

Оформление знака ударения

Знак ударения — это символ с кодом 0x301, имеющий нулевую ширину. Этот знак используется для ударения над строчными и прописными буквами. Есть несколько шрифтов, в составе знаков которых есть такой код. Это PT Serif, Tahoma, Cambria, Calibri, Times New Roman и *немногие* другие, таких гарнитур очень мало. Очевидно, что если в работе именно такая гарнитура, то с ударениями проблем быть не должно. В большинстве случаев на месте кода 0x301 увидишь розовый маркер отсутствующего в гарнитуре знака, и это было причиной искать решение вопроса. Надежда, что в составе знаков будут латинские со знаком ударения весьма зыбкая, можно надеяться на ударения только для букв а, е, о. А таких, только наших ы, э, ю, я не найдёшь. Вопросы работы с знаками ударений периодически обсуждаются в разных аспектах, вот внизу скан с форума **prodtp**: предлагается в текущей гарни-

туре взять апостроф и приспособить его вместо знака ударения. Есть ещё такая рекомендация, как решить этот вопрос: 1) сделать копию используемого шрифта; 2) в программе, вероятно FontLab или подобной, все гласные буквы сделать с ударениями. 3) заменить в тексте ударные буквы знаками из второго набора. Это, во-первых, затратный по времени метод, во-вторых, оформленный такой вспомогательной гарнитурой текст безвозвратно потеряет ударения, если оформить текст другим шрифтом. И все ли знают FontLab? Общее у этих подходов то, что ищется решение в рамках придуманного ограничения — использовать знаки *только* этого шрифта. А что если попробовать рвануть за флажки, подумать иначе: ведь знаки ударения разных шрифтов ну очень похожи, что если попробовать найти решение, как использовать ударения из других шрифтов. Какие могут быть проблемы?

Залазишь в глипсы, находишь апостроф или что-то близкое, вставляешь символ, играешь кеглем, базелиншифтом, трекингом загоняешь его над буквой, следующую букву тем же трекингом отодвигаешь на приемлемое для глаза расстояние. Если кто-то делает по-другому - милости прошу поделиться опытом.

Что надо учесть

Возможность выбора шрифта знака ударения

Итак, надо иметь возможность выбирать из списка шрифтов тот, чей знак ударения будет использоваться. При этом вы должны видеть, как смотрятся рядом гласная и знак ударения.

Буква и ударение: взаимное размещение

Вот ниже две строки с ударениями, в первой шрифты, имеющие знак ударения, Times New Roman и Calibri

крѳша кнѳга

в строке ниже сперва Minion Pro с ударением PT Serif, затем Myriad Pro с ударением PT Sans.

крѳша кнѳга

Очевидно, что когда буквы с засечками/рубленые и символ ударения из подобной гарнитуры, то этот знак не кажется чужим. Но вот то, что во всех таких вариантах размещения ударений этот знак заметно смещѳн к правой границе буквы, это плохо. Хочется этим управлять, ну для сравнения — вон, в первой строке, как гармонично, по центру буквы стоят ударения, когда они из своей гарнитуры.

Сдвинуть ударение влево можно изменением трекинга:

кнѳга кнѳга

шрифт FranklinGothicBookC, знак ударения Cambria. Слева стандартно поставленное ударение, справа буква под ударением имеет трекинг –165, за сѳт чего оно сдвинулось влево, а сам знак ударения имеет трекинг 165, что отодвигает букву после ударения на прежнее место. И пара 'буква - ударение' смотрится красиво, хотя знак ударения из другой гарнитуры.

