

*Горячкин Б. С., кандидат технических наук, доцент; Московский государственный технический университет им. Н.Э. Баумана,*

*E-mail: [bsgor@mail.ru](mailto:bsgor@mail.ru)*

*Щукин А. С., магистрант, Московский государственный технический университет им. Н.Э. Баумана,*

*E-mail: [a.shch.2@yandex.com](mailto:a.shch.2@yandex.com)*

*Тодосиев Н. Д., магистрант, Московский государственный технический университет им. Н.Э. Баумана,*

*E-mail: [todosievnik@gmail.com](mailto:todosievnik@gmail.com)*

*Янковский В. И., магистрант, Московский государственный технический университет им. Н.Э. Баумана,*

*E-mail: [vlyankov@mail.ru](mailto:vlyankov@mail.ru)*

## **АНАЛИЗ СОВРЕМЕННЫХ ЕДИНИЦ ИЗМЕРЕНИЯ В ВЕБ-ДИЗАЙНЕ**

**Annotation:** This article is about units used in web design. The characteristic features of each unit of measure are analyzed, areas for their application are highlighted. The examples showed the ability to display certain units in browsers. The four most used browsers (Mozilla Firefox, Google Chrome, Internet Explorer, Safari, and Opera) were compared for rendering text and headers. This comparison showed the ability of browsers to adequately display a scalable font from small to large.

**Key words:** Web design, units of measure in web design, absolute units of measure, relative units of measure, usability, adaptability.

**Аннотация:** В данной статье речь идет об единицах измерения, используемых в веб-дизайне. Проанализированы характерные особенности каждой единицы измерения, выделены области для их применения. На

примерах были показаны возможности отображения тех или иных единиц измерения в браузерах. Было проведено сравнение четырех наиболее используемых браузеров (Mozilla Firefox, Google Chrome, Internet Explorer, Safari и Opera) на предмет рендеринга текста и заголовков. Это сравнение показало возможности браузеров адекватно отображать масштабируемый шрифт от мелкого до крупного.

**Ключевые слова:** Веб-дизайн, единицы измерения в веб-дизайне, абсолютные единицы измерения, относительные единицы измерения, юзабилити, адаптивность.

## **Введение**

Для того, чтобы предметно понимать, о чем пойдет речь в статье, нужно разобраться с вопросом «Что же такое Веб-дизайн?». Существует распространенное мнение о том, что веб-дизайн - это просто оболочка веб-страниц, которая только влияет на его общее восприятие. Но на самом деле все не совсем так. Прежде всего, под веб-дизайном стоит понимать работу над оптимальной структурой, наиболее удобным расположением контента и, конечно, интуитивно-понятной эргономикой. Поэтому, с нашей точки зрения веб-дизайн – это, во многом, логика подачи информации, исходя, во многом, из которой становится понятно насколько эффективен ваш сайт по большинству показателей.

В интернете можно найти множество определений слову «Веб-дизайн». На самом же деле, правда лежит где-то посередине и под этим словом следует понимать совокупность способов создания интерфейсов пользователей. То есть, задача веб-дизайнера состоит не только в том, чтобы создать эффектный внешний вид, но также продумать графическую составляющую на предмет цветового решения и композиции. Он должен еще и логично распределить отдельные элементы дизайна так, чтобы пользователю удобно было воспринимать информацию и изображения на странице и взаимодействовать с ней. Но, чтобы грамотно обращаться со всевозможными элементами веб-

страницы, нужно четко понимать, какой размер они должны занимать в тех или иных ситуациях (и как изменятся под действиями пользователей).

В данной статье мы рассмотрим существующие единицы измерения и проанализируем их, чтобы дать читателю представление о вариантах их использования.

### **Виды единиц измерения**

Существует два вида единиц измерения в веб-дизайне: относительные единицы измерения и абсолютные [1].

Относительные единицы названы так из-за того, что рассчитываются относительно экрана монитора пользователя, элемента внутри которого располагается текст или размеров активного окна.

