**Так что же такое CSS?**

Как и HTML, CSS на самом деле не является языком программирования. Это не язык разметки - это *язык таблицы стилей.*Это означает, что он позволяет применять стили выборочно к элементам в документах HTML. Например, чтобы выбрать **все** элементы абзаца на HTML странице и изменить текст внутри них с черного на красный, вы должны написать этот CSS:

p {

color: red;

}

*Пример для выполнения:*

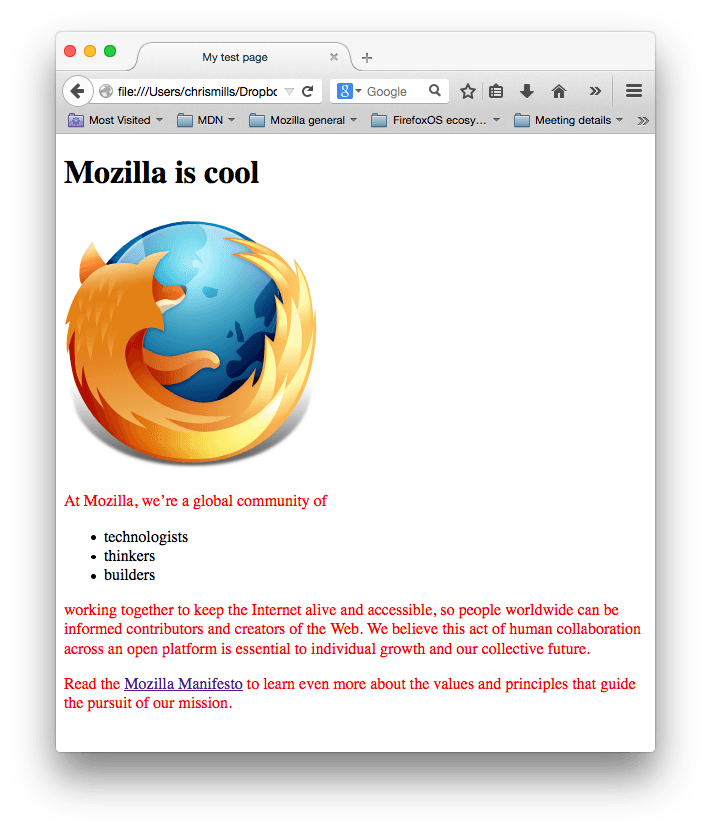
Давайте попробуем: вставьте эти три строки CSS в новый файл в ваш текстовый редактор, а затем сохраните файл как style.css в вашей папке styles.

Но нам все равно нужно применить CSS к нашему HTML документу. В противном случае, CSS стиль не повлияет на то, как ваш браузер отобразит HTML документ. (Если вы не следили за нашим проектом, то прочитайте раздел [Работа с файлами](https://developer.mozilla.org/ru/Learn/Getting_started_with_the_web/Dealing_with_files) и [Основы HTML](https://developer.mozilla.org/ru/Learn/Getting_started_with_the_web/HTML_basics), чтобы узнать, что вам нужно сделать в первую очередь.)

1. Откройте ваш файл index.html и вставьте следующую строку куда-нибудь в шапку, между <head> и </head> тегами:

<link href="styles/style.css" rel="stylesheet" type="text/css">

1. Сохраните index.html и загрузите его в вашем браузере. Вы должны увидеть что-то вроде этого:



Если текст вашего абзаца теперь красный, примите наши поздравления! Вы написали свой первый успешный CSS!

Анатомия набора CSS правил



Вся структура называется **набором правил** (но зачастую для краткости "правило" ). Отметим также имена отдельных частей:

**Селектор (Selector)**

Имя HTML элемента в начале набора правил. Он выбирает элемент(ы) для применения стиля (в данном случае, элементы p ). Для стилизации другого элемента, просто измените селектор.

**Объявление (Declaration)**

Единственное правило, например color: red; указывает, какие из **свойств** элемента вы хотите стилизовать.

**Свойства (Properties)**

Способы, которыми вы можете стилизовать определенный HTML элемент (в данном случае, color является свойством для элементов [<p>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/p)). В CSS вы выбираете, какие свойства вы хотите затронуть в вашем правиле.

