

Практическая работа № 3 "Web-полиграфия. Ограничения в Web-полиграфии. Рекомендации по реализации"

Реализация Web-полиграфии

Полиграфия в Web полностью управляется с помощью CSS.

CSS (англ. *Cascading Style Sheets* — *каскадные таблицы стилей*) — **формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к любым XML-документам**

Используя CSS, вы получаете большие возможности контроля: не только над выбором размера, цвета и гарнитуры *шрифта*, но также над высотой строки, интервалом между буквами, и уровнем использования заглавных букв (все заглавные, начальные заглавные, малые заглавные, отсутствие заглавных вообще) и даже контроль над тем, как стилистически оформляется первая буква или первая строка текста.

При оформлении содержащего блок текста можно также контролировать уровень выравнивания текста и длины строки. Но не только это, вы должны также создавать правила стилового оформления только в одном месте — в таблице стилей — чтобы эти правила применить ко всему тексту, по всему Web-сайту (или можно быть более конкретным и нацелиться на определенные параграфы, или области на странице). Более того, если вдруг понадобится увеличить размер текста Web-сайта, или изменить *шрифт* основного текста, нужно будет изменить значение только в таблице стилей.

Ограничения в Web-полиграфии

Дизайнеры традиционной печати имеют в своем распоряжении огромное количество возможностей, когда речь идет о *полиграфии*, включая огромное число доступных *шрифтов*, и возможностей позиционирования текста. *Полиграфия* в Web значительно более ограничена, так как необходимо создавать дизайн, используя *шрифты* и позиционирование, и т.д., которые, как нам известно, будут доступны на компьютерах пользователей, которые будут просматривать Web-сайты — в Web бесполезно проектировать просто для себя.

Ограниченный выбор шрифтов

Ограниченный выбор *шрифтов* часто является первым вопросом, с которым вы столкнетесь при оформлении текста. Хотя в CSS можно определить любой *шрифт*, посетители сайтов увидят текст в этом виде, только если они имеют такой же *шрифт*, установленным на своем компьютере — если это не так, их браузер будет использовать либо альтернативный *шрифт*, который был определен в CSS, либо воспользуется значением по умолчанию браузера (обычно Times New Roman). Поэтому вы можете захотеть представить свой основной текст специальным *шрифтом*, таким как Trump Medieval или Avant Garde, но если ваша основная аудитория не связана тесно с дизайнерами, зрители вряд ли увидят текст в этом виде. В связи с этим большинство Web-дизайнеров ограничивают себя наиболее общедоступными на всех системах *шрифтами*, список которых обычно ограничен следующим набором:

- Andale Mono
- Times New Roman
- Georgia
- Verdana
- Arial/Arial Black
- Courier/Courier New
- Trebuchet MS
- Comic Sans
- Impact

Они выглядят следующим образом, как показано на рисунке:

Andale Mono	Courier New
Times New Roman	Trebuchet MS
Georgia	Comic Sans
Verdana	Impact
Arial	

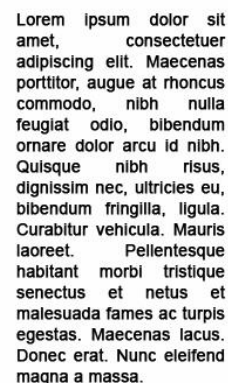
При использовании любого из приведенных выше *шрифтов* вы, скорее всего, выберите *шрифт*, который будут иметь большинство посетителей сайта. Компания Microsoft представила шесть новых *шрифтов*, созданных для использования на экране в Windows Vista и XP, и, по странному решению, выбрала все названия для них, начинающиеся с буквы С. Если вы захотите их использовать, то они называются Cambria, Calibri, Candara, Consolas, Constantia и Corbel. Сравнивая с тысячами *шрифтов*, доступных дизайнерам печатной продукции, дизайнеры Web могут надежно выбирать только примерно из десятка. Является ли это серьезным ограничением?

