Динамическое формирование контента и меню. Таблица умножения.

Лабораторная работа № А-5.

ЦЕЛЬ РАБОТЫ

Освоение навыков динамического формирования контента в зависимости от набора параметров. Закрепление знаний по основам программирования простейших алгоритмов на PHP. Получение навыков использования пользовательских функций.

ПРОДОЛЖИТЕЛЬНОСТЬ

6 академических часов (3 занятия)

РЕЗУЛЬТАТ РАБОТЫ

Размещенный на Веб-сервере и доступный по протоколу http документ (страница сайта) с динамически формируемой таблицей умножения.

ДОПОЛНИТЕЛЬНЫЕ ТРАБОВАНИЯ К РАБОТЕ

Работа оформляется в виде одного HTML-документа с интегрированным PHP-кодом. При открытии страницы в браузере должен отображаться следующий контент.

1. Главное меню (в шапке страницы) из двух пунктов: "Табличная верстка"; "Блочная верстка".
   - Оба пункта меню реализуются с помощью тега `<a>` и передают один и тот-же GET-параметр, значение которого определяет тип HTML-верстки таблицы умножения.
   - По умолчанию (при первой загрузке, если параметр не выбран) используется табличная верстка.
   - Ни один из пунктов меню по умолчанию (при первой загрузке) не выделен!
   - При переходе по ссылке этот пункт меню выделяется каким-либо образом.
   - Внешний вид страницы при обоих типах верстки должен совпадать.
   - При изменении типа верстки содержание таблицы умножения не должно изменяться.
2. Основное меню (в левой части страницы) с пунктами: Всё, 2, 3, 4, 5, 6, 7, 8, 9.
   - При выборе первого пункта (выделен по умолчанию при первой загрузке) выводится вся таблица умножения: восемь колонок.
   - При выборе пункта с цифрой – выводится таблица умножения на соответствующую цифру (более крупно).
   - При выборе любого пункта меню – этот пункт выделяется любым способом.
   - При изменении содержания таблицы умножения тип верстки не должен изменяться.
3. Информация о содержании страницы (в подвале).
   - тип верстки;
   - название таблицы умножения (полная или одна колонка);
   - дата и время.
4. Таблица умножения (в основной части страницы).
   - Тип верстки и содержание определяется заданными параметрами (или их отсутствием).
   - Все цифры (и только цифры) должны быть ссылками на соответствующие таблицы умножения. Например, строка 2x3=6 должна содержать три ссылки: на таблицы умножения на 2, 3 и 6. Строка 3x5=15 – две ссылки: на 3 и 5. Указанные ссылки всегда «сбрасывают» тип верстки – соответствующий параметр не указывается.

РЕКОМЕНДАЦИИ К СТРУКТУРЕ ПРОГРАММЫ

Программу (скрипт) можно разделить на четыре основных блока:

1. главное меню (вверху, определяет тип верстки);
2. основное меню (сбоку, определяет содержание таблицы);
3. таблица умножения;
4. информация о таблице умножения.

Как и любой другой PHP-скрипт, программа должна в зависимости от переданных в нее параметров выполнять какие-либо действия, выводить какую-либо информацию. Исходя из условий данной лабораторной работы, необходимо и достаточно два таких параметра: *html_type* (тип верстки) и *content* (содержание таблицы умножения). В зависимости от их значения каждый блок принимает тот или иной вид. Сочетание параметров влияет на внешний вид каждого блока. Составим таблицы, в которой укажем все сочетания параметров и вид основных блоков. Блоки рекомендуется реализовывать последовательно, в указанном выше порядке.

ГЛАВНОЕ МЕНЮ

	html_type n/a	**html_type = TABLE**	**html_type = DIV**
content n/a	Оба пункта меню не выделены	Пункт «**Табличная верстка**» выделен. Пункт «**Блочная верстка**» – НЕ выделен.	Пункт «**Табличная верстка**» НЕ выделен. Пункт «**Блочная верстка**» – выделен.
content=2			
content=3			
content=4			
content=5			
content=6			
content=7			
content=8			
content=9			

Из таблицы видно, что на главное меню оказывает влияние только один параметр. Тогда реализация блока «Главное меню» может быть представлена следующим кодом.