**Значение свойства (Property value)**

Справа от свойства, после двоеточия, у нас есть **значение свойства**, который выбирает одно из множества возможных значений для данного свойства (существует множество значений color, помимо red).

Обратите внимание на важные части синтаксиса:

* Каждый набор правил (кроме селектора) должен быть обернут в фигурные скобки ({}).
* В каждом объявлении необходимо использовать двоеточие (:), чтобы отделить свойство от его значений.
* В каждом наборе правил вы должны использовать точку с запятой (;), чтобы отделить каждое объявление от следующего.

Таким образом, чтобы изменить несколько значений свойств сразу, вам просто нужно написать их, разделяя точкой с запятой, например так:

p {

color: red;

width: 500px;

border: 1px solid black;

}

Выбор нескольких элементов

Вы также можете выбрать несколько элементов разного типа и применить единый набор правил для всех из них. Добавьте несколько селекторов, разделенных запятыми. Например:

p,li,h1 {

color: red;

}

Разные типы селекторов

Существует множество различных типов селектора. Выше мы рассматривали только **селектор элементов**, который выбирает все элементы определенного типа в HTML документе. Но мы можем сделать выбор более конкретным. Вот некоторые из наиболее распространенных типов селекторов:

| **Имя селектора** | **Что выбирает** | **Пример** |
| --- | --- | --- |
| Селектор элемента (иногда называемый селектором тега или типа) | Все HTML элемент(ы) указанного типа. | p Выбирает <p> |
| ID селектор | Элемент на странице с указанным ID на данной HTML. Лучше всего использовать один элемент для каждого ID (и конечно один ID для каждого элемента), даже если вам разрешено использовать один и тот же ID для нескольких элементов. | #my-id Выбирает <p id="my-id"> или <a id="my-id"> |
| Селектор класса | Элемент(ы) на странице с указанным классом (множество экземпляров класса может объявляться на странице). | .my-class Выбирает <p class="my-class"> и <a class="my-class"> |
| Селектор атрибута | Элемент(ы) на странице с указанным атрибутом. | img[src] Выбирает <img src="myimage.png"> но не <img> |
| Селектор псевдо-класса | Указанные элемент(ы), но только в случае определенного состояния, например, при наведении курсора. | a:hover Выбирает <a>, но только тогда, когда указатель мыши наведен на ссылку. |

Существует еще много селекторов для изучения.

