

Получить ширину и высоту

// Самый простой вариант – использование линейки, просто нажимаем комбинацию клавиш `shift + i` до тех пор, пока на панели инструментов активным инструментом не станет линейка, при измерении ширина выводится сверху экрана

// Если сверху экрана ничего не выводится проверьте выбран ли у вас пункт "параметры" в меню "окно"

// Также можно использовать инструмент "выделение области", горячая кнопка (M). При выделении области за курсором будет следовать рамка, с выводом высоты и ширины выделенной области, осторожно, как только вы отпустите мышку рамка исчезнет =)

// Если значения выводятся не в px, а в pt или cm, необходимо настроить вывод данных

// rus: Редактирование – Установки – Основные. Выбираем "Единицы измерения" и "линейки" заменяем pt на px; "Линейки" и "Текст" – пикс и жмем кнопку ok

// eng: Edit – Preferences – Units & Rulers. Выбираем "Rulers" и "Type" – pixels жмем кнопку ok



Быстро найти нужный слой

// Не можешь найти нужный слой? Ничего страшного, просто выбери инструмент "перемещение" (V), убедись что опция "Auto-select" | Автовыбор" активна (верхний левый угол) и выбран "слой".

// Если не можешь найти "Auto-select | Автовыбор" на экране - убедись, что пункт "окно" - "параметры" активен.

// Теперь стоит только выделить элемент, он сразу выберется в окне со слоями.

// Если вы потеряли окно со слоями выберите: "окно" - "слои" или (f7)

// К полезным окнам для front-end разработчика я отношу: "символ", "слои", "цвет" и "история". Также "рамка", "параметры" и "инструменты". Если тебе известны другие полезные окна просто дай мне знать =)



Копируем картинки

// Для сохранения картинок из шаблона можно использовать инструмент "Рамка" (С)

// Также многие (не все) картинки в макете можно выделить в новый документ и сохранить его, что занимает намного меньше времени, для этого нужно:

// // Выделить нужный слой в списке слоев или с помощью инструмента "перемещение" (V). Описание ищи выше;

// // (Ctrl + A) для выделения этого слоя,

// // (Ctrl + C) для копирования этого слоя,

// // (Ctrl + N) для создания нового документа (размер в ширину/высоту будет соответствовать выделенному слою),

// // (Ctrl + V) для вставки слоя,

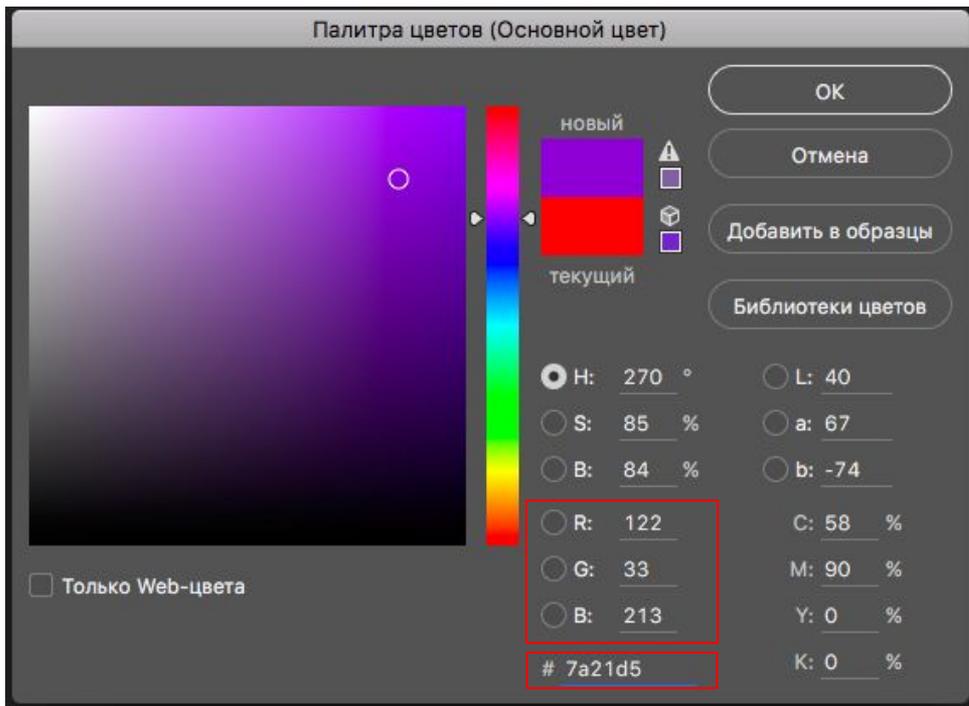
// // (Ctrl + Shift + Alt + S) в Windows для быстрого экспорта нового изображения

// проблемы могут возникнуть из-за векторных масок или перекрытия слоя.