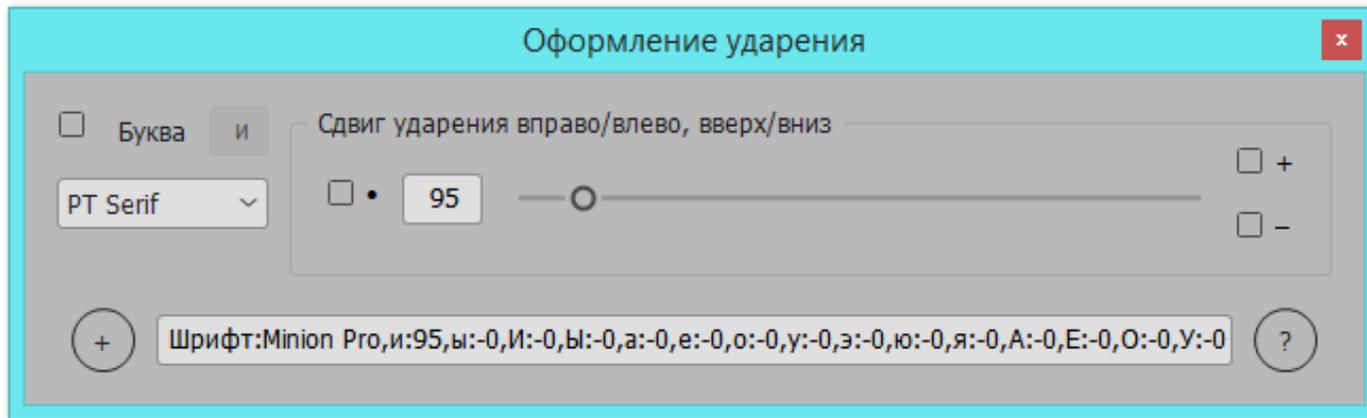
И хочется чуть поднять знак ударения, это тоже момент, который надо попробовать решить.

Очевидно, что в конкретных гарнитурах для каждой буквы будет своё значение трекинга, при котором знак ударения и буква смотрятся гармонично.

Реализация

Идея такая:

- 1) иметь массив, в котором пользователь может определять, какие шрифты с знаком ударения нулевой ширины могут использоваться.
- 2) информацию о трекингах букв, выбранном для ударения шрифте хранить в xml-файле.



3) использовать слайдер для представления положения ударения над буквой, это делает игрой скучный процесс подбора нужных значений трекинга для каждой буквы.

Такая **игровая приставка** к индизайну называется `#TestTrackValue.jsx`, выше рабочее окно этого скрипта.

В файле общих параметров, он называется `#AccentMarkData.jsx` и хранится много чего, сейчас важно знать, что там список шрифтов, которыми предполагается оформлять знак ударения. Это переменная `fontForAccentA`, в ней по умол-

чанию четыре шрифта: "PT Serif", "PT Sans", "Calibri", "Cambria".

Игра запускается так:

В шрифте, не имеющем знака ударения нулевой ширины, вы выделяете гласную и щелкаете на флажке 'Буква', и буква появляется в поле справа от слова 'Буква'.

После этого загружается информация об этом шрифте, её можно видеть в информационной строке внизу окна. В поле слева от слайдера помещается текущее значение трекинга. Если с этой буквой ещё не работали, то значение будет `-0`.

Выбранная буква и ударение над ней отобразятся на экране в масштабе 2000%, и вы ползунком слайдера определите, как должны быть скомпонованы эти знаки.

Выбор другого шрифта оформления ударения, это выпадающее меню слева, изменит картинку, и иногда и расположение знака ударения. В этом и особенность использования разных шрифтов для ударения: они не одинаковые, надо искать то, что подходит. Но это вы сделаете один раз, и потом просто будете использовать снова и снова. Слева от слайдера флажок, он устанавливает значение ползунка, если величина трекинга указывается в поле слева от слайдера. Справа от данного флажка точка, это указатель, что сейчас в этом поле значение трекинга.

Но это поле используется и для отображения другого параметра. Ударение иногда хочется поднять чуть-чуть вверх, как вы это видели на второй странице, или сдвинуть вниз, если это гарнитура Arno, и для таких действий есть пара флажков справа от слайдера. При первом нажатии на любой из них в этом поле отобразится текущая поправка к положению базовой линии, и при последующих щелчках это число будет увеличиваться или уменьшаться. Каждая единица этой поправки равна двадцатой части кегля знака ударения. Слева от информационной строки находится кнопка '+', она для сохранения информации. При каждом нажатии сохраняется информация о трекинге, она разная у разных букв, название шрифта знака ударения и поправка высоты базовой линии ударения.

