Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Веб-Технологии

ОТЧЕТ

по лабораторной работе №3

[«Специальные функции PHP»](http://heldmaschine.hol.es/)

Выполнил

студент: гр. 651001 Степанцов Е.В.

Проверил: Деменковец Д. В.

Минск 2018

# Теоретические сведения по теме лабораторной работы:

Функция это просто блок кода, который имеет название и входные параметры для обработки, который может быть выполнен, когда нам это нужно. Может показаться, что толку от них мало, но, поверьте мне, когда вы поймете и научитесь использовать функции, вы сможете сэкономить кучу времени и написать код, который будет гораздо более удобным для чтения!

Хотя тема функций часто воспринимается как сложная для начинающих программистов, но их в любом случае нужно знать.Возможно это займет много времени, но не сдавайтесь, ведь функции очень помогут вам в карьере программиста.

### Создаем свою первую php функцию

Когда вы создаете функцию, нужно сначала дать ей имя, например, myCompanyName. Лучше всего называть функции понятными названиями, которые будут иметь смысл, чтобы не запутаться. Это же касается и [переменных](http://myblaze.ru/urok-2-php-peremennyie/#_blank).

Синтаксис для создания функции довольно очевидный и логичный. Во-первых, вы должны сказать PHP, что вы хотите создать функцию. Это можно сделать, введя ключевое слово function и ваше имя функции и некоторые другие вещи (о которых мы поговорим чуть позже).

Вот как выглядит объявление функции myCompanyName:

<?php

function myCompanyName(){

}

?>

**Примечание**: Название функции может начинаться со знака «\_», но не с цифры!  
Это лишь каркас нашей функции, а внутри него мы можем поместить любой код, который захотим. При вызове функции он будет выполняться. Видите в фигурных скобках в приведенном выше примере «{}»? Эти скобки определяют, где должен быть наш код функции. Открывающая фигурная скобка «{» говорит PHP, что код функции начинается, а закрывающая фигурная «}» говорит PHP, что наша функция окончена!

Давайте добавим в нее что-то простенькое, например [echo](http://myblaze.ru/urok-3-php-echo/#_blank).

<?php

function myCompanyName(){

echo "Вы находитесь на блоге myblaze.ru <br />";

}

?>

Вот и все, довольно просто, правда? Теперь научимся вызывать нашу функцию.

### Вызов готовой функции php

Теперь, когда вы закончили кодирование простейшей функции, пришло время ее испытать. Ниже приведен простой скрипт. Давайте сделаем вот что: объявим свою функцию и используем ее.

function myCompanyName(){

echo "Вы находитесь на блоге myblaze.ru <br />";

}

echo "Привет всем! <br />";

myCompanyName();

echo "Рад вас видеть! <br />";

Хотя это был простой пример, важно понимать, что любая невнимательность может привести к ошибке. Когда вы создаете функцию, следуйте этим простым рекомендациям.

* Всегда начинайте ваши функции с ключевым словом function.
* Помните, что ваш код функции должны быть между «{» и «}».
* Когда вы вызываете функцию, убедитесь, что пишете ее имя правильно.

### Функции с параметрами

Еще одна полезная возможность состоит в том, что функции вы можете вызывать с параметрами, то есть передавать внутрь ее кода дополнительную информацию. Наша первая функция myCompanyName не очень-то и полезна, потому что все что она умеет, это печатать одну и ту же строку.

Однако, если бы мы использовали параметры, то мы могли бы добавить некоторые дополнительные фишки в нашу функцию! Параметр появляется скобках «()» и выглядит как обычная переменная PHP. Давайте создадим новую функцию, которая создает пользовательское приветствие основанное на имени человека, которое мы передадим в качестве параметра.

<?php

function sayHello($Name){

echo "Hello there ". $Name . "!<br />";

}

?>

Теперь когда мы будем вызывать функцию sayHello, то в скобках обязательно должны указывать параметр (имя), так как функция объявлена с параметром и игнорировать его нельзя.  
Если мы хотим указывать несколько параметров, то просто перечисляем их при объявлении функции через запятую и все, ничего больше не требуется.

### Возвращение значения функцией php

Помимо того, что вы имеете возможность передать функции информацию, она так же может и сама вернуть значение. Но помните, функция может возвращать только одно значение, хотя это может быть что угодно: целое число, вещественное, булевое значение, как вы пожелаете!

Как она возвращает значение? Лучше всего разобраться на примере. Объявим функцию с параметрами, а затем присвоим результат некоторой переменной, вот так:

<?php

function mySum($numX, $numY){

$total = $numX + $numY;

return $total;

}

$myNumber = 0;

echo "Перед использованием функции, myNumber = ". $myNumber ."<br />";

$myNumber = mySum(3, 4); // Сохраняем результат вызова функции mySum в переменную $myNumber

echo "После вызова функции, myNumber = " . $myNumber ."<br />";

?>

Чтобы присвоить функции значение — в нее добавляется оператор return, видите его? Результат сложения двух чисел хранится в переменной $total, а с помощью return $total это значение передается самой функции. Затем мы просто присваиваем его переменной $myNumber после вызова функции mySum с двумя параметрами 3 и 4 (это числа которые следует сложить).

2. **Цель работы:** изучение функций, определяемых пользователем, функций по работе с датой и временем, функций по работе с файловой системой языка программирования PHP**.**

**Стандартное задание:**

**Вариант 8:** написать функцию, преобразующую число в словесную форму записи (например, 127 преобразуется в "сто двадцать семь"). Число (до 20 разрядов) получать через веб-форму.

**Расширенное задание**

Написать часть системы управления сайтом, отвечающую за добавление, удаление и перемещение файлов. Фактически требуется реализовать примитивный веб-ориентированный файловый менеджер для управления файловой системой на стороне сервера.