Абсолютные единицы имеют постоянный размер, который никак не зависит от экранов пользователей, размеров элемента, внутри которых располагаются или размеров активных окон.

Относительные единицы чаще всего используются в тех местах, где необходимо масштабировать размер между объектами или при работе с текстами. В табл. 1 приведены основные относительные единицы.

Табл. 1. Относительные единицы измерения.

<b>Единица</b>	<b>Описание</b>
<u>em</u>	Высота шрифта текущего элемента
<u>ex</u>	Высота символа x
<u>px</u>	Пиксел
%	Процент
rem	<u>em</u> корневого элемента
<u>vw</u>	1/100 ширины окна пользователя
<u>vh</u>	1/100 высоты окна пользователя
<u>vmin</u>	Эквивалентно 1% меньшего размера окна браузера по высоте или ширине.
<u>vmax</u>	Эквивалентно 1% большего размера окна браузера по высоте или ширине.

В современных браузерах всегда предусмотрен стандартный размер шрифта, который применяется в тех случаях, когда размер не задан явно. Относительные единицы измерения обычно указываются явно. Они зависят от размера родительского элемента (см. свойство font-size), вследствие чего масштабируются без проблем.

Что же касается абсолютных единиц, то они используются реже, и, как правило, нужны при работе с текстом. В табл. 2 приведены основные такие единицы.

Табл. 2. Абсолютные единицы измерения.

<b>Единица</b>	<b>Описание</b>
<u>in</u>	Дюйм (1 дюйм равен 2,54 см)
<u>cm</u>	Сантиметр
<u>mm</u>	Миллиметр
<u>pt</u>	Пункт (1 пункт равен 1/72 дюйма)
<u>pc</u>	Пика (1 пика равна 12 пунктам)

### ***Относительные единицы измерения***

Единицы **em** и **ex** зависят от размера шрифта и их можно задать свои для каждого элемента в документе[2]. Единица **em** - размер шрифта. К примеру, давайте зададим в родительском элементе шрифт в 4pt, следом, при использовании другого свойства, например, **margin: 1em**, то мы получим отступы вокруг элемента равными 4pt. Что вообще нам дает указание размерности в em? В первую очередь гибкость в отображении. Если у пользователя достаточно крупный экран, то можно задействовать всю рабочую область экрана и отобразить шрифт крупнее. И наоборот, например, на мобильном экране, где экран сравнительно мелкий, можно отображать более мелкий шрифт. При это, все размеры останутся пропорциональными (Исходя из примера выше, объявление **margin: 1,5em** даст отступы в 6pt.).

*Ex* используется реже. Она равна X-высоте — это, грубо говоря, высота строчных букв, таких как: *a, c, m*, или *o*. Вполне реальна ситуация, когда у шрифтов с одинаковым размером строчные буквы имеют разную высоту. Поэтому, когда важно, чтобы какой-то элемент, был равен x-высоте, на помощь нам приходит **ex** единица.

Пиксел - это элементарная точка, отображаемая монитором или другим подобным устройством, например, смартфоном.

Почему пиксель относительная величина? В спецификации однозначно написано, что  $1\text{ px} = 3/4\text{ pt}$ . Но это работает только в типографских условиях. Давайте окунемся в реалии web-дизайна. Мы берем физический монитор, у которого есть диагональ — это реальная физическая величина. Но ведь мы, так же как и любой пользователь, можем поставить на нем разрешение любое, правильно? В зависимости от поставленного нами разрешения, величина пикселя может меняться. Вот почему пиксель — относительная величина. То есть на разных мониторах **px** имеет разный размер. Размер пикселя зависит от разрешения устройства и его технических характеристик.

Единица **px** в веб-дизайне является своеобразной величиной. Ее не связывают с выбранным шрифтом, также как и с физическими реальными единицами измерения (см, мм, м) обычно не связана.