Если вы двигаете ударение вверх или вниз, эта кнопка недоступна. Чтобы её активировать, надо щёлкнуть на флажке слева от слайдера. В информационном поле вновь появится найденное вами значение трекинга, оно и попадёт в xml-файл.

В русском языке всего 18 букв, с которыми надо поиграть этим скриптом — девять строчных и девять

прописных. Проблемные это и И ы Ы, с другими может быть и так, что вас устроит, как ударение и буква смотрятся вместе. Тогда просто щёлкните на флажке слева от слайдера, чтобы -0 стал нулём, и сохраните это значение. После щелчка на кнопке сохранения '+' информационная строка очищается, и все флажки изменения положения и слайдер отключаются. До выбора следующей буквы. И как выберете, вы можете в информационной строке увидеть сохранённые результаты, в частности, меня всегда поначалу интересовало, записался ли только что сохранённый трекинг. :)

И хранится эта информация в файле, который называется LetterTracking(UTF-8).txt.

Подробно о LetterTracking(UTF-8).txt

LetterTracking(UTF-8).txt — xml-файл с трекингами букв разных шрифтов, кодировка текста UTF-8. Сперва там ничего нет, только строки

```
<LetterTracking>
  <font name="Новый шрифт">
    <data/>
    <acute/>
```

```
<shift/>
</font>
</LetterTracking>
```

Это простейший xml-файл, даже декларация опущена. Теги <data>и </data> для хранения строки с информацией о трекингах знака ударения над конкретными буквами. Теги <acute>, </acute> обрамляют название шрифта, которым будет оформлено ударение. Теги <shift> и </shift> содержат поправку смещения базовой линии знака ударения.

Вот как в этом файле определено, что у Minion Pro для ударений будет использоваться шрифт PT Serif, а для Myriad Pro шрифт PT Sans:

```
<LetterTracking>
<font name="Новый шрифт">
<data/>
<acute/>
<shift/>
</font>
<font name="Minion Pro">
<data>и:-0,ы:130,И:-0,Ы:-0,а:-0,е:-0,о:-0,у:-0,э:-0,ю:-0,я:-0,А:83,Е:-0,О:-0,У:-0,Э:-0,Ю:-0,Я:-0</data>
```

```
<acute>PT Serif</acute>
<shift>0</shift>
</font>
<font name="Myriad Pro">
<data>и:-0,ы:121,И:-0,Ы:-0,а:-0,е:-0,о:-0,у:-0,э:-0,ю:-0,я:-0,А:0,Е:-0,О:-0,У:-0,Э:-0,Ю:-0,Я:-0</data>
<acute>PT Sans</acute>
<shift>0</shift>
</font>
</LetterTracking>
```

Итак, есть инструмент, с помощью которого, просто играючи, не напрягаясь, вы подберёте трекинги для всех шрифтов, не имеющих своих знаков ударения. Работа же должна быть в удовольствие, даже подготовка к ней.

А как этим реально распорядиться?

Есть три задачи, где фигурируют ударения:

- 1) поставить ударение над выбранной буквой;
- 2) удалить ударения в выбранной области;
- 3) заменить розовые квадратики потерянных ударений на полноценные знаки.

Ударение над буквой

Теперь это просто: выделяете гласную букву и запускаете скрипт **PlaceAccentMark.jsx**, который поставит над ней знак ударения.

Скрипт помещает знак ударения, символ с кодом 0x301 *после выделенной буквы*.

Скрипт разбирается, есть ли в используемой гарнитуре настоящий знак ударения нулевой ширины, и если есть, то ставит именно его.