Полиграфия включает в себя значительно больше, чем только выбор привлекательного *шрифта*, она требует также рассмотрения длин строк, *кернинга* и пробелов — вспомните, что печатники до появления электронных *шрифтов* сталкивались с аналогичными ограничениями.

Переносы слов

Когда речь идет о выравнивании текста в пределах его контейнера, имеется четыре варианта: выравнивание слева, выравнивание справа, выравнивание по центру, и полное выравнивание. Полностью выровненный текст, когда правый и левый края блока текста выравниваются по вертикальным сторонам контейнера, может выглядеть более привлекательно, чем текст с "неровными" краями, и его можно видеть в большом количестве в журналах и книгах. Однако в сети Web это связано с некоторыми трудностями, в связи с отсутствием автоматического переноса слов, который разбивает слова в подходящем месте, чтобы лучше разместить их на строке. Чтобы полностью выровнять блок текста, браузеры могут только изменять интервалы между словами, что может приводить к "потокам пробелов", стекающих вертикально через блок — это обычно происходит, когда длина строки в блоке слишком короткая, и имеется недостаточно пространства, чтобы слегка изменить настройку, как показано на рисунке.

Как можно видеть на этом снимке с экрана, отсутствие переносов для разрыва слов в естественных точках создает пробелы между определенными словами, которые увеличиваются до неприемлемых размеров. Чтобы избежать этого необходимо использовать выравнивание текста слева для большей части Web.



>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas porttitor, augue at rhoncus commodo, nibh nulla feugiat odio, bibendum ornare dolor arcu id nibh. Quisque nibh risus, dignissim nec, ultrices eu, bibendum fringilla, ligula. Curabitur vehicula. Mauris laoreet. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Maecenas lacus. Donec erat. Nunc eleifend magna a massa.

Рис. Потоки пробелов могут портить блоки полностью выровненного текста

Кернинг

Кернинг является процессом настройки пробелов между определенными частями букв, когда используемый *шрифт* является пропорциональным (таким как Times New Roman, в котором пробел между отдельными символами изменяется от символа к символу), в отличие моноширинного (такого как Courier, где пробел между символами всегда один и тот же). *Кернинг* используется при печати для уменьшения пространства между буквами, которые размещаются естественным образом, такими как W, за которой следует A, и может создать для текста более профессиональный вид.

Большинство профессиональных *шрифтов* поставляются со встроенными командами *кернинга*, чтобы предоставить информацию о пробелах системе воспроизведения текста. Рисунок иллюстрирует различие, создаваемое *кернингом*.



WASP
WASP

На приведенном выше снимке с экрана первое слово не было подвергнуто *кернингу*. Однако второе слово имеет уменьшенный пробел между буквами W и A, в то время как пробел между A и S был слегка увеличен. В сети Web *кернинг* с таким уровнем точности фактически недоступен.

Единственное, что имеется, что в некоторой степени близко к *кернингу*, является возможность использовать трекинг, что в мире печати означает настройку пробела между символами в основном тексте, независимо от символов — поэтому можно будет уменьшить интервал между буквами W и A, но при этом также будет изменяться интервал между всеми другими буквами. В сети Web трекинг известен больше как интервал между буквами, и контролируется с помощью CSS — что показано на рисунке:



На приведенном выше снимке с экрана интервал между символами был увеличен на одинаковую величину. Хотя это помогло разделить A и S, интервал между W и A теперь стал слишком большим. Свойство интервала между буквами в CSS трудно использовать эффективно, в связи с его свойством либо все, либо ничего, и поэтому лучше использовать его с осторожностью.