Не стоит путать пиксели с CSS-пикселями (англ. CSS - Cascading Style Sheets — каскадные таблицы стилей) [3]. Мы можем задать свойство **width: 20px** и получить один CSS-пиксель, который будет равен двадцати реальным аппаратных. Более того, мы можем задать дробное свойство, например, **width: 20.5px**. Но, к сожалению, браузеры в таких случаях поступают как им угодно: Mozilla Firefox и Google Chrome будут округлять в большую сторону, отрисовывая 21 пиксель, когда Internet Explorer наоборот, отрисует 20px. Поэтому **px** по определению не фиксированная длина, а нечто, зависящее от типа устройства и его обычного использования.

С 2013 года в CSS ввели новую единицу измерения: **rem**. Один **rem** — это размер шрифта корневого элемента в документе. Тогда возникает логичный

вопрос в чем отличие `rem` от `em`? Все просто, **rem** всегда один на протяжении всего документа, когда `em` для каждого элемента может задаваться персонально [4].

Существует в веб-дизайне такое понятие, как «отзывчивый веб-дизайн». Его еще называют «адаптивным». Суть этого подхода в том, чтобы страничка в интернете максимально красиво и эффективно отображалась на любых устройствах, будь то мобильный телефон, планшет или экран монитора ПК. Для достижения такого результата в большинстве своем используют процентные значения. Но это далеко не лучшая практика для всех случаев. Связано это с тем, что процентные значения вычисляются относительно соседних элементов, что не всегда одинаково полезно. Здесь нам на помощь приходят `vh` и `vw`. Они вычисляются как одна сотая от ширины и высоты окна браузера соответственно.

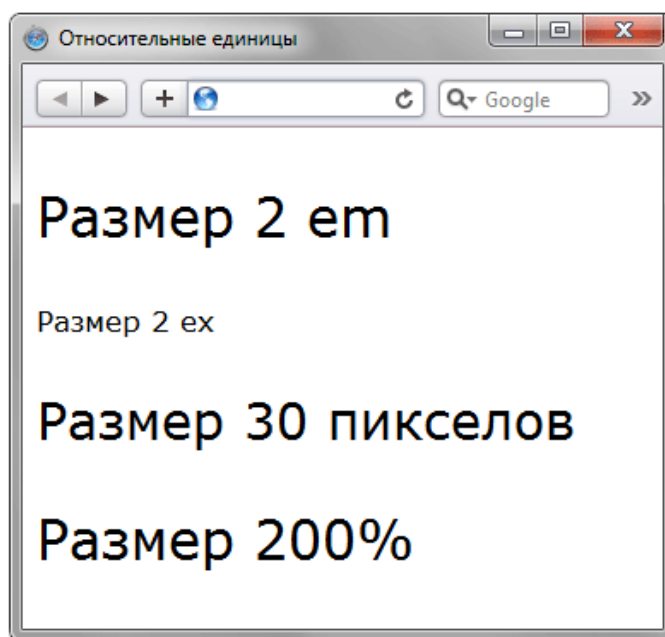


Рис. 1. Результат применения разных относительных единиц измерения.

## *Абсолютные единицы измерения*

Соотношение между абсолютными единицами таково:

Табл. 3. Соотношение абсолютных единиц измерения.

Пункт, <u>pt</u>	<u>Пиксель</u> , <u>px</u>	Единица, <u>em</u>	Процент, %
6pt	8px	0.5em	50%
7pt	9px	0.55em	55%
7.5pt	10px	0.625em	62.5%
8pt	11px	0.7em	70%
9pt	12px	0.75em	75%
10pt	13px	0.8em	80%
10.5pt	14px	0.875em	87.5%
11pt	15px	0.95em	95%
12pt	16px	1em	100%

В общем случае:

$$1 \text{ in} = 2.54 \text{ cm} = 25.4 \text{ mm} = 72 \text{ pt} = 6 \text{ pc} \quad (1)$$

Так называемые абсолютные единицы (**cm**, **mm**, **in**, **pt** и **pc**) означают то же самое, что и везде, но только если у устройства вывода достаточно высокое разрешение. На лазерном принтере 1cm должен быть точно равен 1 сантиметру. Но на устройствах низкого разрешения, вроде компьютерных экранов, CSS этого не требует. И вправду, разные устройства и разные реализации CSS норовят отобразить их по-разному [5]. Лучше оставить эти единицы для устройств высокого разрешения, в частности для печати. На компьютерных экранах и мобильных устройствах может получиться не то, что ожидалось.

