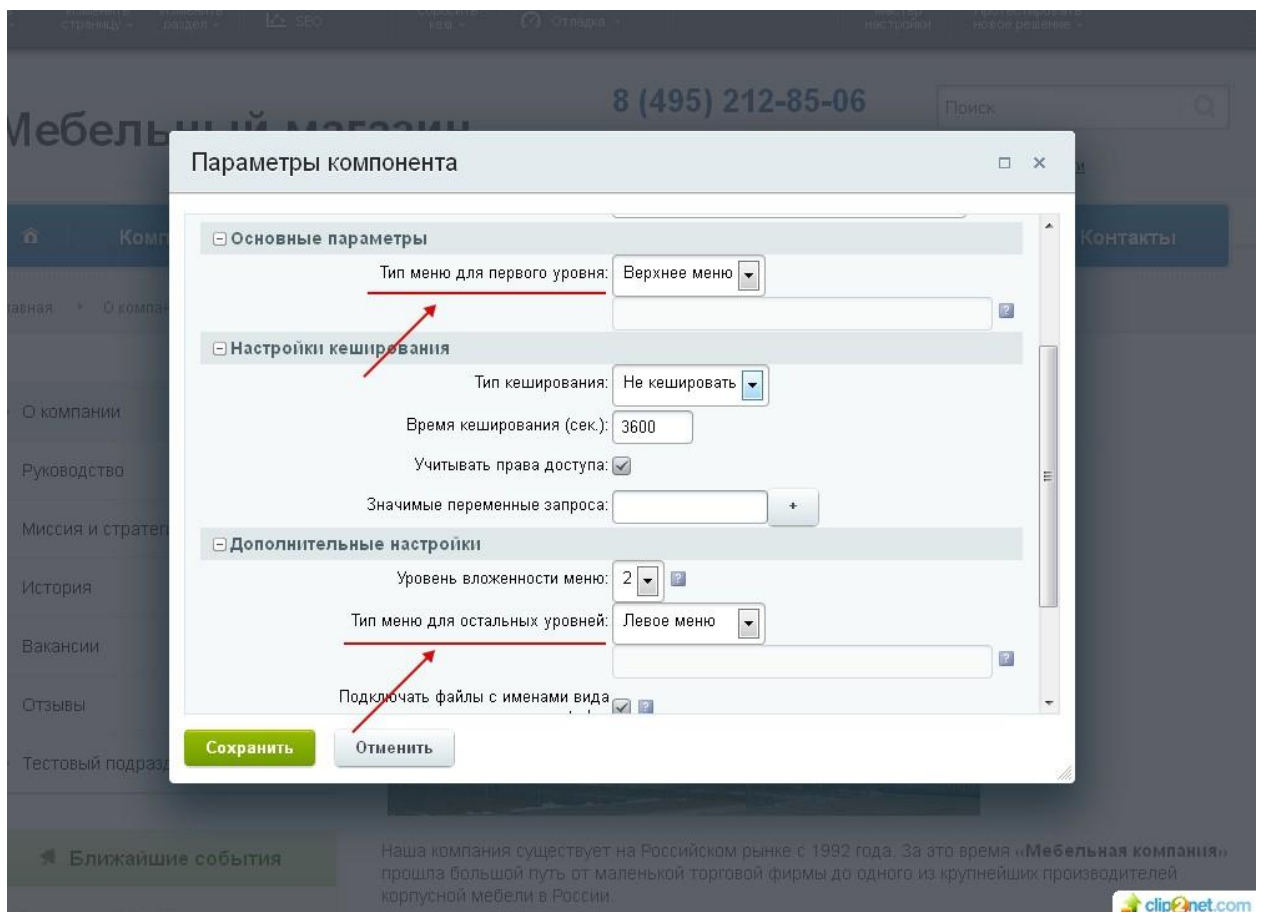


## Комментарии к уроку №4.1

1. Подробнее об управлении меню [http://dev.1c-bitrix.ru/learning/course/index.php?COURSE\\_ID=34&LESSON\\_ID=1862](http://dev.1c-bitrix.ru/learning/course/index.php?COURSE_ID=34&LESSON_ID=1862)  
[http://dev.1c-bitrix.ru/learning/course/index.php?COURSE\\_ID=35&LESSON\\_ID=2018](http://dev.1c-bitrix.ru/learning/course/index.php?COURSE_ID=35&LESSON_ID=2018)
2. Чтобы в верхнем меню получались выпадающие пункты, необходимо добавить левое меню в соответствующие разделы по ссылкам из верхнего меню.
3. По логике установленного решения левое меню появляется только в тех разделах, где есть подразделы. Оно выводит только подразделы (подпункты) текущего раздела. Если в разделе нет подразделов, то меню не выводится.
4. При внедрении включаемой области для раздела можно включить рекурсивное наследование для подразделов. Тогда во всех вложенных подразделах будет подключаться та же область. Но при этом в разделе можно будет создать свою включаемую область, которая будет в нем и в его подразделах. Например, нам нужно во всех разделах сайта, кроме раздела Каталог, показывать в левой колонке баннер. В нужном месте шаблона размещаем компонент «Включаемая область». В параметрах компонента выбираем «Показывать включаемую область для раздела» и отмечаем галочку «Рекурсивное подключение включаемых областей разделов». В корне сайта создаем соответствующую область, теперь она будет наследоваться для всего сайта. Для раздела Каталог создадим свою область с другим баннером или пустую, если ничего показывать не нужно.
