**Settype(… , “тип”)** – смена типа переменной

**Gettype($...)** – вывод типа

**Define(“…”, …)** – создание константных переменных

**Foreach($... as $... => $...) {…}** – создание цикла для массивов ( в первую вносим массив, во вторую ключ, а в третью значение ключа )

**Continue(2)** – пропуск родительського цикла ( 1 – этот цикл, 2 – родительський и т. д. )

**Continue(2)** – прерыв родительського цикла ( 1 – этот цикл, 2 – родительський и т. д. )

**List($... , $... , …..)** – этой фигне можно присвоить функцию, которая возвращает массив и элементы массива распределяться соответственно элементам списка

**Global $...** – в функции из локальной переменной делает глобальную

**Var\_damp($...)** – вывод всей информации об переменной

**get\_defined\_vars()** – вывод всех переменных и массивов файла ( возвращает массив всего, по этому для вывода нужна функция **print\_r()** )

**var\_dump(debug\_backtrace())** – вся информация про функцию, из которой вызываться

**empty($...)** – проверка на пустоту

**time()** – текущее время

**die(…);** - полное закрытие php кода и выход из страницы с текстом в душках

**json\_encode(…);** - массив PHP в JSON

**Строки**

**“{$...}…” “$... …”** – переменные в строках

**… .= …** - конкатенация

**strtolower($...)** – снятие загланых букв

**strtoupper($...)** – все буквы загланые

**ucfirst($...)** – первая буква загланая

**ucwords($...)** – все слова начинаются с заглавных

**strlen($...)** – длина строки

**strstr($..., «…»)** – вывод строки с указанного слова/символа

**str\_replace(“…”, “…”, $...)** – замена строки ( что, на что, где )

**trim(…)** – убирает пробелы сначала и в конце строи

**str\_repeat($..., …)** – повтор строки

**substr($..., …, …)** – обрезать строку

**trops($..., «…»)** – позиция слова/символа

**strchr($..., «…»)** – псевдоним функции strstr

**crypt($..., “…”)** – создание хеш функции ( сам сделай б\*\*, х\*\* сложного-то )

**preg\_match("…", $...)** – проверка регулярным выражением

**strpos("…", $...)** – возвращает позицию элемента из регулярного выражения, или false

**Math**

**Abs(…)** – модуль

**Pow(число, степень)** – число в степень

**Sqrt(…)** – корень

**Fmod(что, на что)** – остаток от деления ( % )

**Rand(от …, до …)** – рандомное число

**round(… , …)** – округлить до такого-то количества знаков после запятой

**ceil(…)** – округлить в большую сторону

**floor(…)** – округлить в меньшую сторону

**is\_int(…)** – целое?

**Is\_float(…)** – дробное?

**Is\_null($...)** – пустая ли переменная?

**Arrays**

**$... = array(… , … , …..) или $... = [… , … , …..]** – создание массива

**print\_r($...)** – возвращает весь масив

**count($...)** – количество

**max($...)** – максимальное

**min($...)** – минимальное

**sort($...)** – сортировка

**rsort($...)** – перестановка

**implode(«…»,$..)** – превратить в строку и разбить на “…”

**in\_array(…, $...)** – проверка на наличие в массиве

**isset($...)** – заполнена ли переменная

**unset($...[…])** – удаление ячейки, или затирание

**end($...)** – переместить курсор в конец

**reset($...)** – переместить курсор в начало

**next($...)** – переместить курсор вперед

**prev($...)** – переместить курсор назад

**current($...)** – вывести текущее положение курсора

**Objects**

**$... = array(“…” => … , “…” => … , …..) или $... = [“…” => … , “…” => … , …..]** – создание объекта

**URL запросы**

**$\_GET** – массив, содержащий ключи и их значения, переданные в url строке

**$\_POST** - массив, содержащий ключи и их значения, переданные в форме ( проверка на наявность через функию isset($\_POST[“…”]) )

**rawurlencode($…)** – для url типа строки кодировка текста ( в часть до знака вопроса )

**urlencode($...)** – для url типа строки кодировка текста ( в часть после знака вопроса )

**htmlspecialchars($...)** – автокодировка спецзнаков html (только 4 )

**htmlentities($...)**– автокодировка спецзнаков html (все)

**Header(“…”)** – задание значений заголовка

**header("Location: " . …)** – изменение директории ( url ) (после смены адреса нужно вызвать функцию

**exit;** которая е передаст лишних данных вродь )

( при передачи файла на сервер заголовок можно менять только с самого начала, перед передачей каких либо файлов в html, но можно подключить буферную страницу, в которой это можно делать в любой части кода, и по завершению ее работы скинуть все на сервак )

