

Разработка Web- приложений

Пользовательские функции

Функции в php

- Функцией называется фрагмент программного кода имеющий уникальное имя и предназначенный для решения конкретной задачи

Синтаксис определения функции

1. **function** **Имя_функции** (**параметр1**,
параметр2, ... **параметрN**)
2. { **Блок_действий**
3. **return** "**значение, возвращаемое**
функцией"; }

Пример

1. `function sum($a, $b)`
2. `{ $rez=$a+$b;`
3. `return $rez; }`

Особенности использования функций

- Если функция однажды определена в программе, то переопределить или удалить ее позже нельзя.
- Имена функций нечувствительны к регистру, но лучше вызывать функцию по тому же имени, каким она была задана в определении.

Особенности использования функций

- В функции можно определять значения аргументов, используемые по умолчанию. Само значение по умолчанию должно быть константным выражением

1. **function my_f(\$a, \$b, \$c="val")**

2. **{ \$rez=\$a.\$b.\$c;**

3. **return \$rez; }**

Особенности использования функций

- Если у функции несколько параметров, то те аргументы, для которых задаются значения по умолчанию, должны быть записаны после всех остальных аргументов в определении функции.
- В противном случае появится ошибка, если эти аргументы будут опущены при вызове функции.

Особенности использования функций

- Чтобы позволить функции изменять ее аргументы, их нужно передавать по ссылке.

1. **function add_label(&\$data_str)**

2. **{ \$data_str .= "checked"; }**

- *Задание 1*: написать функцию, которая меняет местами значения двух переменных.

Функции с переменным числом аргументов

- В *RНР4* можно создавать функции с переменным числом аргументов.
- Доступ к аргументам осуществляется с помощью *встроенных функций*
 - *func_num_args()*,
 - *func_get_arg()*,
 - *func_get_args()*.

Функции с переменным числом аргументов

1. `function DataCheck()`
2. `{ $n = func_num_args();`
3. `echo "Число аргументов функции $n"; }`

Функции с переменным числом аргументов

- Функция

func_get_arg (целое номер_аргумента)
возвращает аргумент из списка переданных в функцию аргументов, порядковый номер которого задан параметром номер_аргумента.

- Аргументы функции считаются начиная с нуля.

Функции с переменным числом аргументов

1. `function sum()`
2. `{ $n = func_num_args();`
3. `$rez=0;`
4. `If ($n==3) $rez=func_get_arg(0)+
func_get_arg(1)+ func_get_arg(2);`
5. `If ($n==2) $rez=func_get_arg(0)+
func_get_arg(1);`
6. `return $rez;}`

Функции с переменным числом аргументов

- *Задание 2:* написать функцию вычисляющую сумму произвольного числа введенных аргументов.
- *Задание 3:* написать функцию вычисляющую среднее арифметическое произвольного числа введенных аргументов.
- *Задание 4:* написать функцию вычисляющую сумму двух аргументов, если это целые числа и конкатенацию этих аргументов в противном случае.

Функции с переменным числом аргументов

- Функция ***func_get_args()*** возвращает массив, состоящий из списка аргументов, переданных функции.
- *Задание 5*: переписать функции из заданий 2 и 3 с использованием массивов и функции ***func_get_args()***.

Использование глобальных переменных внутри функции

- Чтобы использовать внутри функции переменные, заданные вне ее, эти переменные нужно объявить как глобальные. Для этого в *теле функции* следует перечислить их имена после ключевого слова *global*.

1. `$a=1;`
2. `function Test(){`
3. `global $a; $a = $a*2;`
4. `echo 'в результате работы функции $a=', $a; }`

Статические переменные

- Чтобы использовать переменные только внутри функции, при этом сохраняя их значения и после выхода из функции, нужно объявить эти переменные как статические.
- ***Статические переменные*** видны только внутри функции и не теряют своего значения, если выполнение программы выходит *за пределы функции*.
- Объявление таких переменных производится с помощью ключевого слова *static*

Статические переменные

1. `function Test(){`
2. `static $a = 1; // нельзя присваивать выражение или ссылку`
3. `$a = $a*2;`
4. `echo $a; }`

Статические переменные

- *Задание 6*: написать функцию, которая подсчитывает число её вызовов.