**Шрифты и текст**

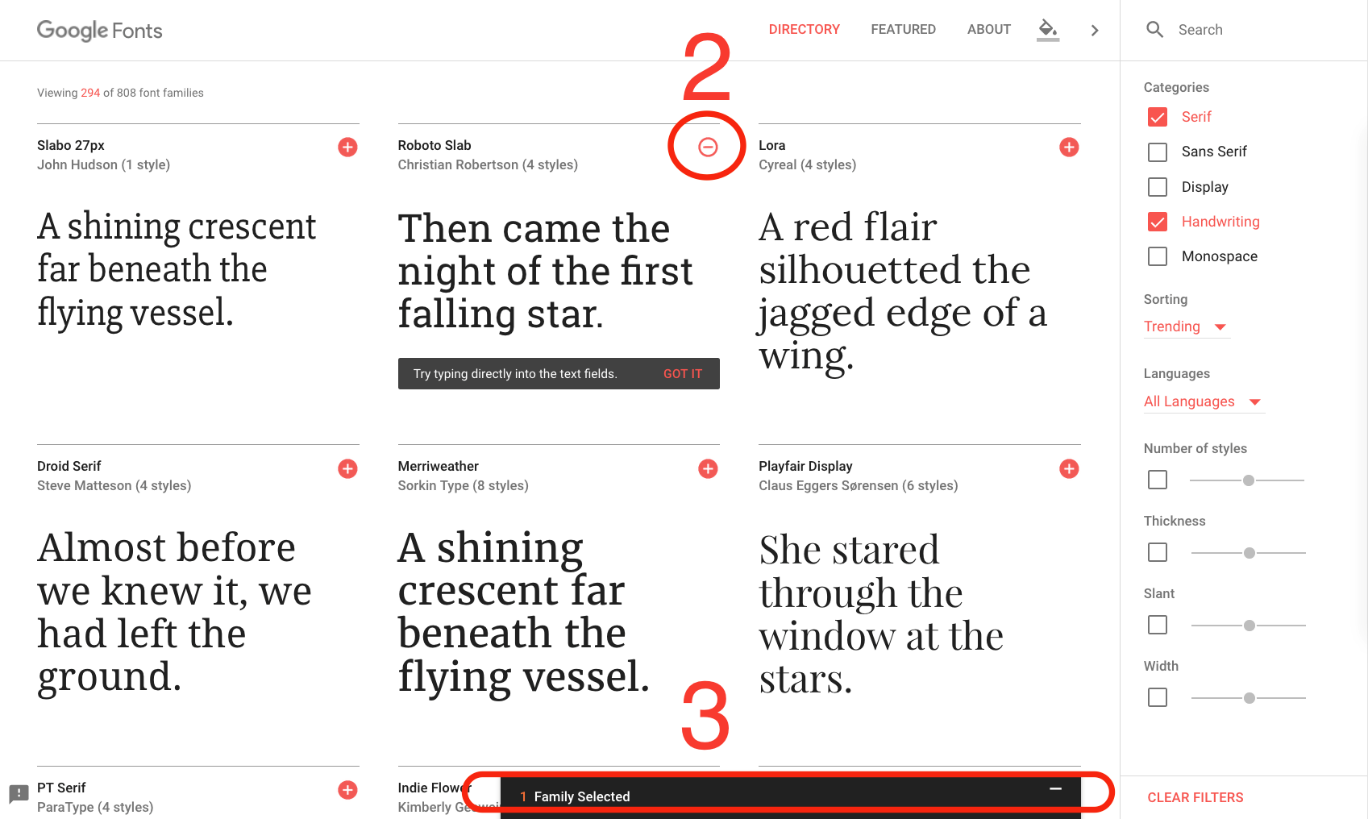
Теперь, когда мы изучили некоторые основы CSS, давайте добавим ещё несколько правил и информацию в наш файл style.css, чтобы наш пример хорошо выглядел. Прежде всего, давайте сделаем, чтобы наши шрифты и текст выглядели немного лучше.

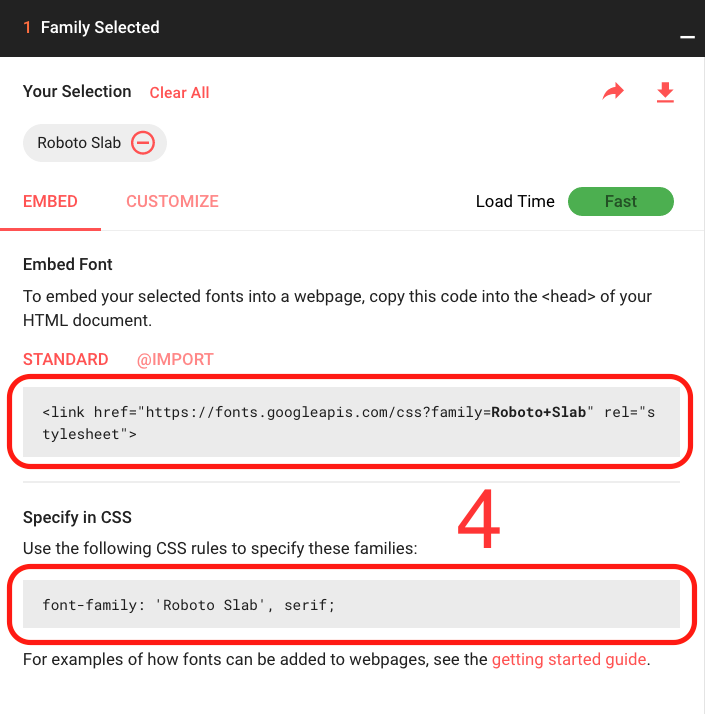
*Пример для выполнения:*

1. Прежде всего, выполните следующее:

Чтобы выбрать шрифт:

1. *Перейдите на [Google Fonts](http://www.google.com/fonts) и прокрутите список вниз, пока не найдете шрифт, который вам понравится. Вы также можете использовать элементы управления справа для дальнейшей фильтрации результатов.*
2. *Щёлкните по кнопке "плюс" рядом со шрифтом, который вы хотите выбрать.*
3. *Щёлкните по кнопке "\* Family Selected" на панели в нижней части страницы. ("\*" зависит от того, сколько шрифтов вы выбрали)*
4. *В всплывающем окне вы можете увидеть и скопировать строки кода, которые предоставляет Google, чтобы сохранить их позже в вашем текстовом редакторе.*

**



Добавьте элемент [<link>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/link) где-нибудь внутри шапки вашего index.html (снова, в любом месте между тегами <head> и </head>). Это будет выглядеть примерно так:

<link href='http://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css'>

Этот код связывает вашу страницу с таблицой стилями, которая загружает семейство шрифтов Open Sans вместе с вашей страницой и позволяет вам применять их к вашим HTML элементам используя свою собственную таблицу стилей.

1. Затем, удалите существующее правило в вашем style.css файле. Это был хороший тест, но красный текст, на самом деле, не очень хорошо выглядит.
2. Добавьте следующие строки в нужное место, заменив строку placeholder актуальной font-family строкой, которую вы получили из Google Fonts. (font-family просто означает, какой шрифт(ы) вы хотите использовать для вашего текста). Это правило устанавливает глобальный базовый шрифт и размер шрифта для всей страницы (поскольку <html> является родительским элементом для всей страницы, и все элементы внутри него наследуют такой же font-size и font-family):
3. html {
4. font-size: 10px; /\* px значит 'пиксели': базовый шрифт будет 10 пикселей в высоту \*/
5. font-family: placeholder: здесь должно быть имя шрифта из Google fonts