**ob\_start()** – ручное открытие буфера в коде ( можно задать автоматом в php.ini файле )

**ob\_end\_flush()** – ручное открытие буфера в коде

**headers\_list()** – возвращает массив значений заголовка

**Cookies & Sessions**

**$\_SESSION** - массив, содержащий cookie и их значения, переданные как обыкновенному массиву

SESSION имеют обработку на сервере, а не браузере пользователя и н имеют ограничений, в куках оно записываеться как ссылка на файл в срвре, не дая хакерам инфы

**$\_COOKIE** - массив, содержащий cookie и их значения, переданные в setcookie функции

COOKIE имеют обработку в браузере и могут быть вызваны тольк в самом начале работы кода, до первого пробела в html, они держат читаемую для всех информацию и их можно удалить вручную

**setcookie($name, $value, time)** – установка cookies ( имя, значение, время жизни ) ( для времени используют функцию time() + … для установки время жизни от момента записи, время в секундах )

**setcookie($name, null, time() - …)** – удаление cookie, можно просто задать имя или имя + null( но сказали лучше **имя, null, time() - 3600** – или любое другое число )

**session\_destroy();** - удалить все сессии

**session\_name()** – получить или установить имя текущей сессии

**Запросы**

**include("…")** – добавление файла php в другой код ( не выдаст ошибки при отсутствии части файла, например функции )

**require(“…”)** - добавление файла php в другой код (выдаст ошибки при отсутствии части файла, например функции )

**include\_once("…")** – добавить лишь раз ( при следующих вызовах игнорируется )

**require\_once(“…”)** – добавить лишь раз ( при следующих вызовах игнорируется )

**SQL**

**1.**

**$connection = mysqli\_connect("localhost", "имя пользователя", "пароль", "имя базы");** - соединение с базой данных ( в начале стр. ) ( в первом водиться тип сервера или id )

**2.**

**$... = "select \* …";** - создание переменной для хранения данных

**$... = mysqli\_query($connection, $...);** - функция mysql в первый параметр передаеться база данных, а во второй переменная с текстом кода

создание переменной которая хранит значение было ли успешное захватывание файлов из базы, если нет то через if и функцию die скинуть все

**3.**

**$... = mysqli\_fetch\_row($...)** – записывает в переменную первую или следующую строку из базы, при окончании запишет null, что выкинет из цикла while

**$... = mysqli\_fetch\_assoc($...)** – записывает в переменную первую или следующую ассоциативную (как массив но цельный ) строку из базы, при окончании запишет null, что выкинет из цикла while

**$... = mysqli\_fetch\_array($... , MYSQL\_NUM/MYSQL\_ASSOC/MYSQL\_BOTH)** – записывает в переменную первую или следующую сразу двумя типами, описанные выше, строку из базы, при окончании запишет null, что выкинет из цикла while

**В $**... храниться массив элементов

**4.**

**mysqli\_free\_result($...);** - возвращает обновленные данные в базу, переменная из функции mysqli\_query($connection, $...);

**5.**

**mysqli\_close($connection);** - закрытие базы данных

**mysqli\_connect\_errno($connection)** – возвращает код ошибки при подключении, можно использовать в if ибо при отсутствии ошибок вернет 0 и условие слетит

**mysqli\_connect\_error($connection)** – возвращает текст ошибки

**mysqli\_insert\_id($connection)** – возвращает последнее установленное id

**die(…);** - полное закрытие php кода и выход из страницы с текстом в душках

**mysqli\_affected\_rows($connection)** – количество измененных строк в sql

**mysqli\_real\_escape\_string($connection, $...)** – защита от sql injections, в строке можно передать незаметную команду (для взлома сайта), это функция экранирует все одинарные кавычки прочие спец символы, вторая переменная это та, которую нужно изменить и ей же присвоить результат

**mysqli\_num\_rows($result)** – возвращает количество рядков

**echo "selected", “checked”** - в формах через цикл и условие можно зафиксировать выбор

**strip\_tags($...)** – удаляет все html теги и знаки в тексте

**nl2br($...)** – включает переходы строк в тексте ( по умолчанию переходы не работают )

**Hashing**

**Md5(…)** – хеширование (старое )

**hash(‘…’, $...)**  - быстрое хеширование, не для паролей

**crypt($password, $salt)** – медленное хеширование для паролей (в соли больше 22 знаков)

**mt\_rand()** – рандомное значение, принимает 2 значения min и max

**uniqid($..., true)** - Сгенерировать уникальный ID, второй параметр, чтоб дольше было, генерация в зависимости от времени

**base64\_encode($...)** - Эта кодировка предназначена для корректной передачи бинарных данных по протоколам, не поддерживающим 8-битную передачу, например, для отправки тела письма. Данные, закодированные base64 занимают на 33% больше места по сравнению с оригинальными данными.