**PHP**

PHP — препроцессорный язык. Он может собирать страницы из кусочков, как конструктор. PHP превращает код собранных страниц в HTML. Это и есть препроцессинг.

**Комментарии**

Комментарий — пояснение на человеческом языке, которое оставляет программист для других разработчиков, чтобы пояснить код программы. Код, заключённый в комментарий, не выполняется и не оказывает влияния на результат.

Однострочный комментарий в PHP — две косых черты //. Код **на той же строке** после этих символов будет закомментированным. Чтобы убрать комментарий, надо удалить //.

// Я — текст в комментарии, я не влияю на программу,

// как и код ниже

// require('путь\_к\_файлу.php');

**Синтаксис**

Синтаксис — слова и правила написания этих слов. С помощью него компьютер нас понимает. У каждого языка программирования свой синтаксис.

**Команды**

Команда — указание программе выполнить какое-то действие.

Команда require добавляет код из указанного файла на страницу, где используется require:

require('путь\_до\_файла.php');

В PHP есть и другие команды для подключения файлов. Например, include. Подробнее об этой команде можно узнать в [спецификации](https://www.php.net/manual/ru/function.include.php).

Каждую команду нужно писать с новой строки, а в конце ставить точку с запятой ;. Так мы сообщим PHP, что запись одной команды закончена и дальше будет другая команда.

**Склеивание файлов**

При добавлении файлов в PHP происходит склеивание. Когда по ходу выполнения сценария встречается команда require, она заменяется на содержимое подключаемого файла, ровно в том порядке, в каком расположены require.

require('header.php');

require('content.php');

require('footer.php');

В примере к разметке шапки добавится разметка контента, а к ним разметка подвала.

**PHP-теги**

Чтобы PHP-код работал, он должен находиться внутри PHP-тегов:

<?php // Открывающий PHP-тег

// Какой-то PHP-код

?> // Закрывающий PHP-тег

Теги работают, как сигнал. С их помощью мы как будто говорим «обратите внимание, внутри PHP-код».

Иногда закрывающий тег можно опустить. Например, когда мы работаем со сценарием, где есть только PHP-код. Если мы внедряем фрагмент PHP в HTML, то закрывающий тег нужно использовать обязательно. Так мы обозначаем границы PHP-кода внутри шаблона.

**Переменные**

Переменная — способ сохранить информацию под определённым именем. **Объявление** переменной — написание имени переменной, первое её упоминание. Запись информации в переменную называется **присваиванием** значения.

$name = 'Семён';

// Объявили переменную $name

// Присвоили переменной значение 'Семён'

Имя переменной должно начинаться со знака доллара $, дальше может идти буква или символ подчёркивания. Название должно быть понятным для людей и описывать то, что находится в переменной.

**Шаблоны**

В шаблонах страницы хранят заготовки кода. Шаблон может быть статичным или динамичным, то есть включать в себя изменяемые данные.

**Добавление PHP в разметку**

// Полная запись

<p><?php echo($name); ?></p>

// Короткая запись

<p><?= $name ?></p>

Работают эти два варианта одинаково. Разница только в количестве символов.

**База данных**

База данных — сложная система, которая хранит информацию сайта в организованном виде.

**Работа с адресом**

Адресная строка — специальное поле в браузере, в котором написаны адреса страниц в интернете.

В адресе есть специальная часть, которая называется «параметры запроса». Параметры запроса располагаются после знака вопроса.

https://www.gloevk.ru/product.php?product\_id=1

С помощью команды $\_GET можно получить информацию из адреса.

Чтобы получить значение product\_id с помощью $\_GET, нужно написать $\_GET['product\_id'].