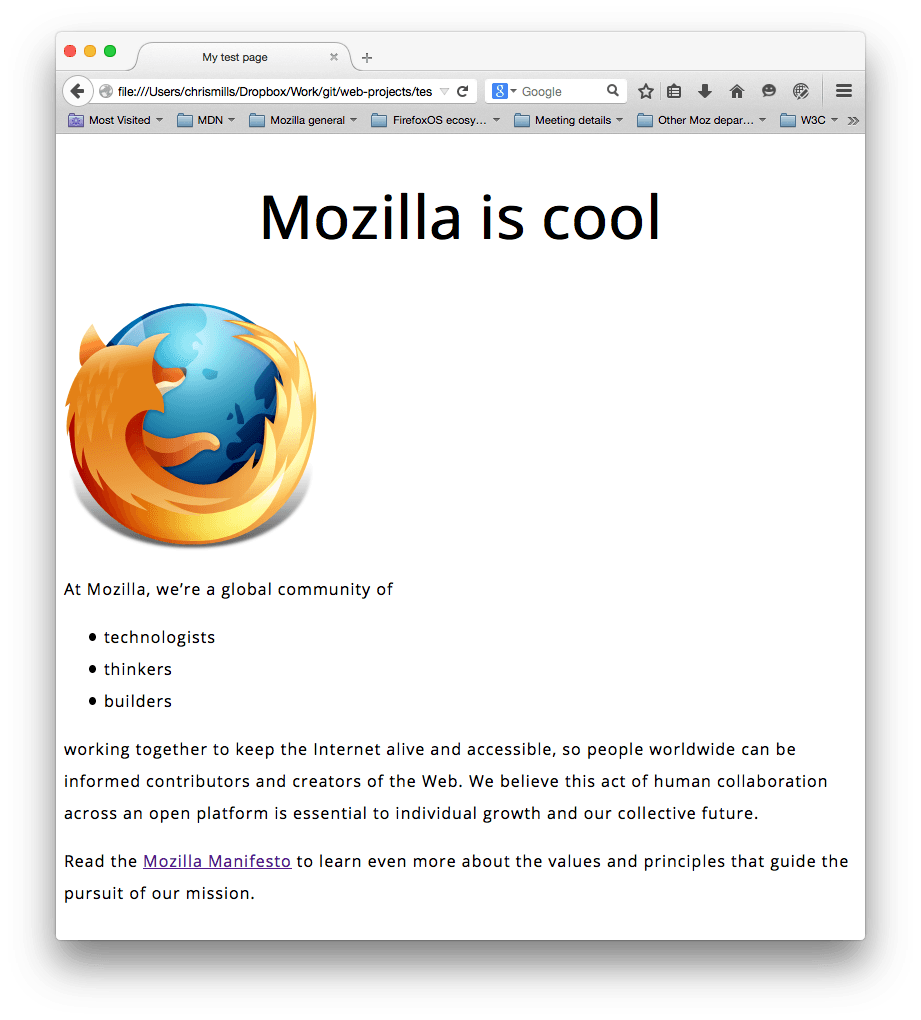
}

**Примечание**: Все в CSS документе между /\* и \*/ является **CSS комментарием**, который браузер игнорирует при исполнении кода. Это место, где вы можете написать полезные заметки о том, что вы делаете.

1. Теперь мы установим размер шрифта для элементов, содержащих текст внутри HTML тела ([<h1>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/h1), [<li>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/li), и [<p>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/p)). Мы также отцентрируем текст нашего заголовка и установим некоторую высоту строки и расстояние между буквами в теле документа, чтобы сделать его немного более удобным для чтения:
2. h1 {
3. font-size: 60px;
4. text-align: center;
5. }
6. p, li {
7. font-size: 16px;
8. line-height: 2;
9. letter-spacing: 1px;

}

Вы можете настроить значения px так, как вам нравится, чтобы ваш дизайн выглядел так, как вы хотите, но, в общем, ваш дизайн может выглядеть вот так:



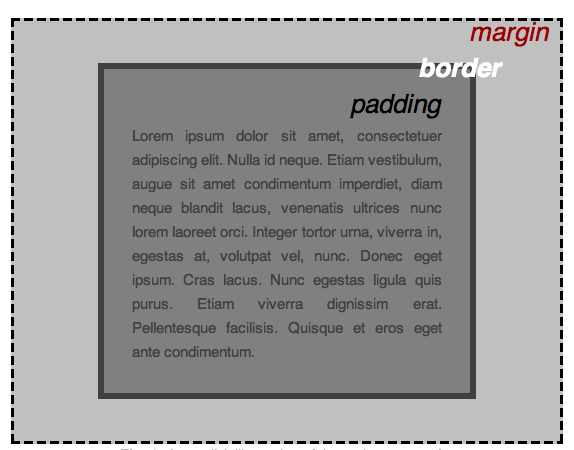
**Блоки**

Одна вещь, которую вы заметите в написании CSS, заключается в том, что многое из этого касается блоков - настройка их размера, цвета, положения и т.д. Большинство HTML элементов на странице можно рассматривать как блоки, расположенные друг над другом.



Не удивительно, макет CSS основан, главным образом, на *блочной модели (box model)*. Каждый из блоков, занимающий пространство на вашей странице имеет такие свойства, как:

* padding, пространство только вокруг контента (например, вокруг абзаца текста)
* border, сплошная линия, которая расположена рядом с padding
* margin, пространство вокруг внешней стороны элемента



В этом разделе мы также используем:

* width (ширину элемента)
* background-color, цвет позади контента и padding элементов
* color, цвет контента элемента (обычно текста)
* text-shadow: устанавливает тень на тексте внутри элемента
* display: устанавливает режим отображения элемента (пока что не волнуйтесь об этом)

Итак, давайте начнем и добавим больше CSS на нашей странице! Продолжайте добавлять эти новые правила, расположенные в нижней части страницы, и не бойтесь экспериментировать с изменением значений, чтобы увидеть, как это работает.

Изменение цвета страницы

html {

background-color: #00539F;

}

Это правило устанавливает цвет фона для всей страницы. Измените код цвета сверху, на цвет который [вы выбрали при планировании вашего сайта](https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/What_will_your_website_look_like).