Дополнительными возможностями такого файлового менеджера может быть управление каталогами, редактирование текстовых файлов, конвертация графических файлов, шифрование и дешифрование файлов, создание и распаковка архивов и тому подобное.

**Стандартное задание**

<?php

$Arr\_numGroups\_rus = array("тысяч", "миллион", "миллиард", "триллион", "квадриллион", "квинтиллион");

$Arr\_hungredsName\_rus = array("", "сто","двести","триста","четыреста","пятьсот","шестьсот","семьсот","восемьсот","девятьсот");

$Arr\_betw\_11\_19\_rus = array('10' => "десять", "одиннадцать", "двенадцать","тринадцать", "четырнадцать", "пятнадцать", "шестнадцать", "семнадцать", "восемнадцать", "девятнадцать");

$Arr\_numeral\_rus = array("","один","два","три","четыре","пять","шесть","семь","восемь", "девять");

$Arr\_numeral\_1000\_rus = array("","одна","две","три","четыре","пять","шесть","семь","восемь", "девять");

$Arr\_Decimals\_rus = array('2' => "двадцать","тридцать","сорок","пятьдесят","шестьдесят","семьдесят","восемьдесят", "девяносто");

$Arr\_suffix\_rus = array('10^3' => array("а", "и", ""), '>10^3' => array("", "а", "ов") );

$Arr\_numGroups\_eng = array("thousand","million","billion","trillion","quadrillion","quintillion");

$Arr\_betw\_11\_19\_eng = array('10' => "ten","eleven","twelve","thirteen","fourteen","fifteen","sixteen","seventeen","eighteen", " nineteen");

$Arr\_numeral\_eng = array("","one","two","three","four","five","six","seven","eight","nine");

$Arr\_Decimals\_eng = array('2' => "twenty","thirty","fourty","fifty","sixty","seventy","eighty","ninety");

$Arr\_hungredsName\_eng = array("", "one hundred","two hundred","three hundred","four hundred","five hundred","six hundred","seven hundred","eight hundred","nine hundred");

function processHungreds($num, $Arr\_hungredsName, $Arr\_betw\_11\_19, $Arr\_Decimals, $Arr\_numeral)

{

$str = "";

$str .= " " . $Arr\_hungredsName{$num / 100};

$temp = $num % 100;

if (($temp >= 10) && ($temp <= 19))

{

$str .= " " .$Arr\_betw\_11\_19{$temp};

} else

{

$str .= " " . $Arr\_Decimals{$temp / 10};

$str .= " " . $Arr\_numeral{$temp % 10};

}

return $str;

}

function getSuffix($num ,$suffix)

{

if(($num % 100 <= 11) || ($num % 100 >= 19))

{

if ($num % 10 == 1) { return $suffix{'0'}; }

if (($num % 10 >= 2) && ($num % 10 <= 4)) {return $suffix{'1'};}

return $suffix{'2'};

}

return $suffix{'2'};

}

function getStringValueOfNum($NUM, $Arr\_hungredsName,$Arr\_betw\_11\_19,$Arr\_Decimals,$Arr\_numeral,$Arr\_numeral\_1000, $Arr\_numGroups, $Arr\_suffix, $zeroVal)

{

$countEl = 0;

$i=-1;

if ($NUM == 0) { return $zeroVal;}

while ($NUM > 0)

{

$str = "";

$temp = $NUM % 1000;

if (($i == 0) && ($Arr\_betw\_11\_19{'10'} == "десять")) { $temp\_forNum = processHungreds($temp, $Arr\_hungredsName, $Arr\_betw\_11\_19, $Arr\_Decimals, $Arr\_numeral\_1000); }

else { $temp\_forNum = processHungreds($temp, $Arr\_hungredsName, $Arr\_betw\_11\_19, $Arr\_Decimals, $Arr\_numeral); }

$str .= $temp\_forNum;

if ($i >= 0)

{

if ($temp\_forNum != " ") { $str .= " " . $Arr\_numGroups{$i}; }

if (($i == 0) && ($Arr\_betw\_11\_19{'10'} == "десять") && ($temp\_forNum != " ")) { $str .= getSuffix($temp, $Arr\_suffix{'10^3'}); }

if (($i > 0 ) && ($Arr\_betw\_11\_19{'10'} == "десять") && ($temp\_forNum != " ")) { $str .= getSuffix($temp, $Arr\_suffix{'>10^3'});}

}

$Temp\_Arr[$countEl++] = $str;

$NUM = intdiv($NUM, 1000);

$i++;

}

$i = $countEl-1;

$str = "";

while ($i >= 0) {

$str .= $Temp\_Arr{$i};

$i--;

}

return $str;

}

if (isset($\_POST['inputArea']))

{

$NUM = $\_POST['inputArea'];

if (is\_numeric($NUM))

{

echo "<div style='background-color:black; color:white;'> $NUM: " . getStringValueOfNum($NUM, $Arr\_hungredsName\_rus, $Arr\_betw\_11\_19\_rus, $Arr\_Decimals\_rus,

$Arr\_numeral\_rus, $Arr\_numeral\_1000\_rus, $Arr\_numGroups\_rus, $Arr\_suffix\_rus, 'ноль') . "</div>";

echo "<div style='background-color:black; color:white;'> $NUM:" . getStringValueOfNum($NUM, $Arr\_hungredsName\_eng, $Arr\_betw\_11\_19\_eng, $Arr\_Decimals\_eng,

$Arr\_numeral\_eng, 0, $Arr\_numGroups\_eng, 0, 'zero') . "</div>";

} else {echo "ошибки ввода";}

include "news.html";

}