Самой, пожалуй, распространенной единицей является пункт, который используется для указания размера шрифта. Многие люди привыкли задавать размер шрифта в текстовых редакторах, например, 12. А что это число означает, не понимают [6]. Так это и есть пункты. Мы привыкли измерять все в

миллиметрах и подобных единицах, но пункт, пожалуй, единственная величина не из метрической системы измерения, которая используется у нас повсеместно. И все благодаря текстовым редакторам и издательским системам.

Типографский пункт - единица измерения кегля шрифта. Один пункт равен 1/12 высоты шрифта цисеро.

Также пункт равен 1/12 pc и 1/48 квадрата. Во входящих в Таможенный союз странах СНГ пункт исчисляется по типометрической системе Дидо, в которой 1 пункт равен 0,376 мм [7]. На рис. 2 показан пример использования четырех относительных единиц измерения.

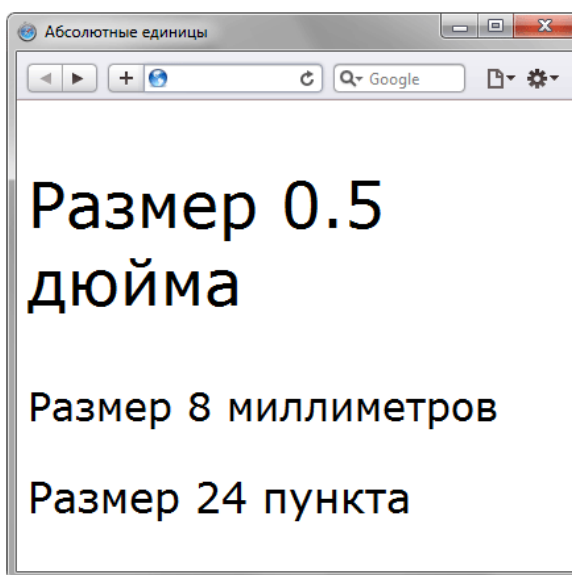


Рис. 2. Результат применения разных абсолютных единиц измерения.

## **Сравнительный анализ браузеров на предмет рендеринга текста**

Перед проведением теста стоит отметить, что затрагивать мы будем только шесть самых распространенных на сегодняшний момент браузеров: Mozilla Firefox, Google Chrome, Internet Explorer, Safari и Opera.

Сам тест был проведен сайтом [www.graphicrating.com](http://www.graphicrating.com) и состоял из нескольких частей, речь о которых пойдет далее.

В качестве первого теста используем шесть заголовков (h1-h6) и отметим изменения. Конечно, пользователь может изменить стандарты шрифтов в



браузере и тогда заголовки не будут соответствовать стандартным размерам. Кроме того, эти значения могут варьироваться из-за разницы операционных систем. Однако, в общем случае получаем:

- h1 – 32 пикселя.
- h2 – 24 пикселя.
- h3 – 19 пикселей.
- h4 – 16 пикселей.
- h5 – 13 пикселей.
- h6 – 11 пикселей.

На рис. 3 показан результат этого теста.

