МИНИСТЕРСТВО ОБРАЗОВАНИЯ и науки РЕСПУБЛИКИ БАШКОРТОСТАН

Государственное автономное ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

УФИМСКИЙ КОЛЛЕДЖ СТАТИСТИКИ, ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 1

По МДК.05.02. Проектирование информационных систем

Выполнил студент группы 21ВЕБ-1

Ардуванов Радмир

Проверил(а) преподаватель информатики

и программирования

Дмитриева Елизавета Константиновна

2024

Тема

Обработка элементов формы на языке PHP

Цель

* повторить основные теги языка HTML;
* понять принцип формирования динамических страниц;
* научиться работать с переменными, условным и циклическим операторами языка PHP.

Задачи

* Создать учебник по одной из задач ЕГЭ, содержащий варианты заданий, решения и ответы на задания (2 веб-станицы). Для оформления страниц использовать максимальное количество тегов HTML5, при этом их использование должно быть уместным.
* Создать программу, решающую поставленную задачу. На экран выводятся однотипные задания (не менее 4-х вариантов), содержащие случайные значения (таким образом, чтобы задания не повторялись), формулы, демонстрирующие решение этих заданий (с подставленными значениями), и ответы к этим заданиям.
* Дополнить задание формой для осуществления ввода ответа на задание. Предусмотреть ввод с использованием метода POST, проверку верности вводимого ответа и вывод IP-адреса компьютера, отправляющего запрос.

Ход работы

Задача 1

Создали файл index.php, в котором хранятся варианты заданий. Для каждого варианта указали свой ключ, по которому будет совершаться переход на другую рабочую область. рисунок 1



Рисунок 1

Далее создаем файл style.css, в котором прописываем стили для нашей страницы. рисунок 2

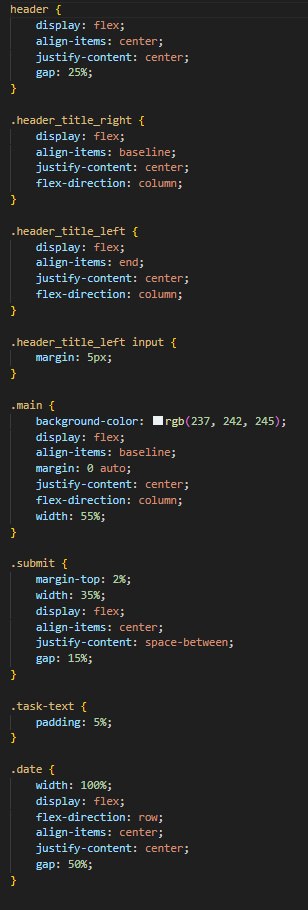


Рисунок 2

Открываем файл через OpenServer «Мои Проекты» и проверяем промежуточный результат. рисунок 3



Рисунок 3

Далее создаем файл task.php, в котором указываем переменные. Далее создаем массив, в котором прописываем варианты заданий, вписав вместо значений, наши переменные. рисунок 4

