### Вёрстка текстового содержимого страниц

Выполнен на 0%

В Обзор основных тем и элементов

В Углублённая теория Методология вёрстки текстовых элементов

Основные элементы текста — Заголовки Текст и контрастность — Лид — Цитаты Врезки, обтекания — Списки Метаданные, дата, автор

— Подписи к демонстрационному материалу Ссылки, подчёркивания

 Акценты внутри текста — Направление текста и трансформация Дополнительные элементы текста и декорирование отдельных слов

— Использование верхнего и нижнего индекса для формул и не только — Таблицы Колонки на странице

 Переполнение текстом, однострочное и многострочное

Декоративные элементы Перенос строк и слов

— Буквицы и красная строка Вставка знака рубля Сложные обтекания

Кейс 1, лёгкий уровень № Кейс 2, лёгкий уровень

Прозрачные буквы

№ Кейс 3, лёгкий уровень

№ Кейс 4, средний уровень № Кейс 5, средний уровень

Кейс 7, сложный уровень № Кейс 8, сложный уровень

Кейс 6, средний уровень

🖸 Кейс 9, сложный уровень

Главная / Моё обучение / Вёрстка текстового содержимого страниц / Методология вёрстки текстовых элементов Дополнительные элементы текста и декорирование отдельных слов

# Переполнение текстом, однострочное и многострочное

не раздвигать соседние элементы? Например, если лид новости выводится на общей странице новостей и он слишком велик для своего поля?

Что сделать, чтобы текст помещался в оставленное для него пространство или хотя бы обрезался так, чтобы

Если контейнер, содержащий текст, имеет конкретные размеры, то судьба того, что за эти размеры вылезает, решается свойством overflow. По умолчанию у свойства значение visible, и контент выходит за границы контейнера и может находить на соседние блоки. Со значением overflow: hidden всё, что не поместилось, просто обрежется. Значение auto в случаях переполнения контейнера выводит полосу прокрутки. Если полоса прокрутки нужна всегда, выбирается overflow: scroll. Свойство overflow обрабатывает действия при переполнении контентом по горизонтали и вертикали. Если нужно управлять одним из направлений, используется overflow-х для горизонтальной оси и overflow-у для вертикальной.

Однострочный длинный текст

Теперь рассмотрим другие возможные случаи переполнения текстом и способы их обработки.

## CSS-свойство text-overflow применяется для управления переполнением текста. У свойства есть два основных

Было бы лучше посвятить то время еде и сну.

значения: clip для обрезания содержимого, которое не поместилось, и ellipsis для подстановки многоточия вместо невошедшей части. Значение text-overflow: ellipsis обычно и применяется в таких ситуациях, и многоточие возникает в самом лучшем случае после случайного слова. То есть смысл фразы может быть безвозвратно испорчен.

Однажды я провёл в размышлениях целый день без еды... Фраза должна была быть гораздо длиннее, но контейнер однострочный, фраза не влезла, смысл высказывания остаётся неясным

Вёрстка элемента

<р>Однажды я провёл в размышлениях целый день без еды и целую ночь без сна, но я ничего не добился.

```
p {
 width: 580px;
  font-size: 24px;
  line-height: 32em;
  overflow: hidden;
  white-space: nowrap;
  text-overflow: ellipsis;
```

добился. Было бы лучше посвятить то время еде и сну">Однажды я провёл в размышлениях целый день без еды и целую ночь без сна, но я ничего не добился. Было бы лучше посвятить то время еде и сну.

на компьютерах при наведении курсора на «сокращённый» текст будет появляться подсказка с полным текстом.

Для удобства пользователя элементу можно добавить атрибут **title** с полным текстом. В таком случае

Многострочный длинный текст Для случая, если необходимо поместить в ограниченную область не одну строку, а несколько, не всё так просто.

#### CSS-свойство text-overflow: ellipsis для таких элементов не работает. И тут приходиться использовать все возможные «костыли» и «хаки».

Замечание

Обрезание после определённой строки

У этого свойства есть ряд ограничений: оно работает не во всех браузерах и не включено в официальную

Для этого существует CSS-свойство -webkit-line-clamp. Свойство отрезает строки после указанного их числа.

спецификацию. Кроме того, свойство работает только в паре с другим устаревшим проприетарным свойством

display: -webkit-box. И ещё такой метод может оборвать строку в неудачном месте, и смысл высказывания

# может поменяться.

lastline. Свойство отрезает часть строки снизу.

Например, для длинного текста, который должен уместиться в три строки, можно использовать следующие стили:

В старых версиях браузера *Opera* также работает проприетарное значение text-overflow: -o-ellipsis-

eaque reiciendis libero possimus et aperiam consequuntur numquam atque sequi fugit aut voluptatum ad, id dignissimos nisi porro veritatis, accusantiu...