А в случае, если в шрифте полноценного знака ударения нет, при попытке ввести знак с кодом 0x301 появится розовая заплатка, то ставится знак ударения на основании информации из файла LetterTracking(UTF-8).txt. В случае, если гласная прописная, знак ударения будет поднят, величина сдвига подобрана опытным путём, это кегль, делённый на 4,5. Знак ударения сдвинется вверх и в случае искусственного переключения строчных в прописные (All CAPS).

Если в xml-файле LetterTracking(UTF-8).txt нет информации о шрифте, скрипт сообщит об этом и оформит ударение первым шрифтом из массива fontForAccentA.

Знак ударения и настройки кернинга

Тестирование показало, что этот метод работает прекрасно только в случае, когда в оформлении выбранной буквы используется метрический кернинг.

В случае оптического кернинга позиция клинышка ударения может меняться, он заметно смещается вправо, и пара 'буква - знак ударения' смотрится некрасиво.

Поэтому в случае использования в работе оптического кернинга буква под ударением и следующая за ней будут иметь метрический кернинг. И сам знак ударения тоже оформляется метрическим кернингом.

Использование разных начертаний

Программа запоминает начертание буквы, над которой будет ударение, и присваивает его знаку ударения.

Это присвоение в режиме try-catch, т.е. выполняется попытка присвоения.

Она будет удачной, если было одно из четырёх начертаний Regular, Italic, Bold, Bold Italic.

Если начертание другое, то знак ударения будет оформлен как Regular.

Скрипт помогает расставлять ударения в обычном тексте, ячейках таблицы и сносках. Если в текущем шрифте есть такой знак, он будет поставлен. Если нет, знак ударения будет оформлен скриптом на основе данных из xml-файла.

Удаление из выделенного текста знаков ударения выполняет скрипт [DeleteAccentMark.jsx](#).

Удаляются все найденные знаки. Трекинг и кернинг ударной буквы и следующей после неё восстанавливаются по параметрам соседних букв.

Откат к предыдущему шагу

Если случайно отметили знаком ударения не ту букву, или убрали нужные ударения, клавиши **Ctrl+Z** вернут вёрстку к состоянию до запуска скрипта.

Результат в PDF и ePub

Файл с ударениями с использованием шрифта PT Serif будет безупречно смотреться, если сохранить его как PDF или ePub с фиксированным макетом.

Но если верстаемый документ предполагается превра-

тить в «перетекающий» формат ePub, то из-за того что читалки могут подставлять свои шрифты, вид ударения может измениться. Например, слово **школа** в вёрстке и PDF будет таким: **шкóла**, а в ePub версии 2 оно может выглядеть и вот так: **шко́ла**.

Это отдельная тема, как работать в вёрстке со знаками ударения, для перетекающего варианта ePub. Есть разные варианты решения данного вопроса. Похоже, все они решаются одинаково, сначала получают набор символов с гласными буквами, имеющими ударения, потом заменяют конкретные буквы, где требуется ударение на такую же букву из новой гарнитуры. Коды букв совпадают, отличается только вид.

И если книга будет в формате ePub, то этот шрифт должен быть загружен в файл, чтобы читалка показывала именно его, а не какой-то свой шрифт. В общем, это особый вопрос, выходящий за тему данного описания.

Тестовый файл

Чтобы понять особенности и возможности нового инструмента, надо попробовать его в деле. Именно для этого в архиве есть idml-файл с разными шрифтами,

различными вариантами набора текста — строчными, прописными, с переключением строчных в прописные (CAPS). Там везде метрический кернинг, и рекомендуется также попробовать установить оптический и посмотреть, как заметно сдвинутся знаки ударений.

На последней странице теста пример потерянных ударений.

Ударения в переданных на вёрстку файлах

Да, это сводящая многих с ума проблема, что делать с ударениями, появившимися после импорта

вордовского файла. Они выглядят как розовые прямоугольники или розовые полосы. Там должны были быть ударения, но их нет.