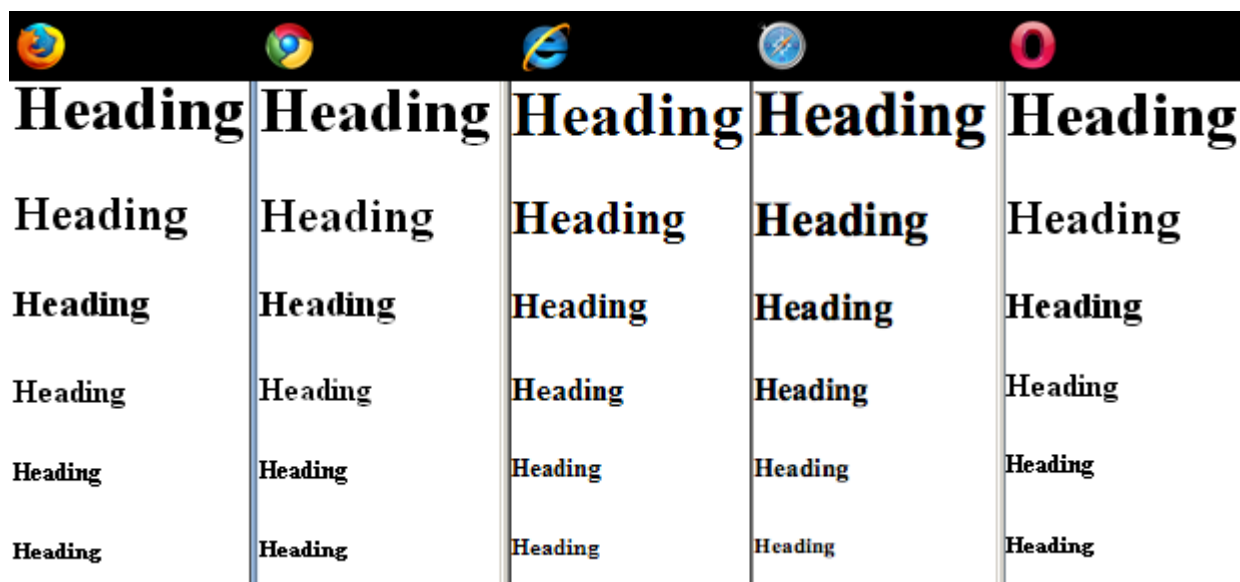


Рис. 3. Результат теста рендеринга заголовков h1-h6 разными браузерами.

При рассмотрении H1 вы можете ясно видеть, что Safari использует слишком сильное размытие краев. Результат Internet Explorer, наоборот, выглядит не сглаженным. Chrome, Firefox и Opera имеют лучший результат. По сравнению с другими браузерами Internet Explorer предлагает более читаемый текст. Такого же результата Internet Explorer достиг и в остальных заголовках, где Opera, Firefox и Chrome показывают одинаковые результаты, а у Safari текст слишком размыт.

## Сравнительный анализ браузеров на предмет отображения различных шрифтов

Этот тест состоит из трех блоков текста с разными шрифтами и является логическим продолжением теста их прошлого раздела, проведенный тем же сайтом с теми же браузерами.

Мы использовали популярные шрифты Arial 13px, Verdana 13px and Georgia 15px. На рис. 4 показан результат этого теста.