Листинг А-5. 1

```
<div id=”main_menu”><?php
	echo '<a href=”?html_type=TABLE”'; // начало ссылки ТАБЛИЧНАЯ ФОРМА

	// если в скрипт был передан параметр html_type и параметр равен TABLE
	if( array_key_exists('html_type', $_GET) && $_GET['html_type']== 'TABLE' )
		echo ' class=”selected”'; // ссылка выделяется через CSS-класс
	echo '>Табличная форма</a>'; // конец ссылки ТАБЛИЧНАЯ ФОРМА

	echo '<a href=”?html_type=DIV”'; // начало ссылки БЛОКОВАЯ ФОРМА

	// если в скрипт был передан параметр html_type и параметр равен DIV
	if( array_key_exists('html_type', $_GET) && $_GET['html_type']== 'DIV' )
		echo ' class=”selected”'; // ссылка выделяется через CSS-класс

	echo '>Табличная форма</a>'; // конец ссылки БЛОКОВАЯ ФОРМА
?></div>
```

Для оформления главного меню в соответствии с дизайном используется тег `<div>` – т.к. он не изменяется в зависимости от переданных параметров, его вывод осуществляется не PHP-скриптом, а статически из HTML-кода страницы. В программе осуществляется вывод двух пунктов меню в виде ссылок. Их оформление в форме кнопок или иной внешний вид осуществляется CSS. Вывод производится в три этапа.

1. Выводится начало HTML-кода ссылки (вывод не зависит от переданного в скрипт параметра *html_type*). Здесь полностью указан адрес ссылки, в котором закодирован передаваемый параметр *html_type* и его значение.
2. Выводится выделение ссылки, если это необходимо. Для этого проверяется наличие имени параметра *html_type* в списке переданных программе методом GET параметров (массив `$_GET`) и его соответствие требуемому значению. Обратите внимание – параметр, который передает ссылка, и параметра от которого зависит ее внешний вид, совпадает. Действительно – внешний вид пункта меню зависит от того параметра, который он и передает (т.е. признаком необходимости выделения пункта меню является его нажатие, а значит передача указанного в нем параметра скрипту).
3. Выводится окончание ссылки с его текстом. Этот этап также не зависит от переданных в программу параметров.

ОСНОВНОЕ МЕНЮ

	html_type n/a	**html_type = TABLE**	**html_type = DIV**
content n/a	Выделен пункт меню «Вся таблица умножения»		
content=2	Выделен пункт меню «Таблица умножения на 2»		
content=3	Выделен пункт меню «Таблица умножения на 3»		
content=4	Выделен пункт меню «Таблица умножения на 4»		
content=5	Выделен пункт меню «Таблица умножения на 5»		
content=6	Выделен пункт меню «Таблица умножения на 6»		
content=7	Выделен пункт меню «Таблица умножения на 7»		
content=8	Выделен пункт меню «Таблица умножения на 8»		
content=9	Выделен пункт меню «Таблица умножения на 9»		

Принцип работы программы для основного меню аналогичен блоку главного меню. Однако в его реализации, в силу его большого размера (а на практике возможны случаи меню из нескольких сотен пунктов, описание которого будет чрезвычайно трудоемко) предполагает использование цикла.

Листинг А-5. 2

```
<div id="product_menu"><?php

      echo '<a href="/"'; // начало ссылки ВСЯ ТАБЛИЦА УМНОЖНЕНИЯ
      if( !isset($_GET['content']) ) // если в скрипт НЕ был передан параметр content
            echo ' class="selected"'; // ссылка выделяется через CSS-класс
      echo '>Вся таблица умножения</a>'; // конец ссылки

      for( $i=2; $i<=9; $i++ ) // цикл со счетчиком от 2 до 9 включительно
      {
            echo '<a href="?content='.$i. '" '; // начало ссылки
            // если в скрипт был передан параметр content
            // и параметр равен значению счетчика
            if( isset($_GET['content']) && $_GET['content']==$i )
                  echo ' class="selected"'; // ссылка выделяется через CSS-класс
            echo '>Таблица умножения на '.$i.'</a>'; // конец ссылки
}
?></div>
```

Первая ссылка выводится вне цикла, т.к. она не передает никакой параметр и признаком ее выделения служит именно отсутствие переданного параметра *content* (функции `array_key_exists('content',$_GET)` и `isset($_GET['content'])` в данном случае эквивалентны).