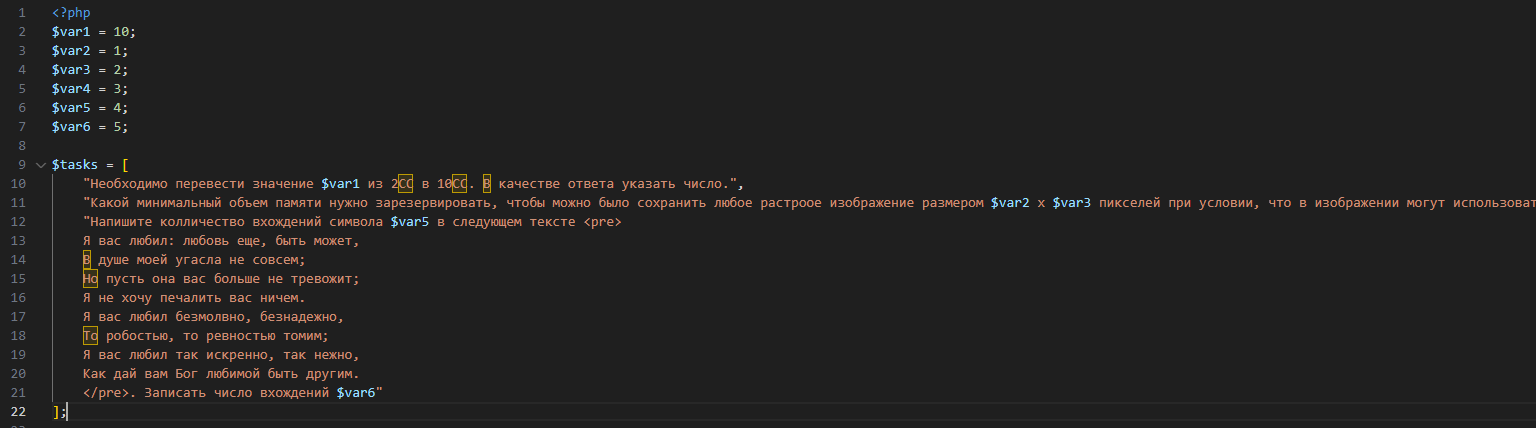


Рисунок 4

Для того, чтобы наши переменные принимали случайные значения, создаем функцию, которая будет генерировать рандомные значения. рисунок 5

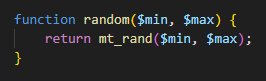


Рисунок 5

Изучили первое задание, в котором нужно перевести данные нам числа из двоичной в десятичную систему исчисления. Создали функцию get2SS, которая формирует строку из случайных двоичных чисел 0 или 1 определенной длины, и передали ему параметр $len. Причем первый символ не может ровняться нулю. Это условие вы прописали в условном операторе if…else. рисунок 6

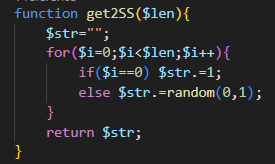


Рисунок 6

Далее создали создали цикл for, выполняющий 4 итерации. В каждой итерации происходит проверка значения переменной $GET['task'] с помощью оператора switch. Если $GET'task' равна 0, то выполняется код, содержащийся в блоке case 0. Следовательно, вызывается функция get2SS с аргументом, представляющим случайное число в диапазоне от 2 до 5. После выполнения функции выводится текст с использованием значения переменной $var1. Далее отображается форма для ввода ответа, содержащая текстовое поле для ввода числа и кнопка для отправки ответа. Для поля сс правильным ответом задаем тип hidden для того чтобы скрыть поле от пользователя. В него будет передаваться функция для перевода из двоичной системы счисления в десятичную. Выводи еще одно поле с параграфом, в которое в дальнейшем будет передаваться значение с правильным ответом для вывода. рисунок 7

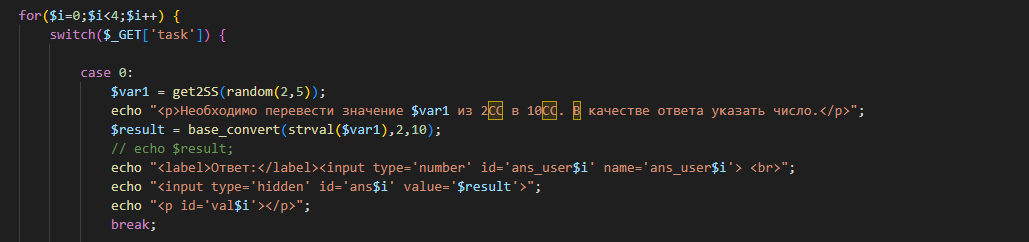


Рисунок 7

К кнопке для отправки формы тоже добавляем id. Пишем скрипт на js. Получаем кнопку и вешаем на нее обработчик события. Создаем цикл и функцию для проверки ответов. Создаем переменные в которых будут храниться значения рандомных чисел, полученных для задач. В условии проверяем равен ли ответ пользователя и вычисленный. Если ответы равны, то он принимает зеленый цвет, если не равны, то красный. Также после нажатия на кнопку будет выводиться параграф с правильным ответом. рисунок 8



