

CSS: о выводе коротких и длинных текстов

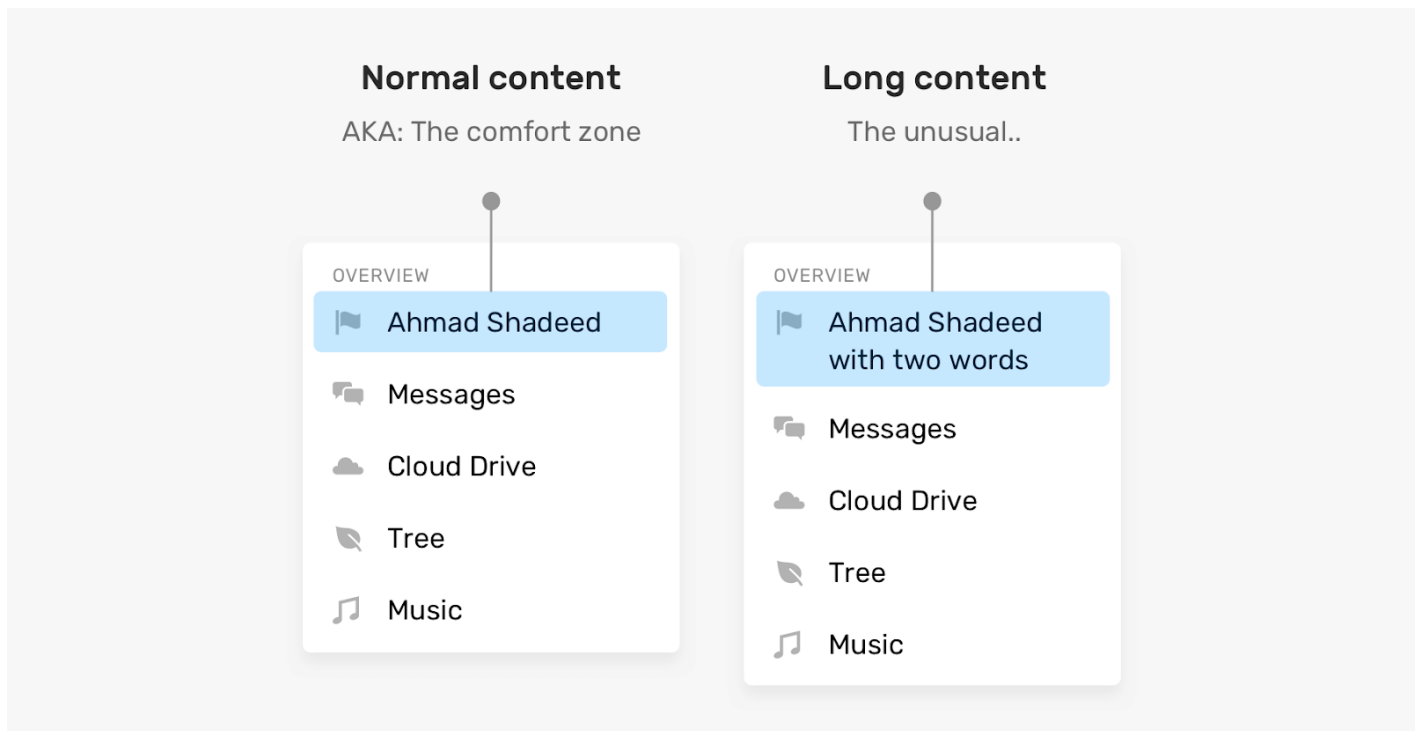
Когда, пользуясь возможностями CSS, создают макет страницы, важно учитывать то, что в различных элементах этой страницы могут выводиться короткие и длинные текстовые материалы. Страницы, кроме того, нужно тестировать на предмет того, как они отображают тексты разной длины. Если разработчик чётко понимает то, как обрабатывать различные тексты, выводимые на странице, если он соответствующим образом спроектировал макет, это способно избавить его от множества неприятных неожиданностей.



Есть много ситуаций, в которых изменение некоего текстового фрагмента путём добавления или удаления всего одного слова способно заметно изменить внешний вид страницы, или, что ещё хуже «поломать» макет и сделать невозможной нормальной работу с сайтом. Когда я только начинал изучать CSS, я недооценивал последствия, к которым может привести добавление единственного слова в некий элемент или удаление из него всего одного слова. Здесь я хочу поделиться различными способами обработки текстов разной длины средствами CSS.

Обзор проблем

Прежде чем я расскажу о том, как работать с текстами средствами CSS, мне хотелось бы остановиться на проблемах, связанных с выводом текстов. Предположим, у нас есть вертикальное меню.