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Modi ullam amet

Применили свойство -webkit-line-clamp к стандартной рыбе Вёрстка элемента

```
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Modi ullam amet eaque reiciendis
   libero possimus et aperiam consequuntur numquam atque sequi fugit aut voluptatum ad, id dignissimos
   nisi porro veritatis, accusantium neque obcaecati ut repudiandae, quisquam. Similique, quam dolores
   voluptatibus enim architecto recusandae molestiae soluta. Explicabo ex, reiciendis delectus
   quisquam, fugiat iusto tempore maiores alias voluptates ratione ipsum ad cupiditate, libero vero
   cumque dolorum quaerat.
   p {
    width: 780px;
     overflow: hidden;
     padding: 0.5em;
     display: -webkit-box;
     text-overflow: ellipsis;
     line-height: 1.5em;
     -webkit-line-clamp: 3;
     -webkit-box-orient: vertical;
Скрытие лишнего текста за декоративными элементами
```

### Lorem ipsum dolor sit amet, consectetur adipisicing elit. Modi ullam amet eaque reiciendis libero possimus et aperiam consequuntur numquam atque

<div class="module fade">

</div>

выглядеть так:

margin: 0 0 1em 0;

overflow: hidden;

nisi porro veritatis, accusantium

и часть символов последней поместившейся строки уходит в непрозрачную область.

sequi fugit aut voluptatum ad, id dignissimos nisi p Градиент перекрывает последние слова строки. Настройки зависят от параметров шрифта

В этом методе с помощью псевдоэлемета ::after добавляется полупрозрачный слой с градиентной заливкой

Вёрстка элемента

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Modi ullam amet eaque reiciendis

libero possimus et aperiam consequuntur numquam atque sequi fugit aut voluptatum ad, id dignissimos

```
.module {
     width: 780px;
     margin: 0 0 1em 0;
     overflow: hidden;
     font-family: "Merriweather", "Georgia", serif;
     font-style: normal;
     font-weight: normal;
     font-size: 20px;
     line-height: 32px;
     color: #333333;
   .fade {
     position: relative;
     height: 5.9em; /* высота в три строки */
   .fade::after {
     content: "";
     position: absolute;
     bottom: 0;
     right: 0;
     width: 70%;
     height: 1.2em;
     text-align: right;
     background: linear-gradient(to right, rgba(255, 255, 255, 0),
   rgba(255, 255, 255, 1) 50%);
Полупрозрачный декоративный элемент может размещаться по всей ширине нижней границы, тогда текст будет
```

 Вёрстка элемента .module { width: 780px;

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Modi ullam amet

eaque reiciendis libero possimus et aperiam consequuntur numquam atque

Градиент частично перекрывает нижнюю строку. Настройки зависят от параметров шрифта

sequi fugit aut voluntatum ad. id dignissimos nisi porro veritatis, accusantium

```
font-family: "Merriweather", "Georgia", serif;
     font-style: normal;
     font-weight: normal;
     font-size: 20px;
     line-height: 32px;
     color: #333333;
   .fade {
     position: relative;
     height: 5.9em; /* высота в три строки */
   .fade::after {
     content: "";
     position: absolute;
     bottom: 0;
     right: 0;
     left: 0;
     width: 100%;
     height: 1.2em;
     text-align: right;
     background: linear-gradient(to bottom, rgba(255, 255, 255, 0),
   rgba(255, 255, 255, 1) 70%);
Градиент завязан на размеры шрифта, высоту строки, а также внутренние и внешние отступы. При их изменении
нужно пересчитать значение CSS-свойства высоты области (в примерах — с классом fade), а также поправить
значения CSS-свойств для псевдоэлемента ::after.
```

Использование JavaScript В этом способе предполагается, что контейнер для длинного текста имеет известную фиксированную высоту, и в ходе выполнения JavaScript рассчитывается, какой текст помещается в необходимую область, а всё что не поместилось — отбрасывается и добавляется символ ..... Ho JS не выполняется сразу при показе страницы и может произойти «мигание», когда вёрстка уже отобразилась,

а затем в течение некоторого времени отрабатывает скрипт, подрезающий текст. Подрезки в принципе

не произойдёт, если JS отключён или не догрузился. К тому же расчёт, который выполняется в скрипте,

## В этом случае с сервера отдаётся уже готовая вёрстка с отрезанной частью текста, сам расчёт длины строки и места отреза производится в бэкенд-коде. Расчёт обычно проводится по количеству символов, а это значит, что

Обрезка текста на веб-сервере

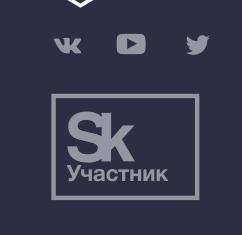
увеличивает нагрузку и может привести к подвисанию сайта.

слово может быть обрезано в середине. Сделать «умный расчёт» возможно, но это, конечно, усложнит задачу. Способ не универсальный, так как интерфейс может измениться, а в этом случае придётся менять и серверную часть сайта.

Сохранить прогресс

Курсы

Ознакомились со статьёй?



Подписка Для команд и компаний Учебник по РНР Профессии Фронтенд-разработчик

Бэкенд-разработчик

Работа наставником

Для учителей

Стать автором

Услуги

Практикум

Тренажёры

React-разработчик Фулстек-разработчик

**К**олонки на странице

JavaScript. Профессиональная разработка веб-интерфейсов JavaScript. Архитектура клиентских приложений React. Разработка сложных клиентских приложений РНР. Профессиональная веб-разработка PHP и Yii. Архитектура сложных веб-сервисов

Анимация для фронтендеров

Vue.js для опытных разработчиков

Регулярные выражения для фронтендеров

Вёрстка email-рассылок

Шаблонизаторы HTML

Анатомия CSS-каскада

© ООО «Интерактивные обучающие технологии», 2013-2022 **</>/> → ♥ →** ₩

HTML и CSS. Профессиональная вёрстка сайтов

HTML и CSS. Адаптивная вёрстка и автоматизация

Node.js. Разработка серверов приложений и API

Шпаргалки для разработчиков Отчеты о курсах Информация Об Академии

Декоративные элементы

Остальное Написать нам Мероприятия Форум

О центре карьеры

Блог

С чего начать

Алгоритмы и структуры данных