Тема 3.2
«Базовые функции PHP»

Вступление

В этой теме мы рассмотрим набор функций, которые «постоянно нужны» при разработке приложений на PHP.

Итак…

echo arg1, arg2, arg3 , …

Если быть точным, то echo – это конструкция языка, поэтому её «аргумент» можно не брать в скобки.

echo выводит свой аргумент (аргументы) в выходной поток.

```
$a=10;
echo $a;  // 10
echo $a,$a;  // 1010
```

print_r (mixed expression [, bool return])

Замечательная функция, широко используемая при отладке.

Выводит в выходной поток или возвращает (если второй аргумент равен TRUE) значение переменной любого типа, включая массивы, экземпляры классов и ресурсы.

```
$arr=array(10=> “B”, 20 => “C”);
print_r($arr);
```

```
Array
 (
  [10] => B
  [20] => C
)
```

var_dump (mixed expression [, …])

Ещё одна замечательная функция, широко используемая при отладке.

Выводит в выходной поток полную информацию о переменной любого типа, включая массивы, экземпляры классов и ресурсы.

```
$arr=array(10=> “B”, 20 => “C”);
var_dump($arr);
```

```
array(2)
{
 [10] => string(1) “B”
 [20] => string(1) “C”
}
```

isset ($var [, …])

Для проверки существования переменной (или элемента массива) применяется функция isset(), которая возвращает TRUE, если переменная или элемент массива существуют, и FALSE – в противном случае.

Пример:

```
$a=10;
$b=20;
$arr[12]=5;
if (isset($a)) echo “OK”; // OK
if (isset($a,$b)) echo “OK”; // OK
if (isset($arr[12])) echo “OK”; // OK
if (isset($c)) echo “OK”; // не существует
if (isset($arr[27])) echo “OK”; // не существует
```

unset ($var [, …])

Для удаления переменной (или элемента массива) применяется функция unset().

Пример:

```
$a=10;
$b=20;
$c=30;
$arr[12]=5;

unset($a);
unset($a,$b);
unset($arr[12]);
```

die([$message]) и exit([$message])

Иногда возникает необходимость немедленно прекратить выполнение программы на PHP. Это можно сделать с помощью функции die(), которая может принимать необязательный параметр, являющийся «предсмертным посланием», которое будет выведено в выходной поток перед завершением программы.

P.S. Функция exit() является синонимом функции die(), т.е. это, фактически, два названия одной и той же функции.

include ($file) и иже с ним

Для того, чтобы разместить код скрипта в различных файлах, а при выполнении программы «объединить» его в один скрипт, применяются следующие функции:

include(string $filename)
include_once(string $filename)
require(string $filename)
require_once(string $filename)

Отличие include от require состоит в том, что, в случае отсутствия «подключаемого файла» include генерирует предупреждение, а require – сообщение об ошибке и прекращает выполнение скрипта.

Отличие функций без _once с с ним состоит в том, что функции без _once могут «подключать» некоторый файл любое количество раз в один и тот же скрипт, а функции с _once «подключат» файл только ОДИН РАЗ, сколько бы раз они ни были вызваны с одним и тем же именем файла в качестве своего аргумента.

ИТОГ

Итак, основное, что нам потребуется:

echo – для вывода информации «на экран» (читается как «эхо» или «икоу», но НЕ как «эчо»);

print_r() и **var_dump()** – для отладки;

isset() и **unset()** для проверки существования и удаления переменных и элементов массива;

die() – для немедленного завершения программы;

include() и его «собратья» – для «сборки» скрипта из отдельных файлов.