Цикл последовательно совершает восемь итераций, выводя восемь однотипных ссылок. При этом значение передаваемого в ссылке параметра *content* и ее текст определяются номером итерации (значением переменной `$i`). Фактически такое использование циклов позволит вывести сколько угодно ссылок – иначе пришлось бы копировать и исправлять код для каждой из них отдельно,

что неизбежно привело бы к увеличению объема кода, ухудшению читаемости, усложнению внесения изменений и росту вероятности ошибок.

На каждой итерации происходит:

- вывод адрема ссылки (со значением параметра *content* равному номеру итерации `$i`);
- проверка передачи в программу параметра *content* и равенства его значения текущей итерации (т.е. проверка, не была ли эта ссылка нажата для перехода на эту страницу) с выделением ссылки при ее прохождении;
- вывод текста ссылки (с использованием номера итерации).

СОПРЯЖЕНИЕ ГЛАВНОГО И ОСНОВНОГО МЕНЮ

Реализация приведенного выше кода главного и основного меню не предполагает сохранение выделенных пунктов меню при переходе по ссылке другого меню. Действительно, если мы перешли по ссылке «Таблица умножения на 4» и нажали на пункт «Блочная верстка», то выделение пункта меню «Таблица умножения на 4» пропадет. А это не соответствует условиям лабораторной работы.

Для решения задачи сопряжения меню и сохранения выделенных пунктов добавим в код ссылок второй параметр. Т.е. ссылки главного меню должны передавать в программу не только параметр *html_type*, но и текущее значение параметра *content* (при условии его существования). Ссылки основного меню – аналогично. Поэтому PHP-код необходимо модифицировать следующим образом (для примера возьмем ссылку главного меню «Табличная форма»).

Листинг А-5. 3

```
echo '<a href="?html_type=TABLE'; // начало ссылки ТАБЛИЧНАЯ ФОРМА

if( !isset($_GET['content']) ) // если параметр content был передан в скрипт
      echo '&content='.$_GET['content']; // добавляем в ссылку второй параметр

echo '"'; // завершаем формирование адреса ссылки и закрываем кавычку

// если в скрипт был передан параметр html_type и параметр равен TABLE
if( array_key_exists('html_type', $_GET) && $_GET['html_type']== 'TABLE' )
      echo ' class="selected"'; // ссылка выделяется через CSS-класс

echo '>Табличная форма</a>'; // конец ссылки ТАБЛИЧНАЯ ФОРМА
```

Начало ссылки слегка отличается от исходного варианта – нет закрывающей ссылку кавычки. Ее нельзя использовать сразу т.к. пока не известно закончено формирование ссылки, или же в ней надо передать второй параметр. Проверка этого осуществляется на второй строке: если второй параметр был передан в скрипт ранее, то его необходимо сохранить и добавить в формируемую ссылку. И только после этого осуществляется вывод закрывающей кавычки. Таким образом, хотя внешний вид меню зависит только от одного параметра – его HTML-код, а именно код ссылок, зависит от обоих параметров!

Модификация остальных пунктов главного и основного меню осуществляется аналогичным образом в соответствии с логикой работы программы и здравым смыслом.

ТАБЛИЦА УМНОЖЕНИЯ

	html_type n/a	**html_type = TABLE**	**html_type = DIV**
content n/a	Верстка табличная. Вся таблица умножения.		Верстка блочная. Вся таблица умножения.
content=2	Верстка табличная. Таблица умножения на 2		Верстка блочная. Таблица умножения на 2.
content=3	Верстка табличная. Таблица умножения на 3		Верстка блочная. Таблица умножения на 3.

content=4	Верстка табличная. Таблица умножения на 4	Верстка блочная. Таблица умножения на 4.
content=5	Верстка табличная. Таблица умножения на 5	Верстка блочная. Таблица умножения на 5.
content=6	Верстка табличная. Таблица умножения на 6	Верстка блочная. Таблица умножения на 6.
content=7	Верстка табличная. Таблица умножения на 7	Верстка блочная. Таблица умножения на 7.
content=8	Верстка табличная. Таблица умножения на 8	Верстка блочная. Таблица умножения на 8.
content=9	Верстка табличная. Таблица умножения на 9	Верстка блочная. Таблица умножения на 9.

