

Разрядка в оформлении текста при вёрстке в индизайне

История вопроса

Выделение слов в тексте при помощи разрядки, т.е. увеличение межбуквенного интервала было одним из вариантов оформления в книгах, выпускавшихся в нашей стране в прошлые века. Но это, видимо, была в основном русская традиция. В Европе она не приветствовалась, вспомните хотя бы рекомендацию Чихольда не применять разрядку в текстах, заменять на курсивное начертание.

Так это европейские традиции. Но есть ведь еще и наши. Листаешь старые книги, сделанные горячим набором, и выделения разрядкой совсем не кажутся чужеродными. Помню, что в учебниках русского языка использовались и разрядка, и курсив.

Так вышло, что мы для вёрстки используем программы, созданные за рубежом, и этот вариант оформления в перечне возможностей программ фактически не представлен. Есть возможность изменять межбуквенный интервал, но это только один из аспектов оформления разрядкой.

Но то что в программе вёрстки нет полноценного инструмента оформления разрядки, не повод отказываться от этого способа, традиционного для русской типографии.

Чтобы снять с пользователя всю рутину оформления текста разрядкой, оставив ему только творческие задачи, сделан этот скрипт.

Проблемы оформления разрядкой

1. Часто разрядку имитируют, ставя между буквами слов пробелы, а межсловный интервал отмечая двумя или тремя пробелами. Это ужасное решение. Во-первых, слова исчезают из текста, их уже не найдёшь поиском. Во-вторых, слова рассыпаются на отдельные буквы, и невозможно управлять величиной межбуквенного интервала.

2. Полноценная разрядка может прийти в тексте на вёрстку: автор изменением межбуквенных интервалов отмечает слова, которые должны быть оформлены этим способом. Это хорошо, но к сожалению, в одном материале может быть несколько вариантов значений интервала. А всё должно быть единообразно.

3. Знаки пунктуации — запятая, точка, двоеточие и пр. — должны иметь нулевой межбуквенный интервал, и буквы слева от них тоже должны быть с таким же значением разрядки. Это верно и для шпаций вокруг тире.

4. В индизайне значение разрядки для текста обычно не превосходит 200 единиц. Но это очень велико для цифр, для них разрядка должна быть 50 единиц. То есть в программе должна быть независимая установка разрядки для текста и разрядки для чисел.

Очевидно, что эти проблемы не добавляют охотников иметь в арсенале своих инструментов разрядку — очень много с ней мороки, если решать все эти задачи теми средствами, что есть в индизайне.

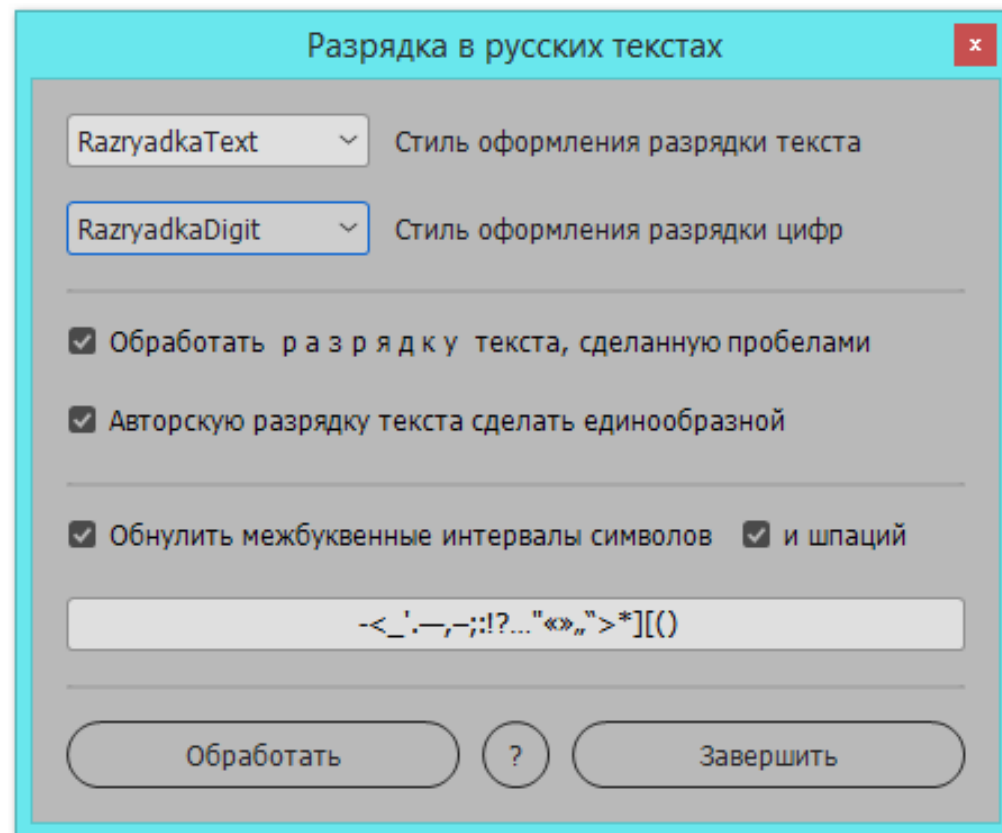
И вот программа, которая берет на себя все эти хлопоты. Пользователю остаётся только применять её к выделенному тексту, и пробовать разные варианты, как текст будет лучше смотреться. Тут выбора особого нет: а) включить / выключить переносы в абзаце; б) выбрать один из компоновщиков — однострочный или многострочный. Но даже два варианта при том, что всю кропотливую работу за тебя сделала программа, это уже большое дело.