// Так же, если не снять выделение слоя, который копировали раньше, могут не срабатывать некоторые команды, чтобы этого не произошло, перед сохранением

картинки выбери инструмент "Выделение Области" (M) и нажми в любом месте макета.

Цвета в CSS



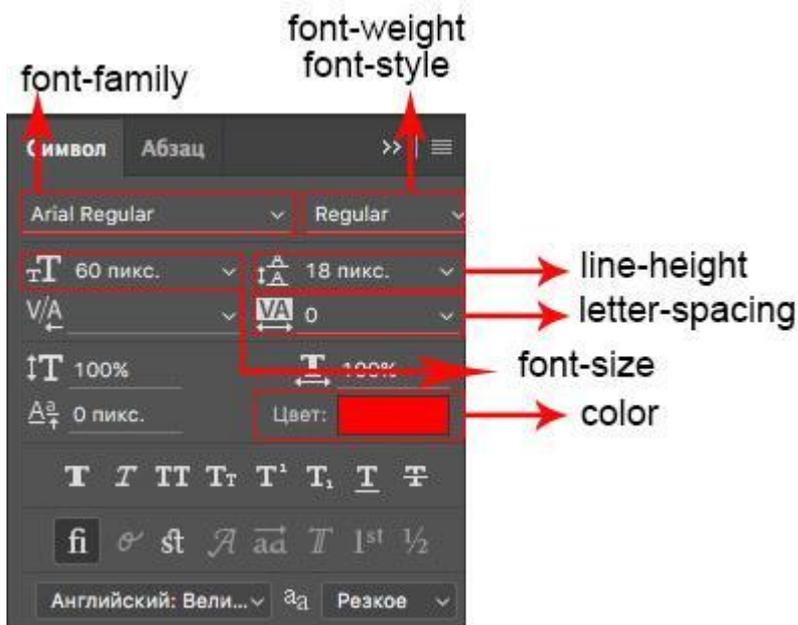
```
// HEX цвет обозначается хэштегом  
"#" и записывается в  
шестнадцатеричной системе счисления,  
белый цвет обозначается #ffffff или  
#fff, а черный #000000 или #000  
// #33aa55 === #3a5
```

```
// RGB цвет обозначается:  
// color: rgb('red', 'green', blue')  
// color: rgb(255,255,255) - белый  
// color: rgb(0,0,0) - черный
```

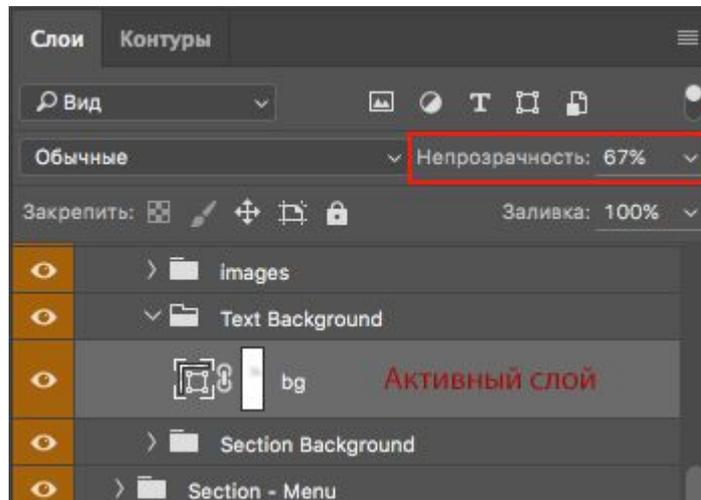
```
// RGBA отличается от RGB только  
четвертым параметром, который  
отвечает за прозрачность, где 1 -  
непрозрачная заливка, а 0 -  
полностью прозрачный цвет. Дробную  
часть можно указывать как 0.4, так и  
просто .4
```

```
// color: rgba(122, 33, 213, .6);
```