5. Для определения текущей папки используется функция `$APPLICATION->GetCurDir()`. Подробнее можно почитать на [http://dev.1c-bitrix.ru/api\\_help/main/reference/cmain/getcurdir.php](http://dev.1c-bitrix.ru/api_help/main/reference/cmain/getcurdir.php)
6. Иногда ошибкой является то, что в разделе «О компании» подпункты 1-го пункта меню дублируют все остальные пункты:

Ваш проект будет создан с учетом всех нюансов и прорисовкой мельчайших деталей. Результаты совместного творчества вы сможете посмотреть в объемном представлении. Вам наглядно продемонстрируют, как будут выглядеть в жизни выбранные элементы интерьера при разном

В настройках компонента Вы указываете тип меню для пунктов 1-го уровня и для остальных уровней:



При формировании меню, компонент «Меню» для каждого пункта 1-го уровня «смотрит», куда ведёт его ссылка. Если ссылка ведёт на раздел, компонент проверяет, есть ли в этом разделе файл `.тип_меню_2-го_уровня.menu.php`, и если находит, выводит из него подпункты для данного пункта 1-го уровня. Если же ссылка ведёт на страницу или не оканчивается на слеш, наличие дочернего меню не проверяется и подпункты у текущего пункта не выводятся.

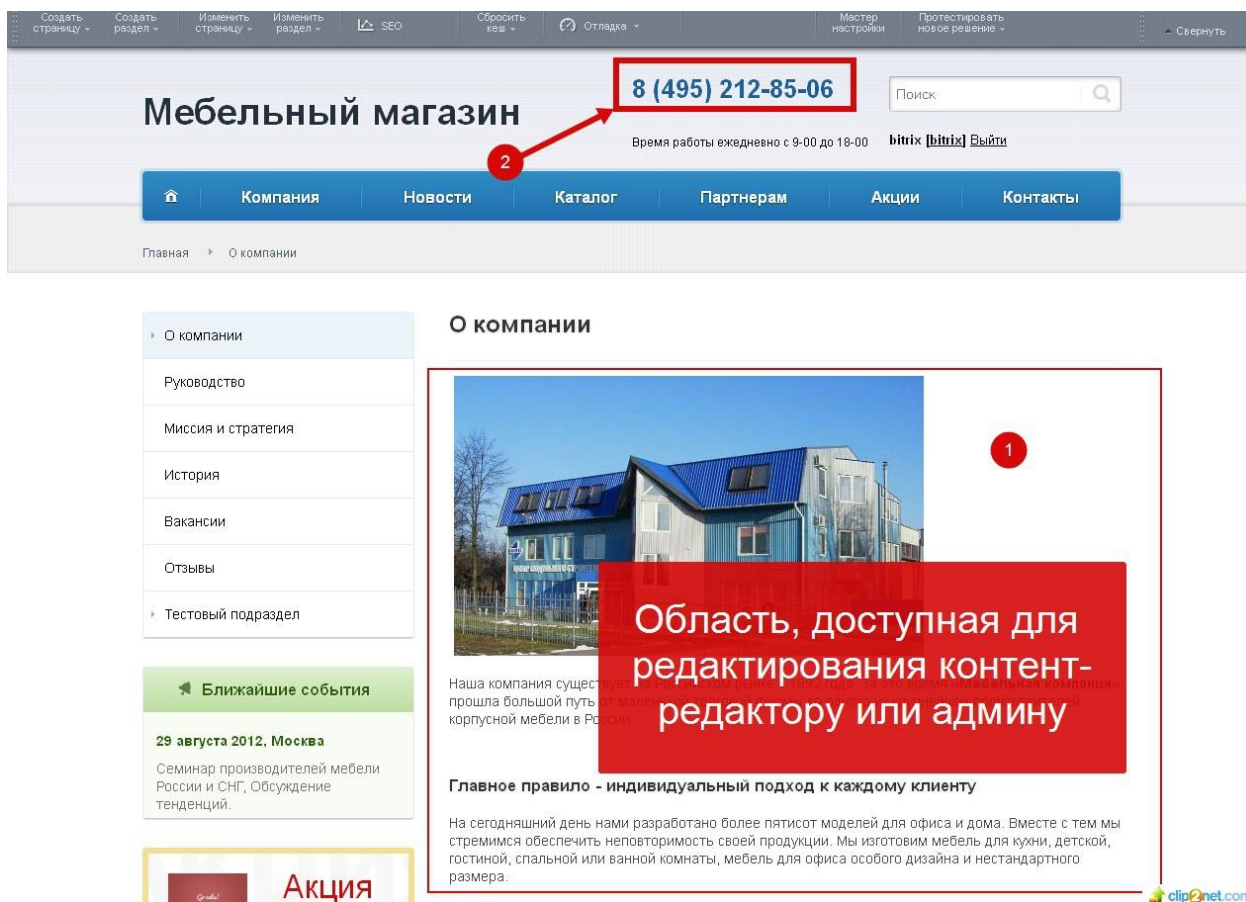
Иногда на площадках слушатели устанавливают для 1-го и 2-го уровня одинаковый тип меню – `left`. В итоге, компонент ищет в текущем разделе файл `.left.menu.php`, выводит из него пункты 1-го уровня. Далее, если 1-й пункт ведёт на этот же раздел, компонент снова подключает файл `.left.menu.php` для подпунктов 1-го пункта.