Или они есть в исходном файле, но как только текст оформляется другим шрифтом, вместо знаков ударений розовые пятна. Эта проблема, но в этом наборе скриптов есть и такой, что решает этот вопрос. Программа называется **RestoreAccentsView.jsx**, выделите текст, дальше она всё сделает сама, анализируя найденные символы и извлекая нужные данные из xml-файла. Буква под ударени-

ем и следующая за ней безусловно будут иметь метрический кернинг. Если встретится отсутствующий xml-файле шрифт, программа только один раз сообщит об этом, не на каждой букве.

Вот внизу скан из сети, он не к этим программам, просто изложен опыт автоматизации использования кода знака ударения. Да есть программы, где можно сделать горячие клавиши для всех гласных букв и знака ударения после неё, а потом ТЫНЦ, ТЫНЦ, ТЫНЦ...

Но если в шрифте такого знака ударения нет, никакого "тынц" не получится. И вряд ли стоит считать кривым шрифт только из-за того, что там нет правильного знака ударения.

Мне кажется не совсем эргономичный метод расстановки ударений. Я делал по-другому... В любом нормальном (некривом) шрифте уже есть знаки ударения как для ЗАГЛАВНЫХ букв так и для прописных. берете любой из скриптов Insert-A-Glyph'o-matic или GID_Injects или MultiplicationInjects или UnicodeInjector, и назначаете ему хоткей... Ставите курсор позади ударной гласной, ТЫНЦ по хоткею и все... 4 учебника русского для иностранцев по 300 полос каждый делал именно так.

Проблемные шрифты

В упомянутом ранее файле настроек 'AccentMarkData.jsxinc', общем для всех скриптов, есть строка `var fontsWithBadAccent = ["Palatino Linotype", "Open Sans", "Consolas"];`

Это шрифты с формально нормальными знаками ударения, но странно реализованными. В тексте есть объяснение, что там не так со знаками ударения. Если эти шрифты будут в работе, обработайте их предварительно *игровой приставкой* :)

Быстрый запуск

Очевидно, что теперь сложным будет не поиск в панели глифов нужного знака ударения, а отыскание нужного скрипта. Но если вы определите для него клавиатурное сокращение запуска, то маркировка буквы знаком ударения будет выполняться очень быстро.

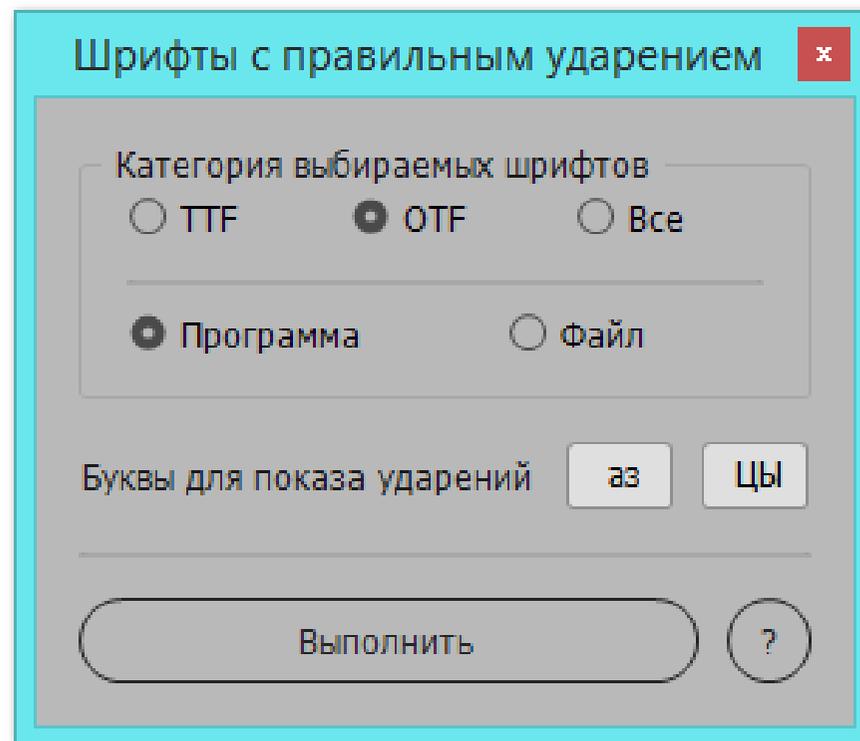
Как найти свободные горячие клавиши в индизайне, и назначить скрипту выбранный код, рассказано тут: <https://dotextok.ru/allscripts/besplatnye/poluchenie-spiska-nezanyatykh-goryachikh-klavish>

А какие ещё есть подходящие шрифты?