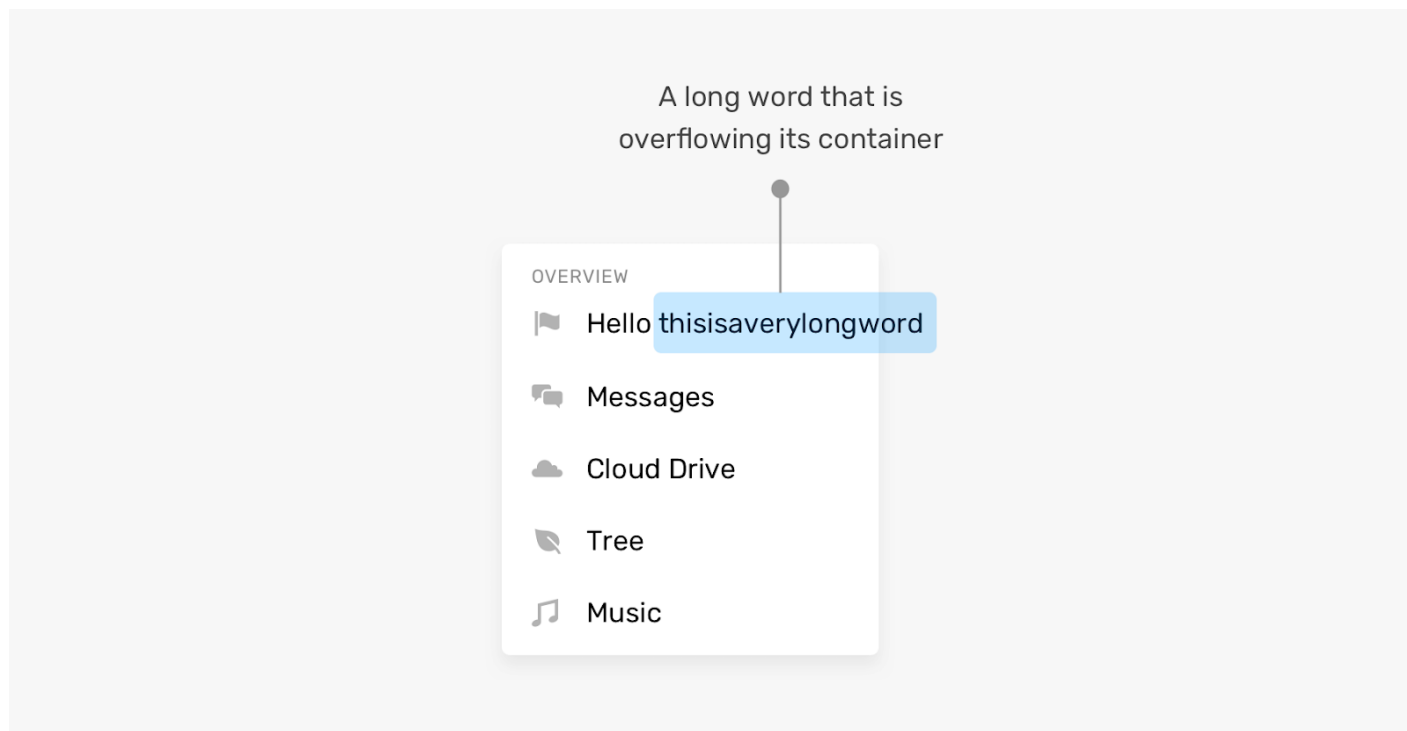
Вертикальное меню при переносе имени пользователя на вторую строку выглядит необычно

В начале меню выводится имя пользователя. Длина имени может варьироваться. Особенно — если речь идёт о мультязычных сайтах. В правой части вышеприведённого примера видно, что имя пользователя, при достижении им определённой длины, занимает две строки. По этому поводу можно задаться несколькими вопросами:

- Нужно ли в такой ситуации обрезать текст?
- Нужно ли размещать текст в нескольких строках? Если да — то каково максимальное количество таких строк?

Это — пример того, что происходит, если в некоем элементе выводится такое количество слов, которое превышает количество, на которое рассчитывал разработчик. А что если некое слово просто оказывается очень длинным? Если используются настройки,

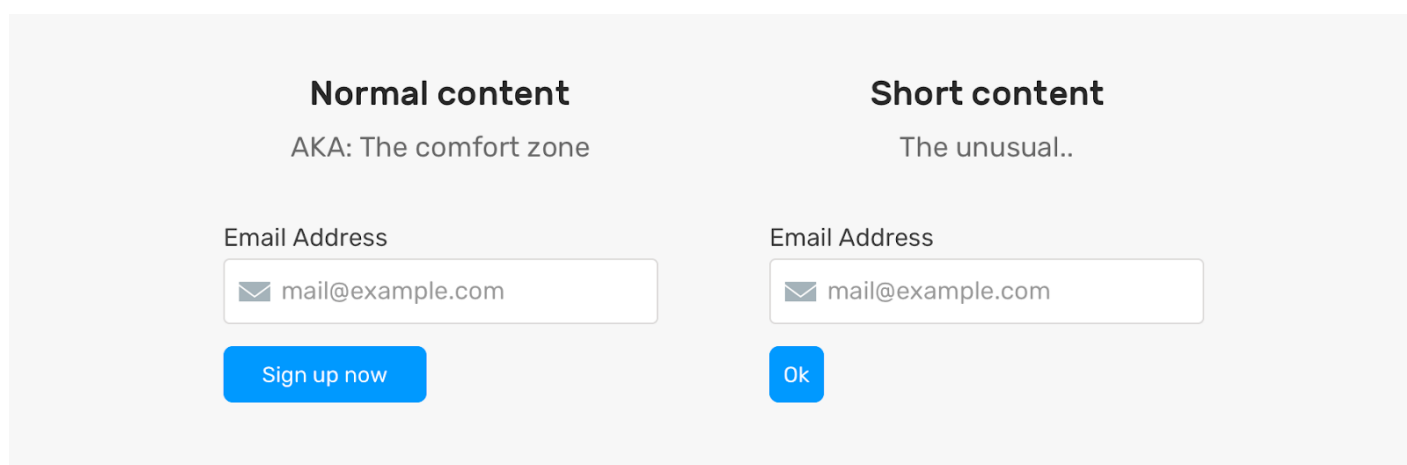
применяемые по умолчанию, то такое слово просто выйдет за пределы своего контейнера.