В отличие от любого из меню на блок таблицы умножения влияют оба параметра. Первый определяет тип верстки, второй – выводить ли всю таблицу или же только один ее столбец. Для реализации программы блока удобно использовать функции – это упрощает структуру программы, уменьшает размер кода, последующее внесение изменений и вероятность ошибок. Определим в программе две функции.

Листинг А-5. 4

```
// функция ВЫВОДИТ ТАБЛИЦУ УМНОЖЕНИЯ В ТАБЛИЧНОЙ ФОРМЕ
function outTableForm()
{
	// код тела функции
}

// функция ВЫВОДИТ ТАБЛИЦУ УМНОЖЕНИЯ В БЛОЧНОЙ ФОРМЕ
function outDivForm ()
{
	// код тела функции
}
```

Первая функция `outTableForm()` – осуществляет всю работу по выводу таблицы умножения в табличной форме, вторая функция `outDivForm()` – в блочной форме. Тогда для обработки первого параметра достаточно следующего кода.

Листинг А-5. 5

```
if (!isset($_GET['html_type']) && $_GET['html_type']== 'TABLE' )
	outTableForm();	// выводим таблицу умножения в табличной форме
else
	outDivForm();  // выводим таблицу умножения в блочной форме
```

В нем проверяется существование параметра *html_type* и его значение. Если он не существует (вывод по умолчанию) или его значение равно *TABLE* – вызывается функция, выводящая таблицу умножения в табличной форме, иначе – в блоковой.

Для примера рассмотрим работу функции `outDivForm()`, функция `outTableForm()` строится аналогичным образом, за исключением выводимых HTML-тегов для верстки результатов в виде таблицы.

Листинг А-5. 6

```
function outDivForm ()
{
	// если параметр content не был передан в программу
	if( !isset($_GET['content']) )
```

```
	{
		for($i=2; $i<10; $i++) // цикл со счетчиком от 2 до 9
		{
			echo '<div class="ttRow">'; // оформляем таблицу в блочной форме
			outRow( $i ); // вызывем функцию, формирующую содержание
				// столбца умножения на $i (на 4, если $i==4)
			echo '</div>';
		}
	}
	else
	{
		echo '<div class="ttSingleRow">'; // оформляем таблицу в блочной форме
		outRow( $_GET['content'] ); // выводим выбранный в меню столбец
		echo '</div>';
	}
}
```

Функция работает следующим образом. Если параметр *content* не был передан в программу, то это означает вывод всей таблицы умножения. Для этого последовательно в цикле восемь раз вызывается функция `outRow()` с параметром, соответствующим номеру итерации. Эта функция полностью формирует содержание указанного столбца таблицы умножения – например, если передан параметр 4, то и сформируется столбец таблицы умножения на 4. Поэтому, вызов этих функций в цикле и соответствующее «обрамление» их HTML-тегами формирует всю таблицу умножения.

Если же параметр *content* указан, то выводимый столбец хранится в нем. Тогда, достаточно вызвать описанную функцию `outRow()` только один раз, передав ей параметр *content*. Обратите внимание – внешний вид блока полной таблицы умножения и одиночного блока может отличаться за счет использования разных CSS-классов: `ttRow` и `ttSingleRow`.

Листинг А-5. 7

```
// функция ВЫВОДИТ СТОЛБЕЦ ТАБЛИЦЫ УМНОЖЕНИЯ
function outRow ( $n )
{
	for($i=2; $i<=9; $i++)		// цикл со счетчиком от 2 до 9
		echo $n.'x'.$i.'='.($i*$n).'<br>';	// выводим строку столбца с тегом
}
```

В теле функции `outRow()` также удобно использовать цикл. С его помощью за восемь итерации выводятся восемь строк требуемого столбца. Обратите внимание – арифметические операции в операторе `echo` рекомендуется заключать в скобки (во избежание путаницы при преобразовании типов переменных)! Небольшая модификация функции позволит выполнить и последнее требование в лабораторной работе – формирование для всех чисел в таблице ссылок.

Листинг А-5. 8

```
// функция ВЫВОДИТ СТОЛБЕЦ ТАБЛИЦЫ УМНОЖЕНИЯ
function outRow ( $n )
{
	for($i=2; $i<=9; $i++) // цикл со счетчиком от 2 до 9.
		echo outNumAsLink($n).'x'.outNumAsLink($i).'='.
			outNumAsLink($i*$n).'<br>';
}
```

Функция `outNumAsLink()` преобразует число в соответствующую ему ссылку (если это возможно) и возвращает ее. Обратите внимание – функция ничего не выводит, а лишь возвращает значение! Поэтому ее код, заданный в начале описания данной лабораторной работы, необходимо самостоятельно модифицировать соответствующим образом.