Теперь, когда вы поняли идею этого решения — в шрифтах, не имеющих своего знака ударения, использовать такой символ из другого шрифта — обязательно возникнет вопрос "Как быстро найти подходящий шрифт, если стандартный стартовый вариант не устраивает?"

Для решения этого вопроса написана программа **FindFontsWithRight0301Char.jsx**.

Вот её рабочее окно:



С её помощью вы соберете информацию о всех шрифтах, имеющих знак ударения нулевой ширины. Можно определить, какие файлы искать, только TTF, только OTF или те и другие; где искать — только на машине, или только в текущем файле. И поскольку важно не только знать название шрифта, но и видеть, как смотрятся знаки ударения над строчными и прописными буквами, вы определите и тестовые буквы для показа ударения в тексте.

Результат работы скрипта выглядит примерно так:

```
áз ЦЫ [ 1 ] [ 0 ] Calibri » Regular¶
áз ЦЫ [ 2 ] [ 8 ] PT Serif » Regular¶
áз ЦЫ [ 3 ] [ 9 ] PT Serif » Bold Italic¶
áз ЦЫ [ 4 ] [ 10 ] PT Serif » Bold¶
áз ЦЫ [ 5 ] [ 11 ] PT Serif Italic¶
áз ЦЫ [ 6 ] [ 12 ] Palatino Linotype » Regular¶
```

Итог

С этими инструментами вы с облегчением поставите жирный крест на потере нервов и времени, когда нужно поставить в любом шрифте знаки ударения.

И розовые пятна в пришедшем на вёрстку файле

отныне и навсегда будут быстро превращаться в нормальные знаки ударения.

Ведь вёрстка должна быть в кайф, согласны?

Расскажите о программе начальнику

Конечно, несколько напрягает вопрос, что это не бесплатно. Но можно ведь попробовать организовать дело так, чтобы этот комплект программ был куплен работодателем у ИП. Вот ссылки (две, на всякий случай) на заготовки трёх документов: договор, счёт, акт. Пусть юристы почитают, бывают особые моменты, учтём. А покупки у ИП обычно для бухгалтерии проблем не создают.

https://mega.nz/#!Y9w2VKol!Db4g-hxr63n0oScl7M_PeRAcy3NI9DvAGDacqjZQfqw

<https://www.dropbox.com/s/jlp1r0j3we5912z/Perenosyu%20v%20InDesign%20%5Bdogovor%2Bschiot%2Bakt%5D.zip?dl=0>