Слово вышло за пределы контейнера

Фронтенд-разработчик должен быть готов к такому, заранее приняв решение о том, как страница должна вести себя в подобных ситуациях. К счастью, существуют CSS-свойства, созданные специально для того чтобы решать подобные проблемы.

Кроме того, проблемы на страницах могут вызывать не только слишком длинные, но и слишком короткие тексты. Вывод короткого текста в элементе, не рассчитанном на такой текст, может либо «сломать» интерфейс, либо, как минимум, сделать так, что этот элемент будет странно выглядеть. Вот пример.



Кнопка, в которой выводится слишком короткий текст, выглядит необычно

Проблема тут в том, что кнопка, в которой выводится текст `Ok`, оказывается очень короткой. Я не говорю о том, что это — страшная проблема, но выглядит подобная кнопка не очень хорошо. Её, к тому же, в определённых ситуациях, может быть сложно найти на странице.

Что делать? Возможно, стоит настроить свойство кнопки `min-width`. Благодаря этому она сможет нормально выводить подписи разной длины.

Как видите, проблемы могут возникать как при выводе длинных, так и при выводе коротких текстов. Но, прибегнув к некоторым возможностям CSS, мы можем, по меньшей мере, ослабить влияние этих проблем на внешний вид и работоспособность страниц.

Длинные тексты

Теперь, когда мы обсудили проблемы, поговорим о возможностях CSS, которые позволяют наладить нормальную работу с длинными текстами.

Свойство `overflow-wrap`

Свойство `overflow-wrap` позволяет сообщить браузеру о том, что он должен разорвать слово, перенеся его на новую строку, в том случае, если слово не помещается в контейнер.

```
.card {  
  overflow-wrap: break-word;  
}
```

Without overflow-wrap

This is a test to see how the overflow wrap works in CSS. This longwordcantfit

```
.card {  
  /* without wrapping */  
}
```

With overflow-wrap

This is a test to see how the overflow wrap works in CSS. This longwordcantfit

```
.card {  
  overflow-wrap: break-word;  
}
```

Без использования свойства overflow-wrap слово выходит за пределы контейнера

Свойство hyphens

Значение `auto` CSS-свойства `hyphens` позволяет сообщить браузеру о том, что он должен самостоятельно принять решение о разделении длинных слов с использованием дефиса и о переносе их на новые строки. Это свойство может принимать и значение `manual`, что позволяет, используя особые символы, предусмотреть возможность и порядок переноса слова на новую строку в том случае, если в этом возникнет необходимость.

```
.element {  
  hyphens: auto;  
}
```

Without hyphens

This is a test to see how the overflow wrap works in CSS. This longwordcantfit

```
.card {  
  /* without hyphens */  
}
```

With hyphens

This is a test to see how the overflow wrap works in CSS. This longwordcantfit

```
.card {  
  hyphens: auto;  
}
```

Без использования свойства `hyphens` браузер не переносит слово на новую строку

Применяя значение `auto` свойства `hyphens` важно помнить о том, что браузер будет переносить любое слово, которое не помещается в строку. Что это значит? Взгляните на следующий рисунок.

Without hyphens

This is a test test to see how the hyphens cool property will handle different word lengths.

```
.card {  
  /* without hyphens */  
}
```

With hyphens

This is a test test to see how the hyphens cool property will handle different word lengths.

```
.card {  
  hyphens: auto;  
}
```

Браузер может использовать знак переноса в любом слове

Обратите внимание на то, что браузер использовал знак переноса в слове, которое вполне может быть целиком перенесено на новую строку. При использовании свойства `hyphens: auto` браузер способен разрывать даже такие слова, которые не выходят за пределы контейнеров.

Обрезка однострочного текста

При обрезке текста, для вывода которого предусмотрено поле, вмещающее лишь одну строку, предложение укорачивается, а в его конец добавляется многоточие, указывающее на то, что текст, на самом деле, длиннее того фрагмента, который выведен на экране.