Отсутствие контроля

Web не является печатью. Поэтому, там, где дизайнер печати не должен беспокоиться о том, что конечный зритель изменит размер текста, или может не иметь требуемый набор *шрифтов*, или не использовать требуемое подавление помех дискретизации, и часто существует искушение проверить определенный дизайн на зрителе — исправление, например, жесткого размера текста, или размещение текста в контейнерах фиксированного размера, или даже замена целых фрагментов текста изображениями. Однако отсутствие контроля не должно быть проблемой — необходимо просто привыкнуть к идее, что люди будут читать контент на множестве различных устройств в различных рабочих средах различным образом. Вы не должны пытаться остановить их, или создавать для них трудности — если они хотят прочитать ваш контент, то это должно делаться как можно проще. Они могут захотеть прочитать его на своем мобильном устройстве, когда они едут домой; они могут предпочесть распечатать и прочитать все на бумаге, а не на экране; они могут быть слабовидящими и нуждаться в некотором увеличении размера *шрифта*. Именно поэтому при стилевом оформлении текста в Web в действительности вы предоставляете указания для всех различных устройств просмотра, как вы предпочитаете представить текст.

РЕКОМЕНДАЦИИ

Выбор набора шрифтов

Хорошей практикой считается включение резервных вариантов выбора при определении предпочтительного *шрифта* вывода. Поэтому вместо того, чтобы просто определить "Georgia", можно задать "Georgia, Cambria, Times New Roman, Times, serif". В этом случае браузер сначала попытается использовать *шрифт* с именем Georgia, но если этот *шрифт* не установлен, он попытается использовать Cambria, затем Times New Roman, а затем Times, после чего тот *шрифт*, который операционная система задала для ключевого слова "serif".

Длина строки

Чтобы повысить удобочитаемость, средняя длина строки текста в охватывающем блоке должна быть около 40-60 символов на строку, хотя это может варьироваться в зависимости от аудитории (дети предпочитают более короткие строки, взрослые предпочитают более длинные). Идеальная длина строки показана на рисунке:

Текст на снимке с экрана содержит примерно 60 символов на строку. Немного больше этого значения, и читатель начнет двигать глазами — или даже головой — чтобы следовать за текстом, что может увеличить напряжение на глаза и сделать текст более трудным для восприятия.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas porttitor, augue at rhoncus commodo, nibh nulla feugiat odio, bibendum ornare dolor arcu id nibh. Quisque nibh risus, dignissim nec, ultricies eu, bibendum fringilla, ligula. Curabitur vehicula. Mauris laoreet. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Maecenas lacus. Donec erat. Nunc eleifend magna a massa.

Рис. 60 символов на строку - идеальная длина строки

Высота строки

Высота строки относится к вертикальному пространству между строками, и можно сделать свой *шрифт* более читаемым, увеличивая его немного относительно используемого в браузере значения по умолчанию (что также создает дополнительное пространство для символов нижних и верхних индексов) — сравните два параграфа на рисунке:

Первый параграф на приведенном выше снимке с экрана имеет высоту строки по умолчанию, и может казаться немного стесненным. Второй параграф имеет увеличенную высоту строки, и текст имеет немного больше пространства для дыхания, что несколько облегчает его восприятие. Однако слишком большая высота строки может снова затруднить восприятие, так что будьте внимательны.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas porttitor, augue at rhoncus commodo, nibh nulla feugiat odio, bibendum ornare dolor arcu id nibh. Quisque nibh risus, dignissim nec, ultricies eu, bibendum fringilla, ligula.

Curabitur vehicula. Mauris laoreet. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Maecenas lacus. Donec erat. Nunc eleifend magna a massa.

Рис. Высота строки может создать большое различие во внешнем виде текста

Буквицы

Определяя псевдо-элемент первой буквы как `p:firstletter { }`, можно представить первую букву отлично от остальных — такое оформление называется обычно буквицей, когда первая буква занимает примерно 3-4 строки текста — см. рисунок.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas porttitor, augue at rhoncus commodo, nibh nulla feugiat odio, bibendum ornare dolor arcu id nibh. Quisque nibh risus, dignissim nec, ultricies eu, bibendum fringilla, ligula.

Капитель

Часто *шрифты* поставляются с вариантом капители — множеством прописных букв, которые находятся в верхнем регистре, но имеют примерный размер варианта строчной буквы. Это полезно в тех случаях, когда вы хотите записать что-то прописными буквами, но не хотите привлекать к этому слишком много внимания, например, это может использоваться для сокращений.