Рисунок 8

Результат первого задания. рисунок 9

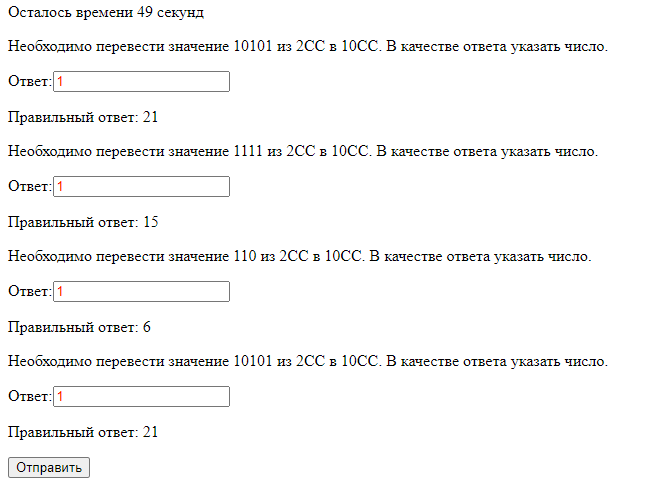


Рисунок 9

Для решения второго задания создаем две функции. Первая функция imgSize с параметром $len1. Далее создаем цикл for, в котором задаем переменную $i. Задаем условие, что она должна быть меньше либо равна 2, так как нам нужно два рандомных значения, указывающих формат изображения. Указываем условный оператор if…else, чтобы присвоить первому значению одно рандомное число, а второму значение другое рандомное число. Далее создаем массив с тремя значениями для цвета. После чего добавляем вторую функцию randColor и передаем в него раннее созданный массив. Возвращаем массив и используем функцию array\_rand, которая рандомно выбирает значения из массива. рисунок 10

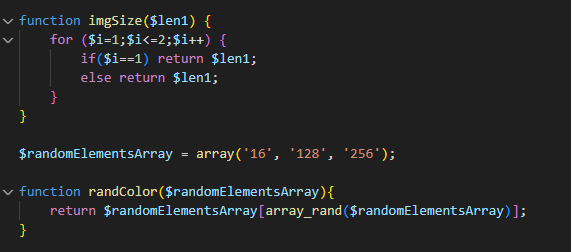


Рисунок 10

Создаем case1, задаем три переменные, в две из которых присваиваем рандомный размер изображения, а в третью присваиваем значение рандомного количества цветов из массива. Выводим текст с переменными $var2 и $var3. рисунок 10