ИНФОРМАЦИЯ О ТАБЛИЦЕ УМНОЖЕНИЯ

Информация о таблице умножения и типе верстки формируется исходя из значений обеих переменных. Для вывода даты и времени используется функция `date()`.Программе проверяет тип верстки и сохраняет его в переменной `$s`. Содержание таблицы добавляется к строке `$s` с помощью конкатенации (символ "."). Сформированная строка выводится вместе с текущей датой и временем.

Листинг А-5. 9

```
if( !isset($_GET['html_type']) || $_GET['html_type']== 'TABLE' )
      $s='Табличная верстка. '; // строка с информацией
else
      $s='Блочная верстка. ';

if( !isset($_GET['content']) )
      $s.='Таблица умножения полностью. ';
else
      $s='Столбец таблицы умножения на '.$_GET['content']. '. ';

echo $s.date('d.Y.M h:i:s');
```

СПРАВОЧНАЯ ИНФОРМАЦИЯ

Для реализации меню используются ссылки, само меню оформляется в виде одного блока. Текущий (выбранный) пункт меню оформляется в виде специального класса. В зависимости от описания блоков в CSS меню будет выглядеть как вертикальное или горизонтальное.

Листинг А-5. 10

```
<div id="main_menu">
      <a href="goTo1.html"></a>
      <a href=" goTo2.html" class="selectedMenu"></a>
      <a href="goTo3.html"></a>
</div>
```

Листинг А-5. 11

```
#main_menu a { display: block; width:25%; color: #F00 } /* вертикальное меню */

/* горизонтальное меню */
#main_menu a { display: block; width:25%; float: left; color: #F00 }
#main_menu { clear: all }
#main_menu a:hover { color: #FF0 }
#main_menu .selectedMenu, #main_menu .selectedMenu:hover { color: #0FF }
```

Если активность пункта меню определяется наличием параметра, то его можно проверить, используя функцию array_key_exists().

Листинг А-5. 12

```
<a href="?myparam=10" <?php
// ссылка активна если параметр не указан или он равен значению ссылки
if( !array_key_exists('myparam', $_GET) || $_GET['myparam']==10 )
      echo 'class="selected"';
?>>10</a>
<a href="?myparam=20" <?php
// ссылка активна только(!) если параметр равен значению ссылки
if( array_key_exists('myparam', $_GET) && $_GET['myparam']==20 )
      echo 'class="selected"';
?>>20</a>
```

При формировании ссылки постоянную часть удобнее генерировать один раз, сохраняя ее в переменной.

Листинг А-5. 13

```
$link='?type=table';
if( array_key_exists('type', $_GET) && $_GET['type']== 'block' )
	$link='?type=block';
for( $i=1; $i<=9; $i++)
{
	echo '<a href="'.$link.'&src='.$i. '">'.$i.'</a>';
}
```

Активное использование пользовательских функций упрощает структуру программы. Для часто повторяющихся действий (в рамках данной работы это оформление ссылки из числа, вывод столбца таблицы умножения) рекомендуется использовать функции.

Листинг А-5. 14

```
function outNumAsLink( $x )  // функция ВЫВОДИТ ЧИСЛО КАК ССЫЛКУ
{
	if( $x<=9 )
		echo '<a href="?content='.$x. '"> '.$x.'</a>';
	else
		echo $x;
}
```

КОНТРОЛЬНЫЕ ВОПРОСЫ К ЛАБОРАТОРНОЙ РАБОТЕ

Для успешной защиты работы помимо соответствующего требованиям результата необходимо уверенно отвечать на нижеперечисленные и другие вопросы, а также на контрольные вопросы всех предыдущих лабораторных работ.

1. Можно ли в ссылке использовать три параметра?
2. Можно ли передавать в ссылку параметр без значения?
3. Какое начальное значение может быть у цикла со счетчиком?
4. Что такое URL?
5. Что такое URI?
6. Что такое конкатенация?
7. Можно ли применить операцию конкатенации к строке и функции?