Without truncation

This is a cool test for
CSS truncation.

```
.card {  
  /* without hyphens */  
}
```

With truncation

This is a cool test fo...

```
.card p {  
  white-space: nowrap;  
  overflow: hidden;  
  text-overflow: ellipsis;  
}
```

Слева — однострочный текст, при выводе которого обрезка не используется. Справа — текст, при выводе которого используется обрезка

В CSS нет свойства, которое могло бы называться «text-truncation», применимого для настройки автоматической обрезки текстов. Тут нам понадобится комбинация из нескольких свойств:

```
.element {  
  white-space: nowrap;  
  overflow: hidden;  
  text-overflow: ellipsis;
```

```
}
```

Обрезка многострочного текста

Если нужно обрезать текст, для вывода которого предусмотрено поле, вмещающее несколько строк, нужно прибегнуть к CSS-свойству `line-clamp`:

```
.element {  
  display: -webkit-box;  
  -webkit-line-clamp: 3;  
  -webkit-box-orient: vertical;  
  overflow: hidden;  
}
```

Для того чтобы это сработало, необходимо использовать и свойство `display: -webkit-box`. Свойство `-webkit-line-clamp` позволяет указать максимальное количество строк, по достижении которого текст надо обрезать.

Single line truncation

This is a cool test fo...

```
.card p {  
  white-space: nowrap;  
  overflow: hidden;  
  text-overflow: ellipsis;  
}
```

Multiline truncation

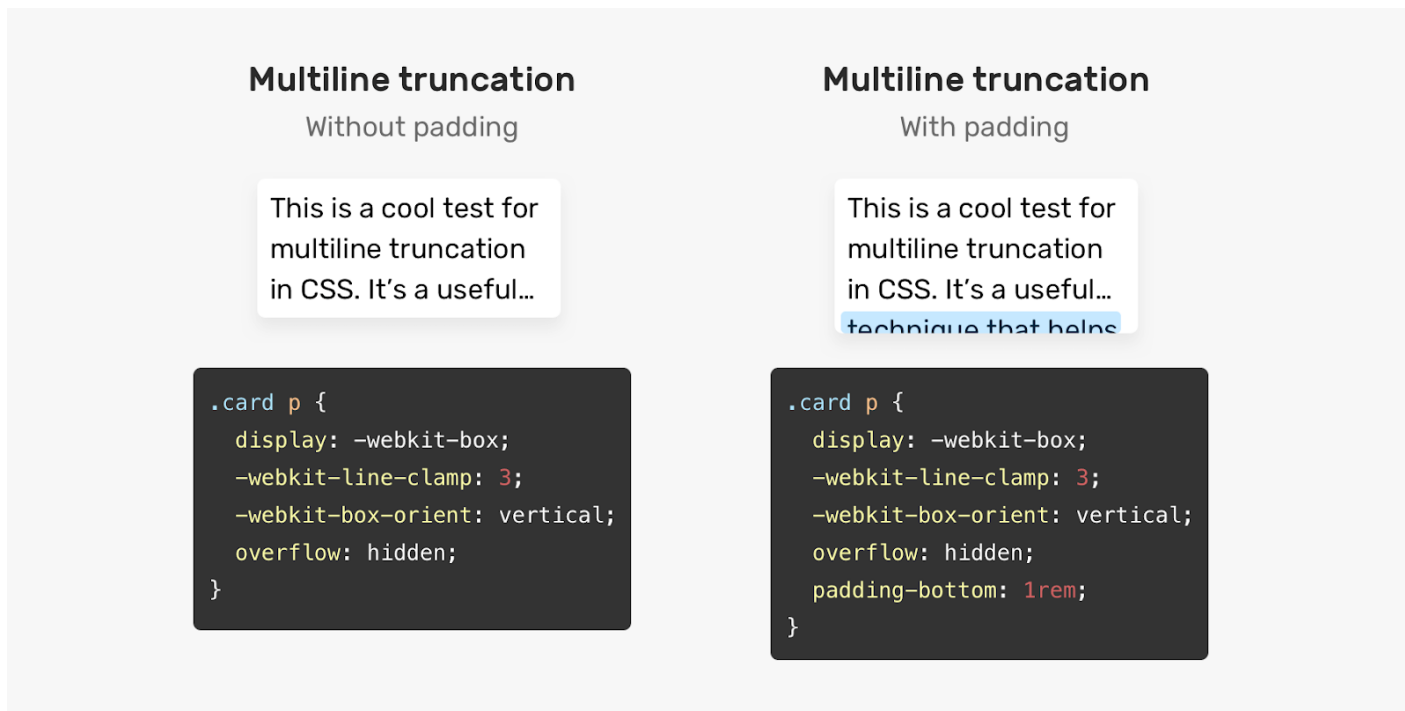
This is a cool test for
multiline truncation
in CSS. It's a useful...

```
.card p {  
  display: -webkit-box;  
  -webkit-line-clamp: 3;  
  -webkit-box-orient: vertical;  
  overflow: hidden;  
}
```

Сравнение обрезки однострочного и многострочного текста

Минус этого подхода заключается в том, что, если у элемента будет настроено свойство `padding`

, нормальный вывод текста может быть легко нарушен. Настройка этого свойства приведёт к тому, что часть текста, которая, как ожидается, должна быть скрыта, окажется выведенной после обрезанного текста.