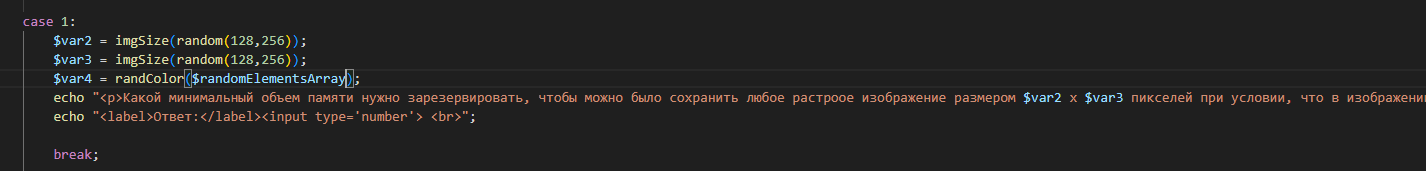


Рисунок 11

Результат второго задания. рисунок 11

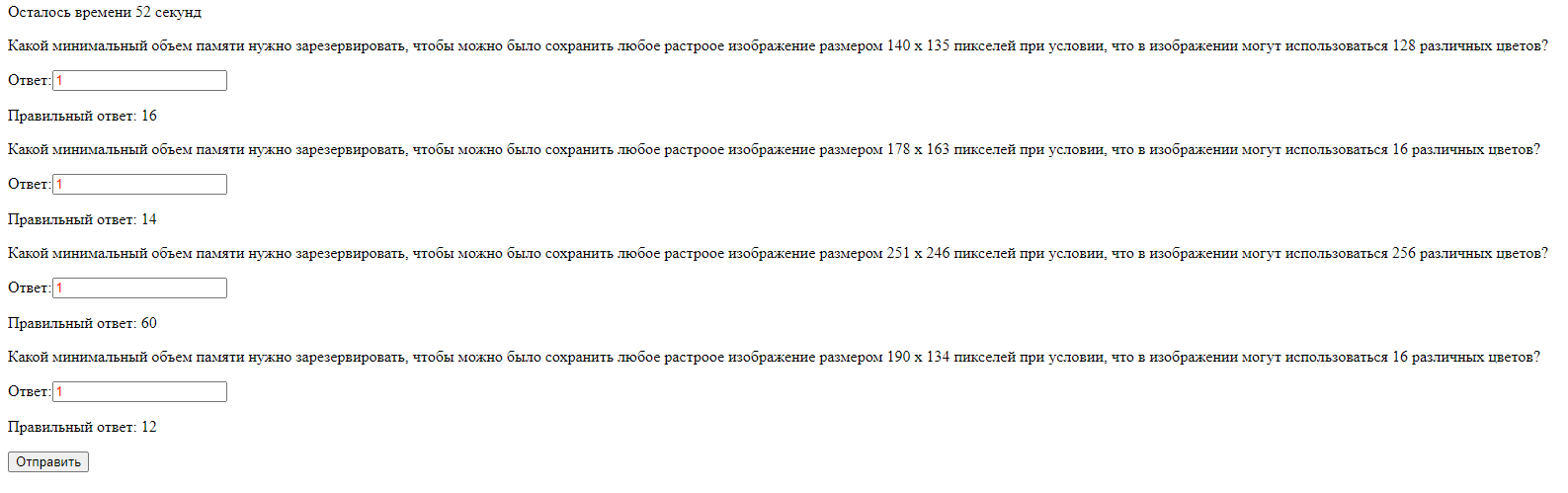


Рисунок 11

Для решения третьего задания создаем массив и заносим в него 4 стихотворения. рисунок 12

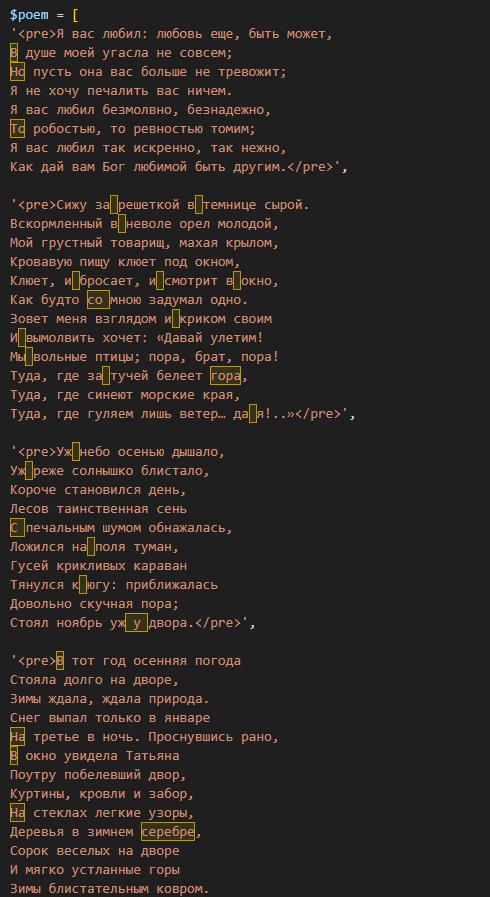


Рисунок 12

Для того, чтобы получить количество вхождений буквы в стихотворения, создаем функцию letterInPoem() c параметрами $var5, $var6. Создаем переменные в которые передаются значения буквы и массив в с буквами стихотворения. Также создаем переменную для подсчета вхождений. Используем цикл foreach для перебора массива по элементам. После чего сравниваем полученный элемент и переменную с буквой и в случае, если они равны, увеличиваем значение $count с шагом 1. После чего возвращаем эту переменную. рисунок 13

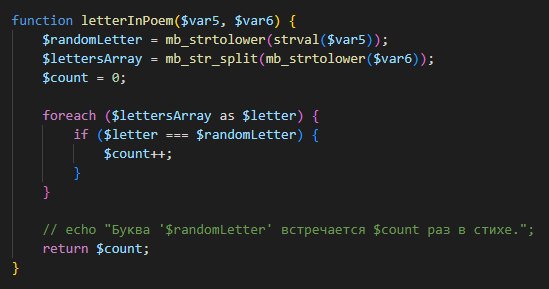


рисунок 13

Создаем функцию randPoem передаем в нее раннее созданный массив. Возвращаем массив и применяем к нему функцию array\_rand для рандомного вывода. рисунок 14

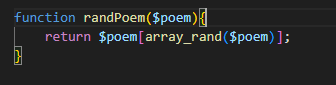


Рисунок 14

Создаем еще один массив с алфавитом, чтобы в дальнейшем сгенерировать в заданиях разные буквы.