© Михаил Иванюшин, 2020
dotextok.ru | dotextok@gmail.com

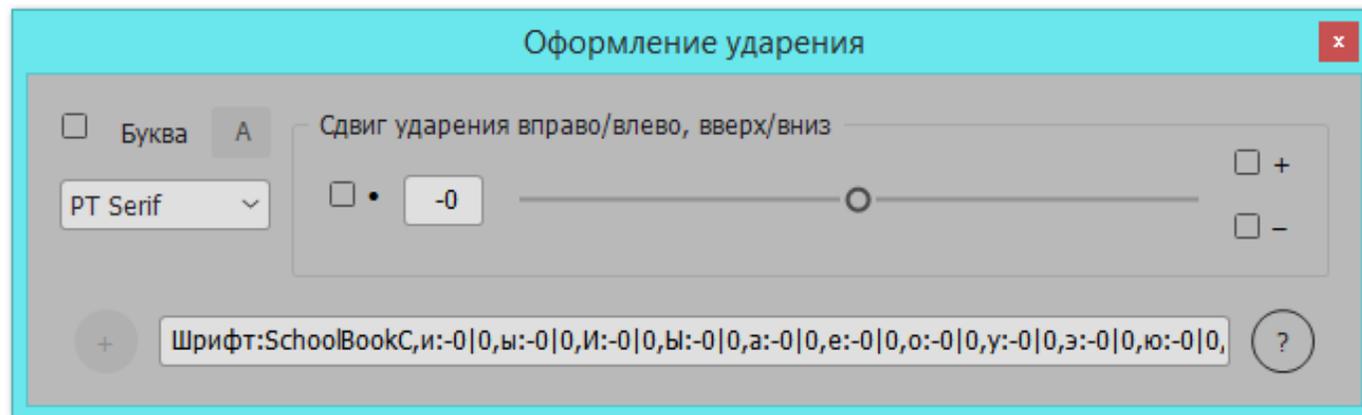
Обновление (17.06.2021)

Переписка с пользователями прояснила, что эта полезная программа будет ещё нужнее, если внести следующие изменения:

1) иметь возможность сдвигать знак ударения в обе стороны над буквой. В первом варианте можно было сдвинуть этот знак за левый край буквы, но нельзя было переместить за правый.

2) запоминать вертикальный сдвиг ударения для каждой буквы, а не усреднённое значение.

Всё это сделано.



Новое окно TestTrackValue.jsx

Вот как выглядит окно выбора положения знака ударения, когда новая буква только выбрана.

Ползунок слайдера размещён посередине, и движением вправо/влево определяется положение знака ударения. Диапазон изменения трекинга от -500 до $+500$.

Каждая гласная буква обрабатываемого шрифта теперь имеет два параметра, разделённые вертикальной чертой. Первое число это значение трекинга, второе — величина сдвига знака ударения по вертикали.

Если первое число записано так: -0 , это значит, что для данной буквы эти параметры ещё не подбирались. Величина сдвига знака ударения по вертикали — это *не число пунктов*, на которое будет смещён знак ударения. Это число шагов перемещения этого знака, и каждый шаг равен одной двадцатой кегля выбранной буквы. Такая привязка к размеру позволяет использовать эти данные для букв любого размера.

Новый формат XML-файла LetterTracking(UTF-8).txt

Поскольку теперь величина вертикального сдвига ударения запоминается для каждой буквы, то строка `<shift/>` не нужна.

Вот так выглядит этот файл в начале

```
<LetterTracking>
  <font name="Новый шрифт">
    <data/>
    <acute/>
  </font>
</LetterTracking>
```

и вот каким он был в процессе отладки новой версии, когда обрабатывались буквы шрифта Georgia:

```
<LetterTracking>
  <font name="Новый шрифт">
    <data/>
    <acute/>
  </font>
  <font name="Georgia">
    <data>и:-0|0,ы:-168|0,И:-0|0,Ы:-246|2,а:11|3,е:-0|0,
о:-0|0,у:-0|0,э:-0|0,ю:-0|0,я:-0|0,А:-72|3,Е:-0|0,О:-0|0,
```

```
У:-0|0,Э:-0|0,Ю:-0|0,Я:-0|0,ё:-41|0,е:-56|0,Ё:-53|2</data>
  <acute>PT Serif</acute>
</font>
</LetterTracking>
```

Используемый знак ударения — из шрифта PT Serif.

Стартовое значение гласных определено в переменной `var toSetValues = "и:-0|0,ы:-0|0,И:-0|0,Ы:-0|0,а:-0|0,е:-0|0,о:-0|0,у:-0|0,э:-0|0,ю:-0|0,я:-0|0,А:-0|0,Е:-0|0,О:-0|0,У:-0|0,Э:-0|0,Ю:-0|0,Я:-0|0";`

она находится в файле **#AccentMarkData.jsxinc**.

Как видите, букв ё, е и Ё в стартовом комплекте нет, они были добавлены в процессе работы.