Разбираемся с телом

body {

width: 600px;

margin: 0 auto;

background-color: #FF9500;

padding: 0 20px 20px 20px;

border: 5px solid black;

}

Теперь для [<body>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/body) элемента. Здесь есть немало деклараций, так что давайте пройдем через них всех по одному:

* width: 600px; — заставляет тело быть всегда 600 пикселей в ширину.
* margin: 0 auto; — когда вы устанавливаете два значения для таких свойств как margin или padding, первое значение элемента влияет на верхнюю **и** нижнюю сторону (делает их 0 в данном случае), и второе значение на левую **и** правую сторону (здесь, auto является особым значением, которое делит доступное пространство по горизонтали поровну слева и справа). Вы также можете использовать один, три или четыре значения, как описано [здесь](https://developer.mozilla.org/ru/docs/Web/CSS/margin#Values).
* background-color: #FF9500; — как и прежде, устанавливает цвет фона элемента. Я использовал красновато-оранжевый для тела, в отличие от темно-синего цвета для [<html>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/html) элемента, но не стесняйтесь и эксперементируйте.
* padding: 0 20px 20px 20px; — у нас есть четыре значения, установленные для padding, чтобы сделать немного пространства вокруг нашего контента. В этот раз мы не устанавливаем padding на верхней части тела, но делаем 20 пикселей слева, снизу и справа. Значения устанавливаются сверху, справа, снизу, слева, в таком порядке.
* border: 5px solid black; — просто устанавливает сплошную черную рамку шириной 5 пикселей со всех сторон тела.

Позиционирование и стилизация нашего заголовка главной страницы

h1 {

margin: 0;

padding: 20px 0;

color: #00539F;

text-shadow: 3px 3px 1px black;

}

Вы, возможно, заметили, что есть ужасный разрыв в верхней части тела. Это происходит, потому что браузеры применяют некоторый **стиль по умолчанию** для элемента [<h1>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/h1) (по сравнению с другими), даже если вы не применяли какой-либо CSS вообще! Это может звучать как плохая идея, но мы хотим, чтобы веб-страница без стилей имела базовую читаемость. Чтобы избавиться от разрыва, мы переопределили стиль по умолчанию, установив margin: 0;.

Затем мы установили заголовку верхний и нижний padding на 20 пикселей, и сделали текст заголовка того же цвета, как и цвет фона html.

Здесь, мы использовали одно довольно интересное свойство - это text-shadow, которое применяет тень к текстовому контенту элемента. Оно имеет следующие четыре значения:

* Первое значение пикселей задает **горизонтальное смещение** тени от текста — как далеко она движется поперек: отрицательное значение должно двигать ее влево.
* Второе значение пикселей задает **вертикальное смещение** тени от текста — как далеко она движется вниз, в этом примере: отрицательное значение должно переместить ее вверх.
* Третье значение пикселей задает **радиус размытия** тени — большее значение будет означать более размытую тень.
* Четвертое значение задает основной цвет тени.

И вновь попробуйте поэкспериментировать с различными значениями, чтобы посмотреть, что вы можете придумать.

Центрирование изображения

img {

display: block;

margin: 0 auto;