В итоге появляются дублирующие подпункты, как на скриншоте.

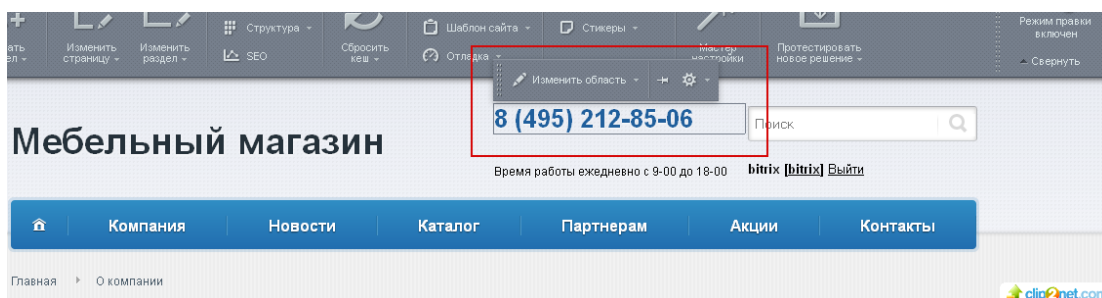
7. По заданию требуется вынести телефон и адрес в шаблоне сайта в подключаемые файлы. При этом в подключаемых файлах должен находиться только текст, без html-тегов.