Настройка свойства padding приводит к нарушению вывода текста

Вывод длинных текстов в полях, поддерживающих горизонтальную прокрутку

В некоторых ситуациях непрактично организовывать разрыв слов или их перенос на новые строки с использованием дефиса. Например, если имеется поле, в котором должен выводиться фрагмент JavaScript-кода, такой код будет тяжело читать в том случае, если слова будут выводиться с переносом на новые строки. В подобном случае облегчить работу с текстом можно, предусмотрев возможность горизонтальной прокрутки содержимого поля.

```
.code {  
  overflow-x: auto;  
}
```

Essential things to think about before starting a blog

It takes several ingredients to create a delicious blog.

It has been exactly 3 years since I wrote my first blog series entitled “Flavorful Tuscany”, but starting it was definitely not easy. Back then, I didn’t know much about blogging, let alone think that one day it could become my full-time job. Even though I had many recipes and food-related stories to tell, it never crossed my mind that I could be sharing them with the whole world.

I am now a full-time blogger and the curator of the Simply delicious newsletter, sharing stories about traveling and cooking, as well as tips on how to run a successful blog.

If you are tempted by the idea of creating your own blog, please think about the following:

Your story (what do you want to tell your audience)

Your audience (who do you write for)

Your blog name and design

After you get your head around these 3 essentials, all you have to do is grab your keyboard and the rest will follow.

Поле, в котором осуществляется перенос слов на новые строки, и поле, в котором применяется горизонтальная прокрутка

Свойство padding

В некоторых случаях, когда у элемента не настроено свойство `padding`, вспоминают об этом лишь встречаясь с проблемами, появляющимися при выводе данных на страницах.

Взгляните на следующий пример.

Sign up for newsletters

☐ The Daily Crunch

☐ Week in Review

☐ Startups Weekly

☐ Extra Crunch
Announcements

☐ dksodskodsodsok
Events

☐ Extra Crunch
Roundup

 mail@example.com

Sign up now

Проблема при выводе подписи к флажку

Тут имеется набор флажков. Подпись одного из них выводится слишком близко к другому. Причина этого заключается в том, что при проектировании макета, используемого для вывода флажков, не настроены промежутки между ячейками сетки, в которых размещены данные. Этот пример, кстати, взят с реального сайта (Techcrunch).

Короткие тексты

Я знаю о том, что проблемы, связанные с короткими текстами распространены не так сильно, как проблемы, связанные с длинными текстами. Но возможность их возникновения, всё равно, очень важно учитывать при проектировании пользовательских интерфейсов.

Установка минимальной ширины элемента

Вернёмся к примеру, который я приводил в начале статьи.

Normal content	Short content
AKA: The comfort zone	The unusual..
Email Address	Email Address
<input type="text" value="mail@example.com"/>	<input type="text" value="mail@example.com"/>
<button>Sign up now</button>	<button>Ok</button>

Кнопка, в которой выводится слишком короткий текст

Как справиться с проблемой, возникающей при выводе на кнопке очень короткой надписи? Решить эту проблему можно, воспользовавшись свойством `min-width`. При таком подходе ширина кнопки, даже при выводе в ней короткой надписи, не будет меньше заданного значения.

Short content
Enhanced
Email Address
<input type="text" value="mail@example.com"/>
<button>Ok</button>
<pre>.button { min-width: 80px; }</pre>