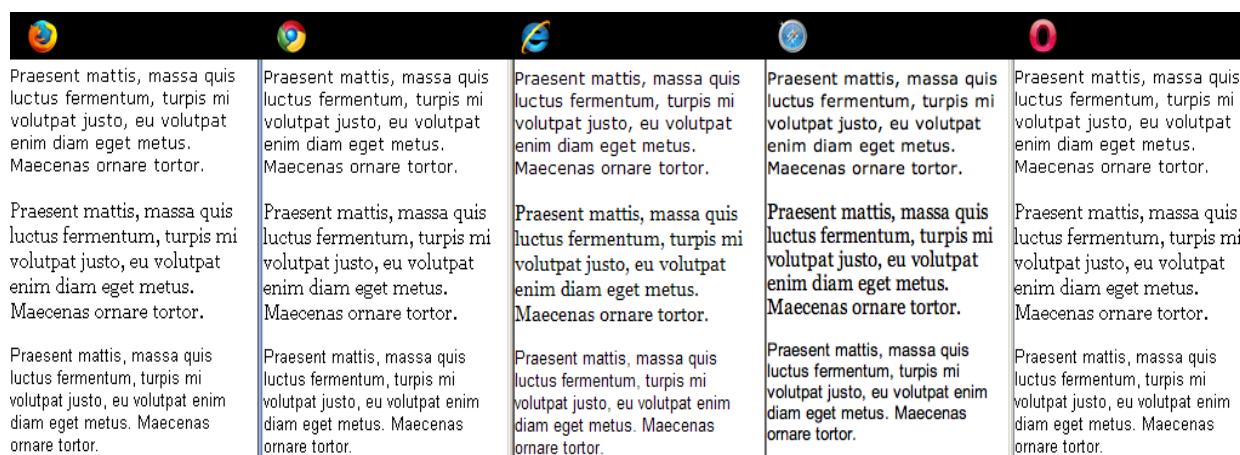


Рис. 4. Результат теста рендеринга шрифтов разными браузерами.

В этом тесте очень просто определить победителя — это Internet Explorer. Его движок рендеринга, наиболее сбалансированный в отображении разнообразных шрифтов. Даже для низких значений, вроде 13px, IE предлагает приятный эффект размытия. Вопреки всеобщему мнению, Safari показал посредственные результаты в рендеринге текста. Его движок отображает дополнительный вес в шрифтах, делая их тяжелее, ухудшая читаемость. Opera, Firefox и Chrome имеют схожее друг с другом поведение при отображении текста и делят почетное второе место

### Что использовать?

Размеры шрифтов еще лучше задавать в **em**. Идея в том, чтобы, во-первых, не задавать размер шрифта для элемента BODY (в HTML), а использовать размер шрифта по умолчанию для устройства, поскольку это

наиболее удобный для читателя размер; и, во-вторых, указывать размеры шрифта других элементов в **em**, например: **H1 {font-size: 2.5em}**. Таким образом, H1 стал в 2½ раза крупнее основного шрифта страницы.

Легко понять разницу между единицами размера шрифта, когда вы видите их в действии. Обычно 1em = 12pt = 16px = 100%. Используя эти размеры шрифтов, давайте посмотрим, что произойдет, когда вы увеличите базовый размер шрифта (используя CSS-селектор body) со 100% до 120%.

	<code>body { font-size: 100%; }</code>	<code>body { font-size: 120%; }</code>
<code>font-size: 1em</code>	The quick brown fox	The quick brown
<code>font-size: 12pt</code>	The quick brown fox	The quick brown fox
<code>font-size: 16px</code>	The quick brown fox	The quick brown fox
<code>font-size: 100%</code>	The quick brown fox	The quick brown

Рис. 5. Пример, демонстрирующий разницу единиц измерения

Как видим на рис.5, em и процентные единицы увеличиваются при увеличении базового размера шрифта, а пиксели и точки - нет. Может быть легко установить абсолютный размер для вашего текста, но вашим посетителям гораздо проще использовать масштабируемый текст, который может отображаться на любом устройстве или любом компьютере. По этой причине em и процентные единицы предпочтительнее для текста веб-документа.

Единственное место, где лучше всего использовать **pt** (либо **cm** или **in**) для размера шрифтов — стили для печати, если нужно, чтобы напечатанный шрифт был строго определенного размера. Но даже там чаще всего лучше использовать размер шрифта по умолчанию.

Еще одна причина не использовать абсолютные единицы где-либо, кроме печати: на разные экраны мы смотрим с разного расстояния. 1 сантиметр на экране настольного компьютера выглядит маленьким. Но на мобильном экране прямо перед глазами — это много. Именно поэтому лучше вместо них использовать относительные единицы, например **em**.