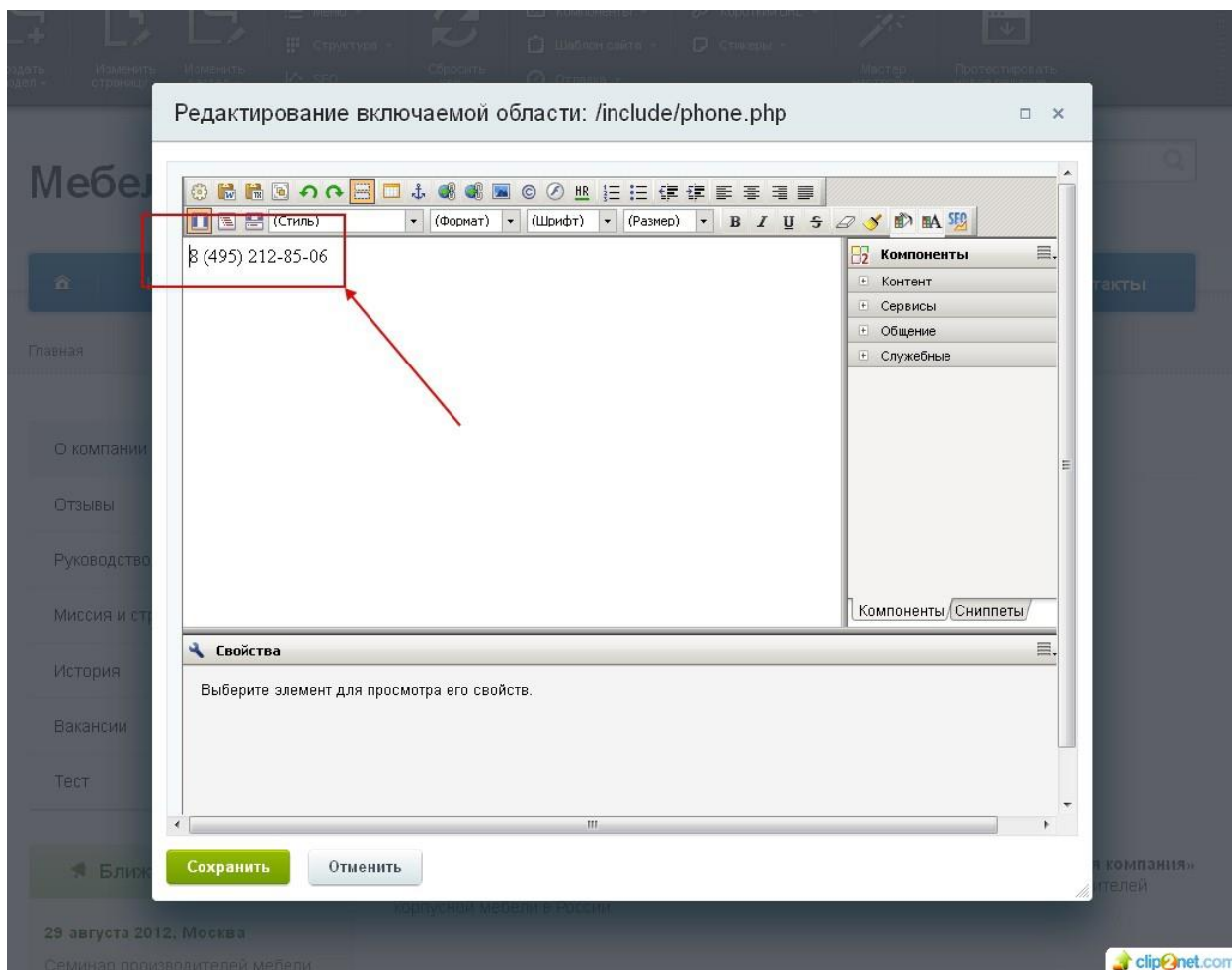
Смысл данного задания в следующем. При наличии прав на редактирование контент-редактор или админ может редактировать основное содержимое страницы (область **1** на скрине). Но помимо основного контента, в шаблоне сайта тоже встречаются области, для которых желательно, чтобы контент-редактор или админ имел возможность править. Например, телефон в шапке (область **2** на скрине).

Если в шаблоне данный текст (для области 2) прописать жёстко, возможности редактировать его в режиме правки не будет. Именно для того, чтобы дать возможность редактировать элементы в шаблоне сайта, они и выносятся в подключаемые файлы с помощью компонента



Тогда в режиме правки у нас появляется возможность их редактировать:



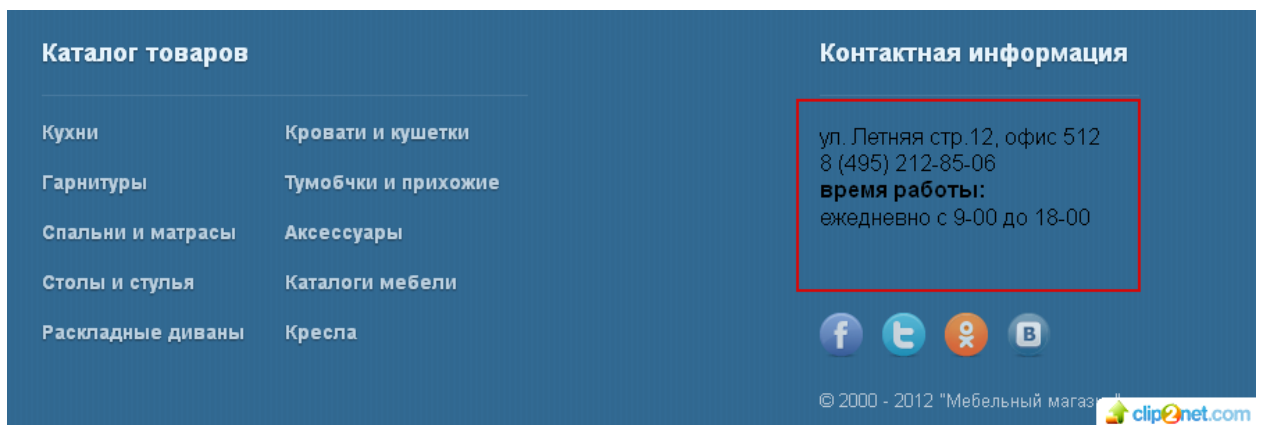


В подключаемом файле нельзя допускать, чтобы содержались элементы вёрстки, чтобы контент-редактор при неаккуратном редактировании не мог как-либо испортить вёрстку на сайте.

8. В режиме правки вёрстка на сайте иногда может «ехать». Так, например, блок «Контактная информация» в футере в обычном режиме выглядит нормально:



При включении режима правки на сайте цвет текста становится чёрным:



Причина в том, что в режиме правки Битрикс обрамляет компонент в свой собственный div для привязки к нему в яваскрипте. Из-за этого и может нарушаться вёрстка.

9. В Битрикс различные элементы сайта могут иметь свои языковые файлы. Например, шаблон сайта, компонент и шаблон компонента. Некоторые языковые файлы подключаются системой автоматически, другие нужно подключать явно вызовом специальных функций.

В данном задании требуется интегрировать шаблон меню «О магазине» и заголовок меню вывести из языкового файла шаблона меню.

Языковой файл для шаблона компонента подключается автоматически. Для файла папка\_с\_шаблоном\_компонента/ **template.php**

соответствующий языковой файл нужно лишь разместить в папке:

папка\_с\_шаблоном\_компонента /**lang/ru/template.php**

Вообще общее правило для размещения языковых файлов следующее. Название языкового файла должно соответствовать названию того файла, для которого он подключается, относительно того файла, для которого подключаются.

10. Реализация шаблона двухуровневого меню кажется довольно сложной поначалу для начинающих знакомиться с Битрикс. Причина в том, что пункты многоуровневого меню имеют древовидную структуру (у каждого пункта могут быть собственные подпункты), но в шаблон меню информация о всех пунктах передаётся «линейно», в виде одномерного массива. Т.е. в массиве порядок элементов такой:

**пункт 1, подпункт 1-а, подпункт 1-б, пункт 2, пункт 3, подпункт 3-а** и т.д.

Поэтому в шаблоне все они обычно выводятся с помощью одного цикла, но имеющего множество условий и проверок.





```

        <?if ($arItem["DEPTH_LEVEL"] == 1):?> // Выводим вёрстку для пунктов 1-го уровня
        <li><a href="<?=$arItem["LINK"]?>" class="<?if ($arItem["SELECTED"]):?>root-item-selected<?else:~>root-
item<?endif?>"><?=$arItem["TEXT"]?></a></li>

        <?else:~> // Выводим вёрстку для пунктов остальных уровней.
        <li><a href="<?=$arItem["LINK"]?>" <?if ($arItem["SELECTED"]):?> class="item-
selected" <?endif?>"><?=$arItem["TEXT"]?></a></li>

        <?endif?>

        <?else:~> // Если пользователю запрещено просматривать страницу по ссылке, выводим лишь текст, даже без
ссылки

        <?if ($arItem["DEPTH_LEVEL"] == 1):?> // Выводим вёрстку для пунктов 1-го уровня
        <li><a href="" class="<?if ($arItem["SELECTED"]):?>root-item-selected<?else:~>root-item<?endif?>"
title="<?=GetMessage("MENU_ITEM_ACCESS_DENIED")?>"><?=$arItem["TEXT"]?></a></li>

        <?else:~> // Выводим вёрстку для пунктов остальных уровней.
        <li><a href="" class="denied"
title="<?=GetMessage("MENU_ITEM_ACCESS_DENIED")?>"><?=$arItem["TEXT"]?></a></li>

        <?endif?>

        <?endif?>

        <?endif?>

        <?previousLevel = $arItem["DEPTH_LEVEL"];?>

<?endforeach?>

<?if ($previousLevel > 1):~> // Ещё раз закрываем все открытые теги.
        <?=str_repeat("</ul></li>", ($previousLevel-1) );?>
<?endif?>

</ul>
<?endif?>

?>

```

Дополнительные материалы по структуре шаблона меню можно посмотреть [здесь](#):