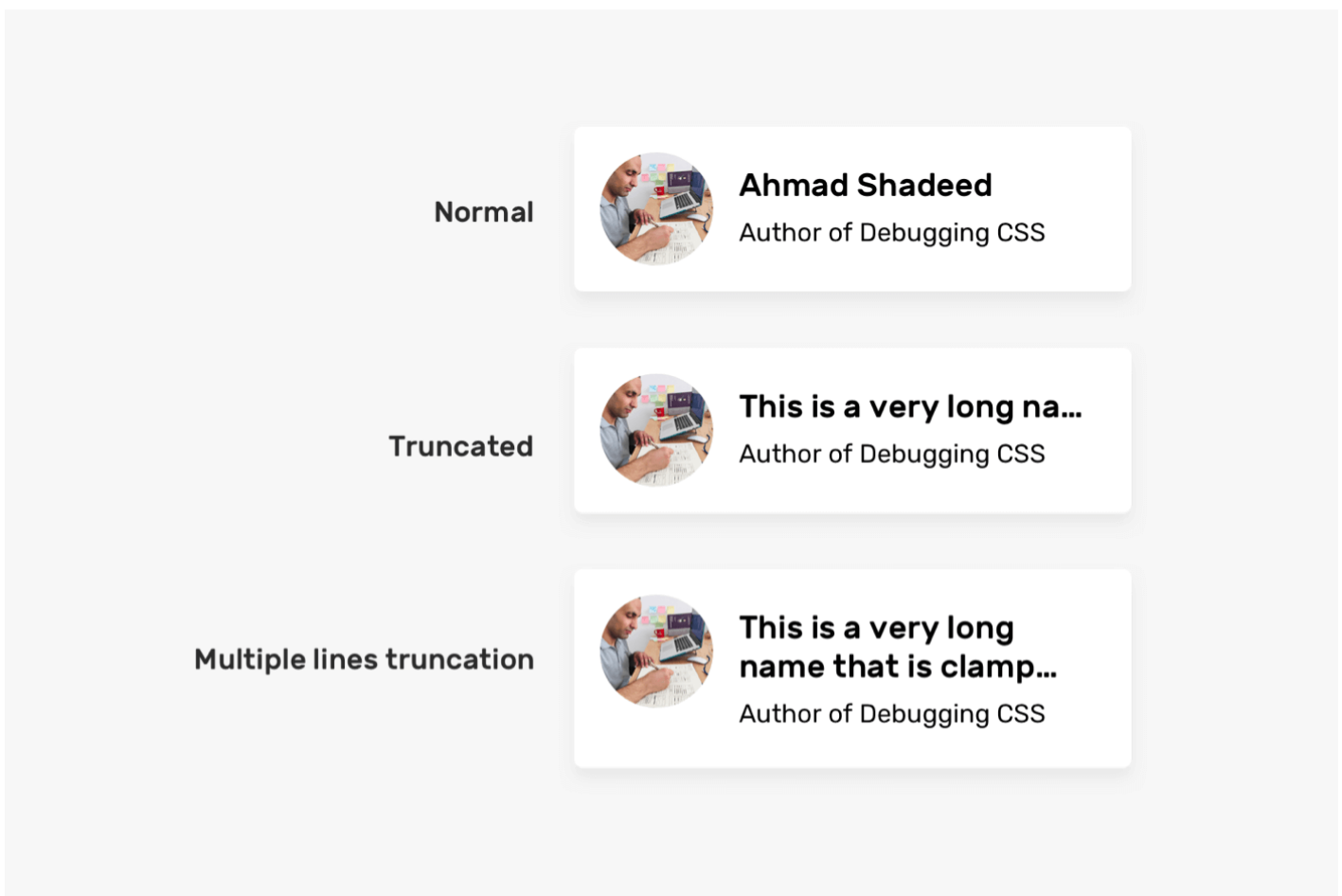
Результаты настройки минимальной ширины кнопки

Теперь, когда мы поговорили о проблемах вывода длинных и коротких текстов и о решениях этих проблем, давайте разберём несколько практических примеров.

Практические примеры

Карточка профиля пользователя

Карточки профиля пользователя часто содержат достаточно длинные тексты. В частности, проектируя подобную карточку, разработчику почти невозможно заранее узнать о том, насколько длинным будет имя пользователя. Как же быть?



Имя пользователя обычной длины, помещающееся на карточке целиком. Обрезанное длинное имя, выводимое в одной строке. Обрезанное длинное имя, выводимое в нескольких строках

Проектируя подобную карточку можно подготовиться к выводу в ней длинного имени пользователя, применив один из следующих двух приёмов:

```

/* Решение 1 */
.card__title {
  text-overflow: ellipsis;
  white-space: nowrap;
  overflow: hidden;
}

/* Решение 2 */
.card__title {
  display: -webkit-box;
  -webkit-line-clamp: 2;
  -webkit-box-orient: vertical;
  overflow: hidden;
}

```

Как видите, длинный текст можно обрезать, выведя его либо в одной строке, либо — в нескольких строках.

Навигационные элементы

При работе над макетами мультязычных сайтов нужно учитывать тот факт, что тексты, имеющие один и тот же смысл, но написанные на разных языках, могут иметь различную длину.

English - LTR - Left to right



Fast Delivery

Home

About

Services

Hire us

Track

Arabic - RTL - Right to left

تتبع

وظفنا

الخدمات

عنا

الرئيسية

فاست ديلفري



Названия навигационных элементов, выведенные на разных языках

Длина слова `About` из LTR-языка больше, чем длина аналогичного по смыслу слова из RTL-языка. При выводе на таком языке соответствующий пункт навигационного меню выглядит слишком коротким. Известно, что если в дизайне страниц используются маленькие области, с которыми нужно взаимодействовать пользователям, это плохо сказывается на UX. Как исправить проблему? В данном случае можно просто настроить минимальную ширину навигационного элемента:

```
.nav__item {  
  min-width: 50px;  
}
```