}

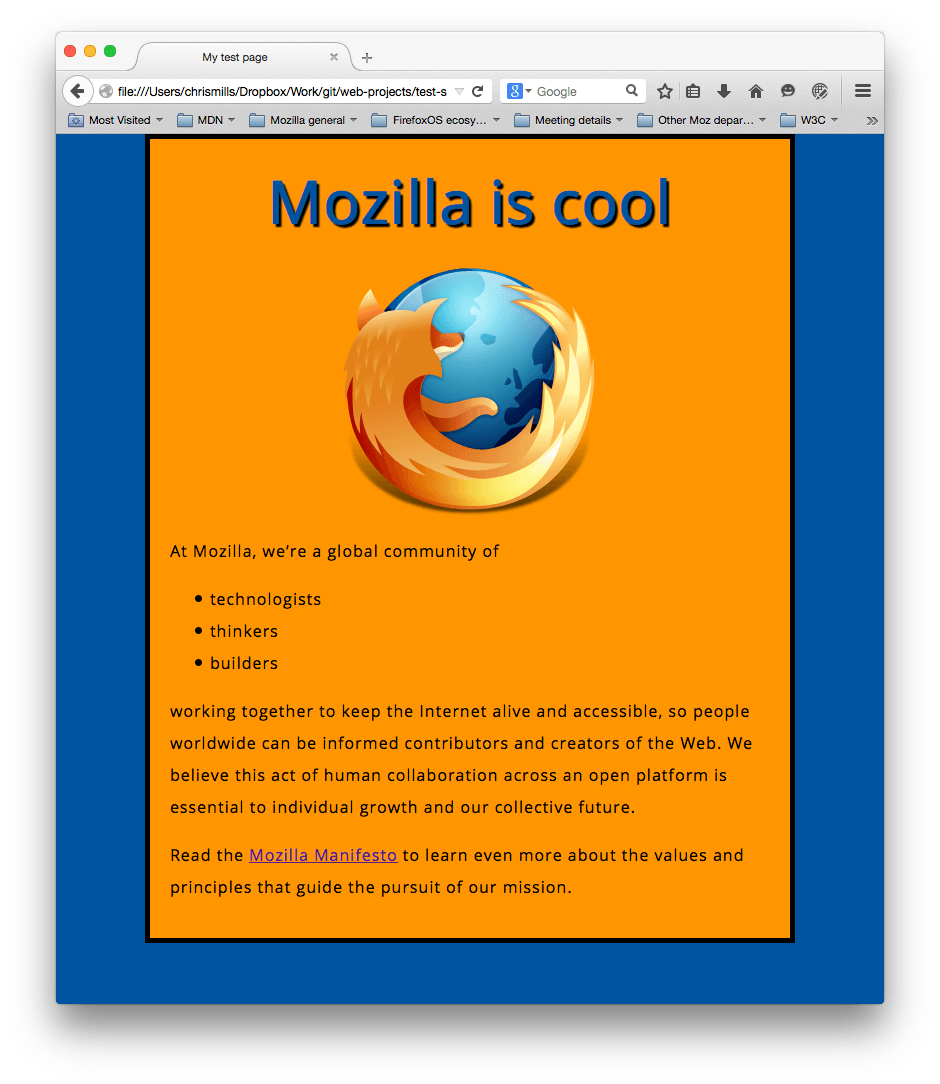
В заключение, мы отцентрируем изображение, чтобы оно лучше выглядело. Мы можем использовать margin: 0 auto уловку снова, как мы это делали раньше для body, но мы также должны сделать кое-что еще. Элемент [<body>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/body) является **блочным**, это значит, что он занимает место на странице и может иметь margin и другие значения отступов, применяемых к нему. Изображения, наоборот, являются **строчными** элементами, то есть они этого не могут. Таким образом, чтобы применить margin к изображению, мы должны дать изображению блочное поведение с помощью display: block;.

**Примечание:**Приведенные выше инструкции предполагают, что вы используете изображение меньшей ширины, чем заданная для [<body>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/body) (600 пикселей). Если ваше изображение больше, тогда оно выйдет за границы [<body>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/body) и займет пространство страницы. Чтобы исправить это, вы можете 1) уменьши ширину изображения используя [графический редактор](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B2%D1%8B%D0%B9_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%80%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80), или 2) изменить размер изображения используя CSS путем установки свойства [width](https://developer.mozilla.org/ru/docs/Web/CSS/width" \o "Свойство CSS width устанавливает ширину элемента. По умолчанию она равняется ширине внутренней области, но если box-sizing имеет значение border-box, то она будет равняться ширине области рамки.) для <img> элемента меньшего значения (например 400 px;).

**Примечание**: Не стоит беспокоиться, если вы еще не понимаете display: block; и различия между блочным/строчным. Вы поймете, когда будете изучать CSS более подробно. Вы можете узнать больше о различных доступных значениях display на нашей странице о свойстве [display](https://developer.mozilla.org/ru/docs/Web/CSS/display).

**Заключение**

Если вы следовали всем инструкциям в этой статье, вы должны получить страницу, которая выглядит примерно так:



Задание для самопроверки:

Попрактикуйся в выборе элементов на странице. Вот пример:

Задание: выбери элементы **p**и задай им цвет текста #333  
Решение:

p {

color: #333;

}

Подсказки:

* цвет текста - color
* цвет фона - background-color
* ширина - width

задай всем элементам **p**цвет текста **#999.**  
Код: <https://codepen.io/mlshv/pen/yQmqaL?editors=1100>

Задание: задай всем элементам **div** цвет фона **red**и цвет текста **#fff.**  
Код: <https://codepen.io/mlshv/pen/gQVKNe?editors=1100>

Задание: задай всем элементам **img** ширину 200px.  
Код: <https://codepen.io/mlshv/pen/XyvBjd?editors=1100>