Таким образом, единицы px и pt, не лучшим образом подходят для веб-документов, заставляет нас использовать em и %. В теории, единицы em и % являются идентичными, но на практике они имеют незначительные различия, которые важно учитывать. В приведенном выше примере мы использовали в качестве базовой единицы font-size проценты (в тэге body). Если вы измените вашу базовую единицу font-size с % на em (то есть body {font-size: 1em;}), вы, должны бы не заметить разницы. Давайте посмотрим, что происходит, когда 1em является нашей базовой единицей, и когда клиент меняет «Размер шрифта» в настройках своего браузера (такая возможность предусмотрена во многих браузерах, например, Internet Explorer). На рис. 6 показан пример такого использования базовой единицы.

<code>body = 1em</code>	Text-Size: "Smallest"	Text-Size: "Largest"
<code>font-size: 1em</code>	The quick brown fox jumps over the lazy dog.	The quick
<code>font-size: 100%</code>	The quick brown fox jumps over the lazy dog.	The quick br

Рис. 6. Разница 1em и 100% при body {font-size: 1em;}

Когда в браузере клиента размер текста установлен в «средней», то незаметно никакой разницы между em и %. Однако, если параметр изменять как на рис.8, разница становится очень большой. При установке «Smallest» em гораздо меньше, чем %, а при установке «Largest» наоборот em отображается гораздо большим, чем %. И хотя многие утверждают, что единицы в em масштабируются так, как задумано, на практике текст в em масштабируется слишком резко, и при этом наименьший текст становится неразборчивым на некоторых машинах.

### Заключение

В настоящее время, когда параллельно существуют множество браузеров, устройств со своими разрешениями и, конечно же, пользователей со своими предпочтениями лучше всего себя показывают относительные единицы

измерения em и %. Они «умеют подстраиваться» под конкретное устройство и замечательно масштабируются, что дает пользователю возможность одинаково комфортно читать как мелкий текст в интернете увеличив его, так и уменьшить заведомо большой текст. Равно, как и длина блока при верстке, выраженная в относительных единицах, прекрасно впишется в любой макет. Каждая из названных единиц измерения находит свое применение. Хотя следует отметить, что в последнее время все больше разработчиков используют в своих проектах при работе с текстами %.

Создавая сайт в интернете нельзя забывать и о том, что разные браузеры по-разному рендерят одинаковые шрифты. Из-за этого следует адаптировать свой код под каждый браузер отдельно с тем, чтобы добиться максимального качества и лучшего соответствия макету заказчика. В этом помогает кросс-браузерная верстка, но это тема для отдельной статьи и в рамках этой освещаться не будет.

#### **Библиографический список:**

1. htmlbook [Электронный ресурс] – Режим доступа: <http://htmlbook.ru/content/edinitsy-izmereniya.html> – Заглавие с экрана. – (Дата обращения: 10.10.2019).
2. W3C [Электронный ресурс] – Режим доступа: <https://www.w3.org/Style/Examples/007/units.ru.html> – Заглавие с экрана. – (Дата обращения: 10.10.2019).
3. Antula Профессиональная студия веб-дизайна [Электронный ресурс] – Режим доступа: <http://www.antula.ru/fonts-3.htm> – Заглавие с экрана. – (Дата обращения: 10.10.2019).
4. Javascript [Электронный ресурс] – Режим доступа: <https://learn.javascript.ru/css-units> – Заглавие с экрана. – (Дата обращения: 12.10.2019).

5. surgebook [Электронный ресурс] – Режим доступа: <https://www.surgebook.com/daniilko0/book/osnovy-web-dizayna-frontend1111/7-css-edinicy-izmereniya> – Заглавие с экрана. – (Дата обращения: 12.10.2019).

6. Кирсанов Дмитрий. «Веб-дизайн» [Текст] / Д. Кирсанов. – Символ - Плюс, 2006. – 231 с.

7. Королькова Александра. «Живая типографика» [Текст] / А. Королькова. – ИндексМаркет, 2012. – 224 с.