Цель домашнего задания  
Научиться подключать и применять подключение файлов на практике.  
Научиться использовать приведение типов.  
  
  
Что входит в задание  
Интеграция страниц сайта.  
Задание на подключение существующих файлов сайта.  
Задание на динамическое подключение файлов сайта.  
Задание на применение приведения типов.  
  
  
Задание 1  
Что нужно сделать  
Создайте новый сайт. В корень сайта разместите приложенный исходный код сайта. Он состоит из трёх публичных страниц: task1.php, task2.php, task3.php. Директория app содержит исходный код приложения. Директория templates содержит части шаблонов приложения.  
На каждой странице сайта подключите ядро приложения. Ядро приложения содержится в файле app/core.php. Ядро должно подключаться в самом начале страниц, до вывода первых тегов страницы. Ядро должно подключаться один раз. Если файла с ядром нет, то приложение должно падать с фатальной ошибкой.  
На каждой странице сайта в самом начале тела страницы подключите блок с навигацией (файл templates/navigation.php).  
Внутри блока с навигацией подключите название сайта templates/site\_name.php в отмеченном месте.  
На странице task1.php в контентной области подключите шаблон с контентом главной страницы: templates/task\_1\_content.php.  
При подключении ядра или шаблонов внутри публичных страниц указывайте полный путь к файлам, а не относительный. Для этого используйте переменную $\_SERVER['DOCUMENT\_ROOT']. При этом при подключении файлов внутри ядра можно применять константу \_\_DIR\_\_.  
  
  
Советы и рекомендации  
Если в коде приложения требуется подключить обязательный файл, используйте директиву require.  
При подключении простых шаблонов используйте директиву include. Ничего страшного, если какой-то визуальной части сайта не появится (при отсутствии соответствующего файла с шаблоном), зато остальная часть сайта будет функционировать.  
Для гарантированного подключения файла только один раз используйте одну из директив: require\_once или include\_once.  
  
  
Что оценивается  
Все страницы сайта должны открываться и выполняться без ошибок.  
Файл с ядром должен быть подключен на каждой странице.  
В верхней части сайта должно выводиться и работать навигационное меню.  
В навигационном блоке в шапке сайта должно выводиться название сайта.  
В теле страницы task1.php должен быть выведен приложенный контент для этой страницы.  
Все файлы подключены без использования относительных путей.  
  
  
Задание 2  
На странице task2.php необходимо вывести форму авторизации, если пользователь не авторизован, или приветственное сообщение и имя пользователя, если он авторизовался.  
  
  
  
Что нужно сделать  
Создайте новый файл с новым исходным кодом: authorization.php  
Внутри этого файла создайте переменную $isAuthorized и положите в нее случайное значение true или false  
Внутри этого файла также создайте переменную $userName и поместите в нее строку, содержащую ваше имя  
Подключите этот файл внутри ядра вашего приложения  
Если значение переменной $isAuthorized истинно, то на странице task2.php подключите файл templates/task\_2\_welcome\_message.php. При этом доработайте этот файл и вместо плесходера #имя# выведите значение переменной $userName  
Если же значение переменной $isAuthorized ложно, то на странице task2.php подключите файл templates/task2\_auth\_form.php.  
  
  
Советы и рекомендации  
Чтобы создать случайное boolean значение, создайте случайно число от 0 до 1 и приведите его к boolean типу.  
Все подключенные переменные в файле с ядром приложения будут доступны во всех шаблонах вашего приложения, так, что не подключайте лишний раз файл authorization.php где-бы то ни было.  
  
  
Что оценивается  
На каждой странице теперь доступны две переменные $isAuthorized - boolean типа и $userName - строка  
На странице task2.php корректно выводится форма авторизации или приветственное сообщение, по требуемой логике  
В тексте приветственного сообщения выводится значение переменной $userName.  
  
  
Задание 3  
Что нужно сделать  
Подключите в файле с ядром файл app/array\_of\_data.php, содержащий массив с данными $data.  
В файле task3.php в отмеченном месте для каждого элемента массива $data подключите шаблон templates/task\_3\_card.php, чтобы он корректно выполнялся. Значение каждого элемента массива $data должно быть помещено в переменную с названием $value.  
В результате вы должны увидеть на странице каталог карточек, по одной на каждый элемент, где указано, какое значение рассматривается, какой у него тип и результат приведения к числу и сравнения с единицей.  
  
  
Советы и рекомендации  
Для прохода по массивам лучше всего подходит цикл foreach.  
Внутри файлов с подключаемыми шаблонами доступны те же переменные, что и в том месте, где они подключаются.  
Не нужно вносить изменения в файл с шаблоном карточки, чтобы всё заработало.  
  
  
Что оценивается  
Массив $data теперь доступен на всех страницах сайта.  
На странице task3.php выводятся карточки, по одной для каждого элемента массива $data.  
  
  
Задание 4  
Что нужно сделать  
Доработайте шаблон templates/task\_3\_card.php. Добавьте в него вывод ещё нескольких элементов списка.  
Добавьте в этот список приведение значения к boolean и сравните его с true.  
Добавьте в этот список приведения значения к строке и сравните его с пустой строкой.  
Добавьте в этот список приведения значения к строке и сравните его с со строкой '1'.  
  
  
Что оценивается  
На странице task3.php в каждой карточке для каждого элемента массива $data должно быть по 4 элемента в списке. В каждом элементе происходит требуемое сравнение.  
  
  
Критерии оценки  
Открываются и выполняются без ошибок все страницы сайта.  
Выполнены все задания.  
Все файлы подключены с указанием их полного пути.  
  
  
Как отправить задание на проверку  
Домашнее задание сдаётся в виде архива, содержащего все файлы разработанного сайта с сохранением структуры директорий.