English - LTR - Left to right



Fast Delivery

Home

About

Services

Hire us

Track

Arabic - RTL - Right to left

تتبع

وظفنا

الخدمات

عنا

الرئيسية

فاست ديلفري

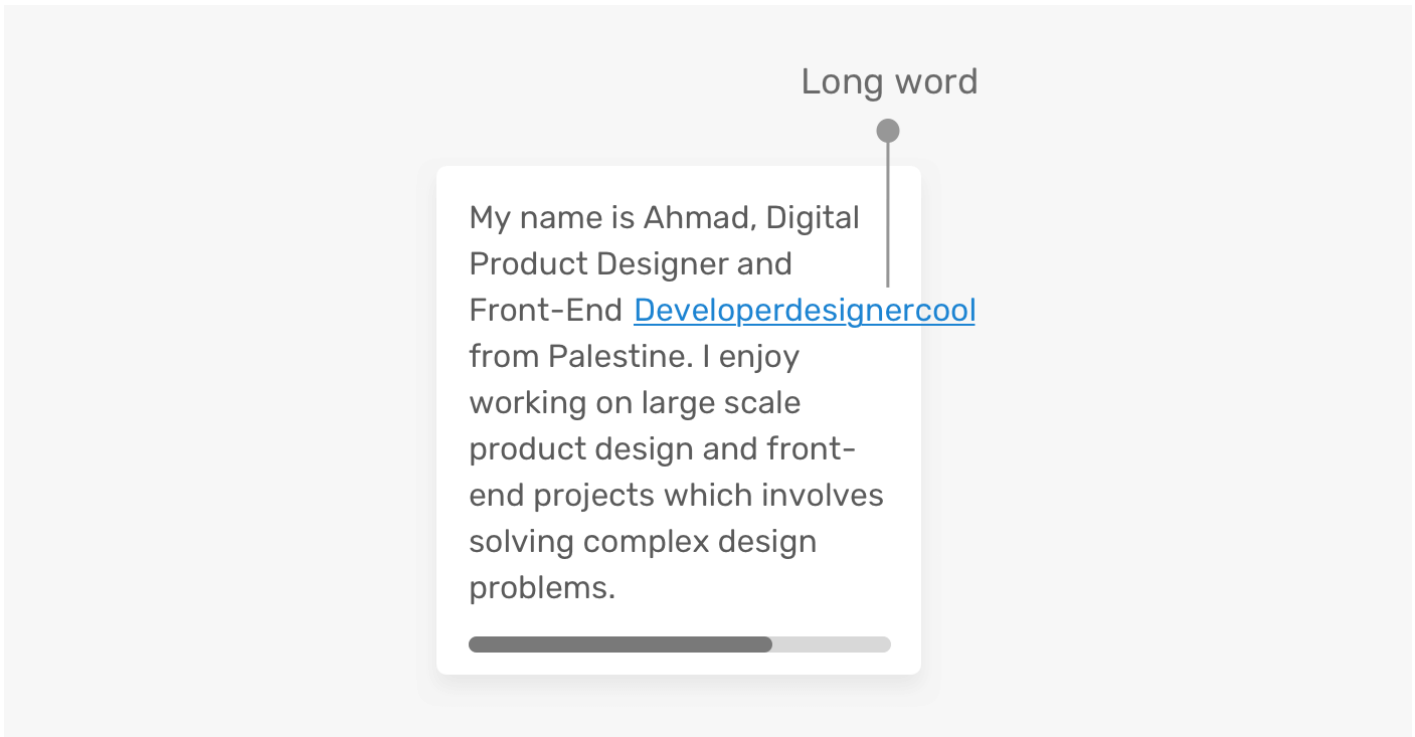


Решение проблемы короткого текста

Если вас интересуют вопросы вывода данных на разных языках — взгляните на этот мой материал.

Поле для вывода содержимого статей

В полях для вывода объёмных текстов вполне могут попадаться очень длинные слова, не помещающиеся в контейнеры. Особенно часто с этим можно столкнуться при работе с сайтами на мобильных устройствах.



Длинное слово выходит за пределы контейнера

Здесь имеется длинное слово, которое выходит за пределы контейнера и является причиной появления горизонтальной полосы прокрутки. Выше мы уже говорили о решениях подобных проблем, которые заключаются в использовании CSS-свойств `overflow-wrap` или `hyphens`.