Даже если система не имеет варианта капители указанного *шрифта*, все будет нормально — браузер сгенерирует свою собственную версию, используя полноразмерные прописные символы и затем сокращая символы примерно на 70-80% от их обычного размера. Рисунок показывает разницу между полноразмерными прописными буквами и капителью.

**ABANDON HOPE
ALL YE WHO ENTER HERE**

Висящая пунктуация

Можно использовать хороший полиграфический эффект, если предложение начинается со знака *кавычки*. Свойство CSS (`text-indent`) для отступа текста с отрицательным значением — применяя значение в `em` (-10em), пунктах (-10pt), пикселях (-10px) или процентах (-10%) — позволяет сдвинуть символ *кавычки* влево, сохраняя левую вертикальную линию блока текста, как показано на рисунке:

“Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas porttitor, augue at rhoncus commodo, nibh nulla feugiat odio, bibendum ornare dolor arcu id nibh.”

“Quisque nibh risus, dignissim nec, ultricies eu, bibendum fringilla, ligula.”

Рис. Висящая пунктуация

Полиграфически-правильная пунктуация и другие вопросы

Можно сделать свой текст более профессиональным и элегантным, используя множество имеющихся полиграфических объектов HTML, таких как "кривые" *кавычки* и короткие (-) и длинные (—) тире. Большая часть программного обеспечения поддержки блогов и обработки текста автоматически делают это во время ввода текста, превращая обычные прямые *кавычки* в полиграфически правильный изогнутый вариант, и превращая строки тире в короткие или длинные тире. На рисунке можно видеть примеры полиграфически правильной пунктуации.

“This paragraph not only has curly quotes but also — yes! — typographically-correct dashes as well.”

Если вы начнете улучшать свой основной текст развитой пунктуацией, текст может выглядеть значительно более элегантным и профессиональным — в большей степени похожим на текст книги или журнала, чем на текст в сети. Помните, однако, что такая пунктуация может выглядеть немного пикселизированной для людей с более старыми экранами или при отключении сглаживания, поэтому старайтесь использовать ее с осторожностью.

Кроме того, существуют объекты, задаваемые специальным кодом HTML, которые можно вставлять в основной текст для генерации специальных символов, обычно отсутствующих на клавиатуре. Рисунок показывает ряд таких объектов:

How many entities can we fit in one paragraph? Well, we could show you a ™ symbol, an @ symbol, a © symbol, a • point, a € currency symbol, a ¥ symbol...and there goes a typographically-correct ellipsis!

And don't forget accented characters, either.

Их можно ввести вручную, но часто программное обеспечение управления контентом может легко преобразовать или вставить такие объекты.

Рис. Объекты HTML

Врезки

Врезки являются небольшими цитатами из текста, которые размещают на странице, используя символы большего размера, и иногда другой *шрифт*, чтобы привлечь к нему внимание. Вы видели их почти в любом журнале, который читали, и они являются эффективным способом разбиения текста и выделения ключевых мыслей или фраз — и они также легко создаются в Web с помощью простой разметки и стилевого оформления. Сделайте просто текст больше, задайте, возможно, другой *шрифт*, разместите его так, чтобы обычный текст обтекал его, и все готово. Существует также несколько более развитых решений, которые включают выбор выделенного текста с помощью JavaScript и автоматическое заполнение с его помощью врезки, что может исключить необходимость дважды писать один и тот же текст в коде разметки. Если вы хотите вставить символ `copyright` в основной текст, вы используете объект HTML. Сделайте просмотр страниц в Интернет и попробуйте найти все другие объекты HTML. Существует около 250 таких объектов.

Контрольные вопросы

- В чем различие между *кернингом* и *трекингом*, и что из них доступно для Web-дизайнера?
- Как можно избежать "потоков пробелов" стекающих по тексту?
- Назовите четыре различных типа использования прописных букв, доступных с помощью CSS.
- Какая длина строки будет оптимальной для основного текста, и какие факторы могут влиять на это?
- В чем различие между *шрифтом* с засечками и без засечек? Приведите пример каждого из них.
- Чем *висящая* пунктуация отличается от обычной пунктуации?