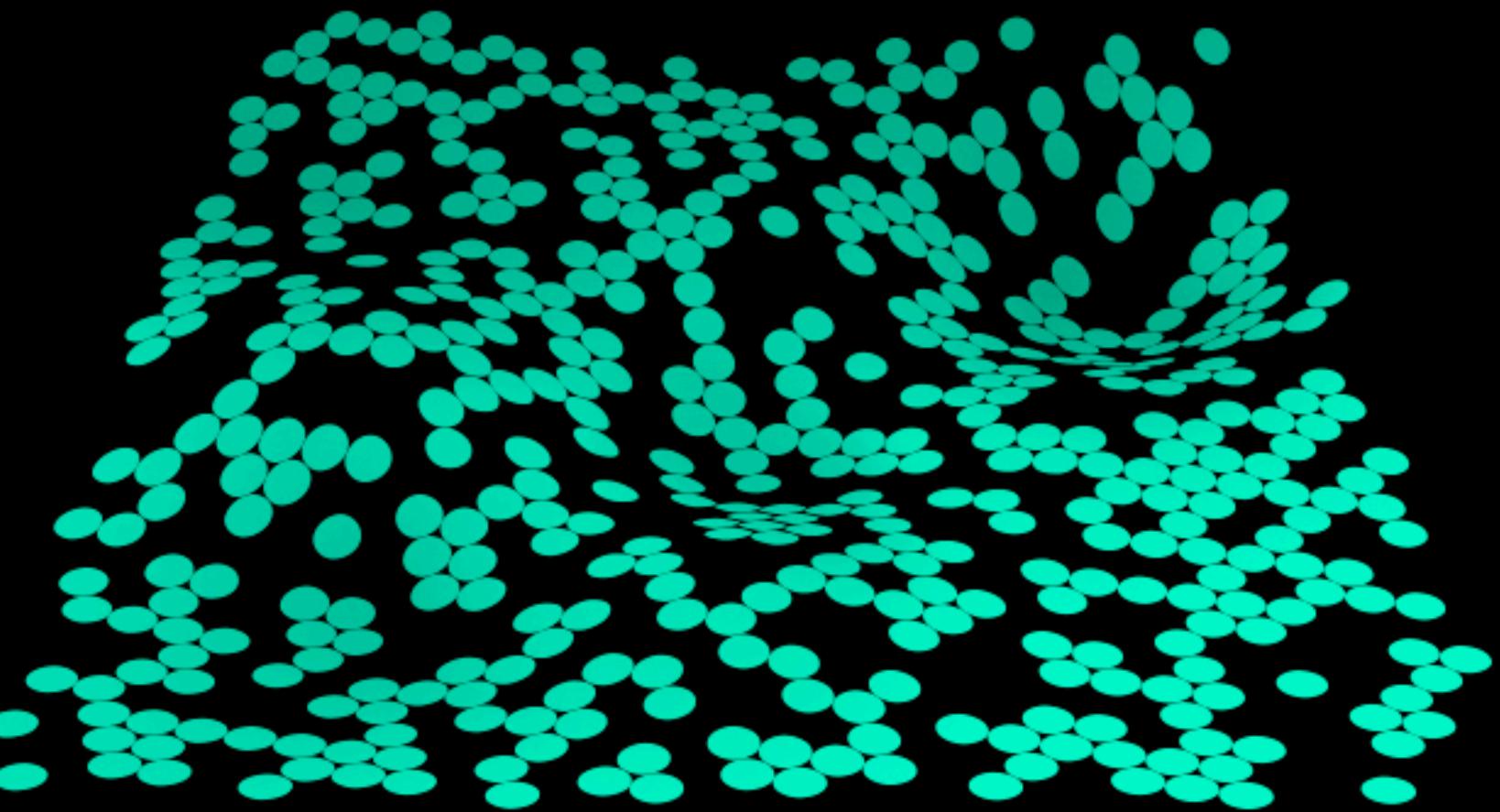


Условные конструкции

Управляющая конструкция if, которая проверяет истинность или ложность условия в круглых скобках, и, если условие истинно выполняет идущее следом выражение или блок выражений, а если условие ложно, пропускает это выражение или блок выражений —

Конструкции выбора

Управляющая конструкция `switch` предназначена для выбора действий, в зависимости от значения указанного выражения —



Сегодня мы узнали

- Какие бывают встроенные константы
- Какие существуют суперглобальные массивы
- Виды операторов и их предназначение
- Что такое управляющие конструкции, и какие функции они выполняют

Домашнее задание

1. Необходимо «оживить» форму авторизации пользователя в системе
2. В качестве логина должен использоваться только цифры в качестве пароля только буквы латинского алфавита. Сделать проверку на корректность ввода логина и пароля согласно заданным условиям
3. Для хранения пары логин пароль необходимо использовать массивы
4. Ролевая модель будет ограничена пользователем и администратором
5. При успешной авторизации обычного пользователя выводить сообщение – «Вы авторизовались, как пользователь», а под администратором – «Вы авторизовались, как администратор»