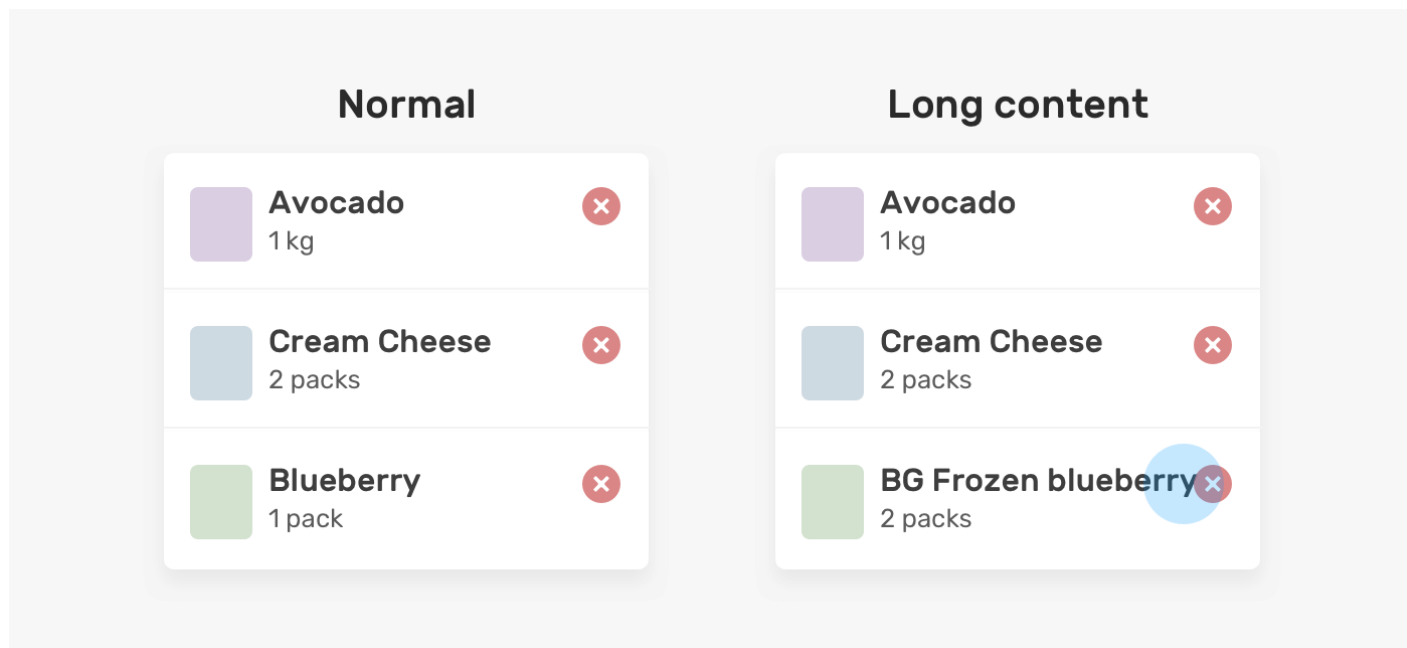
Например, эту проблему можно решить так:

```
.article-content p {
  overflow-wrap: break-word;
}
```

Оформление виртуальной корзины для покупок

Названия товаров, которые покупатели интернет-магазинов «кладут» в корзины, могут быть очень разными. Это может быть и одно слово, и несколько строк. В следующем примере длина названия товара такова, что текст перекрывается кнопкой для удаления товара из корзины. Причина этого в том, что при проектировании макета корзины не было

уделено достаточного внимания настройке расстояния между элементами.



Вывод коротких и длинных названий в макете, который настроен неправильно

Решить эту проблему можно, настроив внутренние или внешние отступы элементов. Конкретные действия зависят от ситуации. Здесь я приведу простой пример, предусматривающий использование свойства `margin-right` при настройке элемента, выводящего название товара.

```
.product__name {  
  margin-right: 1rem;  
}
```

Flexbox-макеты и вывод длинных текстов

При выводе длинных текстов во Flexbox-макетах возможна ситуация, когда такие тексты переполняют родительские элементы. Взгляните на следующий пример.

Ahmad Shadeed

Follow

Элементы выглядят нормально

Вот разметка к этому примеру:

```
<div class="user">
  <div class="user__meta">
    <h3 class="user__name">Ahmad Shadeed</h3>
  </div>
  <button class="btn">Follow</button>
</div>
```

Вот стили:

```
.user {
  display: flex;
  align-items: flex-start;
}

.user__name {
  text-overflow: ellipsis;
  white-space: nowrap;
  overflow: hidden;
}
```

Если имя пользователя не слишком длинно — всё выглядит нормально. Но что случится в том случае, если имя окажется достаточно длинным? В такой ситуации текст переполнит родительский элемент, а это «поломает» макет.

This is a long name here..... Follow

Длинное имя пользователя портит внешний вид страницы

Причина возникновения этой проблемы заключается в том, что размеры Flex-элементов не сокращаются до величин, которые меньше минимального размера их содержимого. Решить эту проблему можно, установив в значение `0` свойство `min-width` элемента `.user__meta`:

```
.user__meta {  
  /* другие стили */  
  min-width: 0;  
}
```

После этого даже вывод в элементе длинного имени пользователя не испортит макет. Некоторые подробности об использовании свойства `min-width` при разработке Flexbox-макетов вы можете найти в этом материале.

Итоги

Надеюсь, теперь вы сможете справиться с проектированием элементов, поддерживающих аккуратный вывод длинных и коротких текстов. Я с большим удовольствием писал эту статью. Я, благодаря этому, вспомнил некоторые тонкости работы с разными текстами. А статья будет служить мне хорошим напоминанием о том, как важно обращать внимание на подготовку элементов страниц к выводу текстов разной длины.

Версия #4

Seryak создал Fri, Jan 22, 2021 1:11 PM

Seryak обновил Fri, Jan 22, 2021 1:13 PM