Опции программы

Стиль оформления разрядки текста

Стиль оформления разрядки цифр

Выбор символьных стилей для текста и для цифр. Создавая эти стили, определяйте только трекинг. Для текста начните с числа 190. Для цифр проверенное значение 50.



Обработать разрядку текста, сделанную пробелами

Обработка разрядки, сделанной добавлением пробелов между буквами. Важно, чтобы межсловные интервалы были отмечены не менее чем двумя пробелами, иначе всё сольется в одну строку. Межбуквенные пробелы будут удалены, буквы станут словами, текст и цифры получают соответствующие символьные стили.

Авторскую разрядку текста сделать единообразной

Разной в используемых в тексте межбуквенных интервалах будет приведён к единому оформлению в соответствии с символьными стилями разрядки.

Обнулить межбуквенные интервалы символов и шпаций

Символы, указанные в строке, и знак слева от каждого такого символа будут иметь нулевой межсимвольный интервал.

Обрабатываются все шпации, кроме обычного пробела и табуляции.

Работа с программой

В папке Info есть файл `BetweencharacterSpaceControl.idml`. Там есть таблица с вариантами выбора величины разрядки для текста. Так, значение 210 соответствует случаю, когда слово разреживается пробелами. Если бы для этой цели использовались шпации 1/6 круглой, то значение разрядки было бы 170, если бы тонкие шпации, то 120. Получается, что оптимальное значение лежит в диапазоне 170–200.

В этом файле есть два символьных стиля для оформления текста и цифр, они называются **RazryadkaText** со значением трекинга 190 и **RazryadkaDigit**, в котором эта величина равна 50. Исключительно для отображе-

ния текста с изменённой разрядкой эти символьные стили красят цвет в разные цвета. Для своей работы задайте в их установках чёрный цвет.

В том же файле даны примеры разных вариантов оформления разрядки, и абзацы с символами и шпациями, для которых надо обнулить межбуквенные интервалы. Для демонстрации всех проблемных случаев оформления разрядки есть исходный вариант и конечный результат.

Выделение разрядкой — это обычно небольшой участок текста. Чтобы случайно не выделить большой текст и потом ждать, когда всё будет обработано, объемом обрабатываемого текста ограничен 12500 знаками. Этот ограничитель расположен в запускающем скрипте `SpaceControl.jsx`, и называется `charLimit`. Не делайте его неоправданно большим.

Стартовый набор знаков, у которых будут обнуляться межсимвольные интервалы, приведён в переменной `noRasrCharLetters` в файле `SpaceControl.jsx`.

Символы, перед которыми для гугл-поиска требуется иметь обратную наклонную черту, собраны в перемен-

```
var charLimit = 12500; // наибольшее число знаков, обрабатываемое за один раз.
var noRasrCharLetters = "-<_'\.,-;:!\?...\|'«»">*]{}"; // ' — это одиночная прямая кавычка |" — это двойная прямая кавычка.
var needDoubleSlashCharacters = "[\]{}\\|"; // для гугл-поиска этих знаков требуется, чтобы перед ними были два обратных слеша.
// Их добавит скрипт. | \ — это одиночная обратная черта.
```

Переменные, определяющие работу программы. Они размещены в запускающем скрипте `SpaceControl.jsx`

ной [needDoubleSlashCharacters](#). Если появится ещё какой такой неучтённый знак, то в строки [noRasrCharLetters](#) и [needDoubleSlashCharacters](#) добавьте его как есть, без обратной наклонной черты. С добавлением обратной наклонной черты скрипт сам разберётся.

Также можно добавлять знаки в строку на экране во время работы программы. Но чтобы они сохранились для следующих сеансов, завершать работу надо нажатием клавиши Завершить.

Клавиши Ctrl+Z вернут вёрстку к состоянию до запуска скрипта, если результат не совпал с ожиданиями. Это будет, например, в случае, если для обработки цифр выбран не тот символьный стиль.

Кнопка Завершить запомнит все установки скрипта и место окна на экране. Настройки сохраняются в файле Razryadka.ini в каталоге скрипта, в папке BIN. Удаление этого файла — как запуск скрипта первый раз, т.е. сброс к начальным установкам: рабочее окно по центру экрана, в строке знаков содержимое переменной [noRasrCharLetters](#).

Михаил Иванюшин
m.ivanyushin@gmail.com
<https://dotextok.ru>