После чего добавляем еще одну функцию и называем ее randletter и передаем массив $letter. Возвращаем его и так же применяем к нему функцию array\_rand для рандомного вывода. рисунок 15

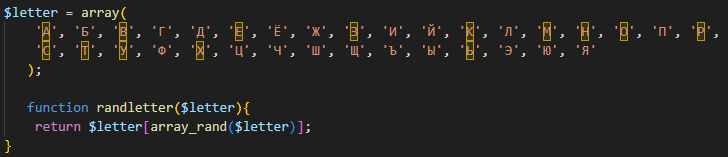


Рисунок 15

Создаем case2, задаем две переменные, в одну из которых присваиваем рандомную букву, а во вторую присваиваем значение рандомного стихотворения. Выводим текст с переменными $var5 и $var6. рисунок 16

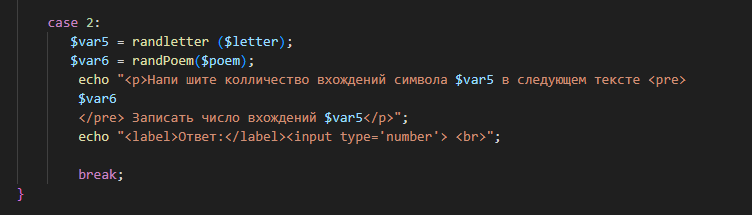


Рисунок 16

Результат второго задания. рисунок 17

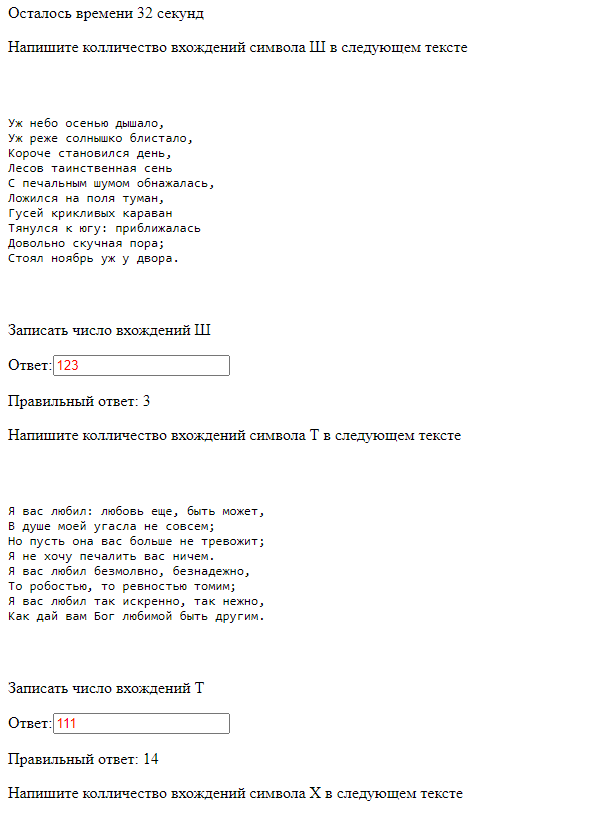


Рисунок 17

Также реализован таймер. Мы создали две переменные. Переменной seconds присвоили значение 60, а переменной interId присвоили функцию setInteval, которая позволяет выполнять функцию с определенным интервалом. Реализуем таймер на JS. В качестве первого параметра задаем функцию для получения элемента из html и присваиваем значение переменной seconds. Дальше задаем условие, что, если seconds ==0, то таймер останавливается и выводится сообщение о том, что время вышло. В другом случае, значение переменной seconds уменьшается на 1. Указанное вторым аргументом число в миллисекундах обозначает время между запусками функции. рисунок 18

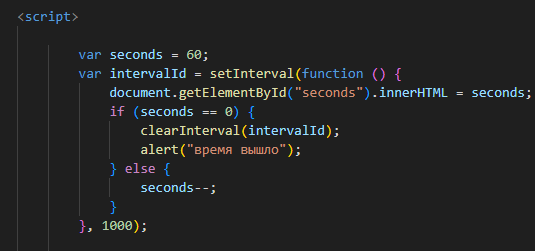


рисунок 18

Результатом работы является учебник по одной из задач ЕГЭ, содержащий варианты заданий, решения и ответы на задания, а также создали программу, решающую поставленную задачу и дополнили задание формой для осуществления ввода ответа на задание предусмотрев ввод с использованием метода POST и проверку верности видимого ответа.

Вывод

В ходе работы мы повторили теги языка HTML, поняли принцип формирования динамических страниц и научились работать с переменными, условными и циклическими операторами языка PHP