Шрифты



Непрозрачность



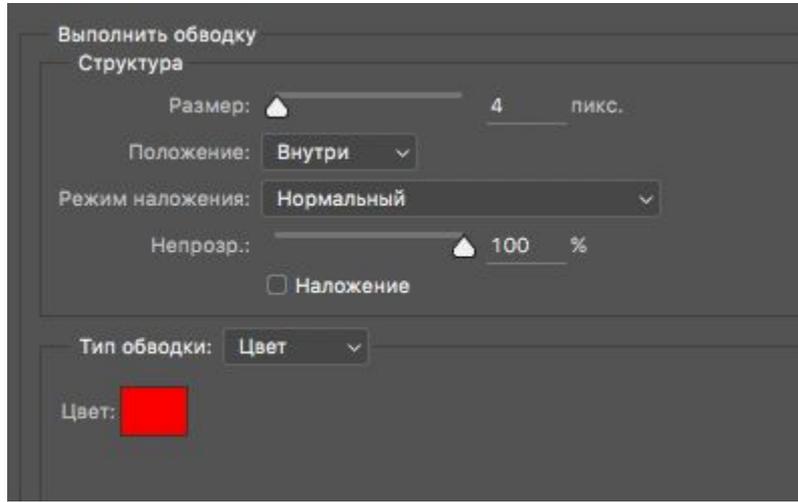
// В данном случае у блока
будет opacity .67 || 0.67,
или заливка RGBA(xxx, xxx,
xxx, .67)

Стили слоя



```
// У слоя также могут быть свои дополнительные стили.  
// Если у слоя есть такая иконка, значит на него влияют дополнительные стили,  
// чтобы их увидеть дважды кликни по слою.  
// Откроется диалоговое окно с перечислением свойств. Нам интересны только:  
// // "Выполнить Обводку | Stroke";  
// // "Внутренняя Тень | Inner Shadow";  
// // "Наложение Цвета | Color Overlay";  
// // "Наложение Градиента | Gradient Overlay";  
// // "Тень | Drop Shadow"  
  
// Рассмотрим их по порядку
```

“Выполнить Обводку” или “Stroke”



// Обводка - это обычный border, нам надо только указать размер и цвет, в 99% случаев это будет solid

// Но обрати внимание на поле "Режим Наложения | position", используй box-sizing если это необходимо

// на примере следующие стили:

```
.element {  
  box-sizing: border-box;  
  border: 4px solid red;  
}
```

Разбираемся с углом для теней



// Для параметров сдвига тени по осям X и Y не стоит досконально рассчитывать градус, я обычно округляю выставленный угол до 0° , 45° , 90° , 135° , 180°
(в примере выше я округляю до 135°)

// Для 0° , 90° и 180° все просто: Для 90° свет падает сверху вниз, значит $X = 0$, а Y будет положительным числом. Для 0° и 180° свет падает сбоку, значит $Y = 0$, а X будет изменяться в зависимости от того, с какой стороны падает свет:

// //если слева-направо (180°), то Y будет положительным,

// //если наоборот (0°), то отрицательным.

// Для 45° и 135° можно представлять с какой стороны падает свет, или просто вычислять разницу

// Прим. 45° : $0^\circ = (-X, 0)$; $90^\circ = (0, Y)$;

// // $X =$ от $-X$ до $0 = -X$;

// // $Y =$ от 0 до $+Y = +Y$;

// Получаем 45° $(-X, +Y)$;

// Для 135° $(+X, +Y)$;

// Если все еще сложно воспользуйся таблицей ниже =)

Градусы	X	Y
180°	+X	0
-135°	+X	-Y
-90°	0	-Y
-45°	-X	-Y
0°	-X	0
45°	-X	+Y
90°	0	+Y
135°	+X	+Y
180°	+X	0

// Осталось понять, сколько нам надо ставить пикселей в эти поля. С этим нам поможет поле "Distance | Смещение".

// Если одно из значений сдвига = 0, то второе сразу принимает значение поля "Distance",

// Если оба значения не равно 0, то каждое равно примерно 2/3 значения "Distance".

*// Прим 1: distance = 35px; angle = 45°;
// // box-shadow: -23px 23px 0 0 rgba(...);*

*// Прим 1: distance = 10px; angle = 90°;
// // box-shadow: 0 10px 0 0 rgba(...);*

“Внутренняя Тень” или “Inner Shadow”

// Для параметров сдвига по осям X и Y смотри описание выше

// Если значение в поле “Размах | Spread” больше нуля (что бывает редко), то мы должны сразу заполнить <радиус размытия> <растяжение> по следующей формуле:

// Где BR - <радиус размытия>, а R <растяжение>

// // $R = \text{“Spread | Размах”} / 100 * \text{“Size | Размер”}$

// // $BR = \text{“Size | Размер”} - R$

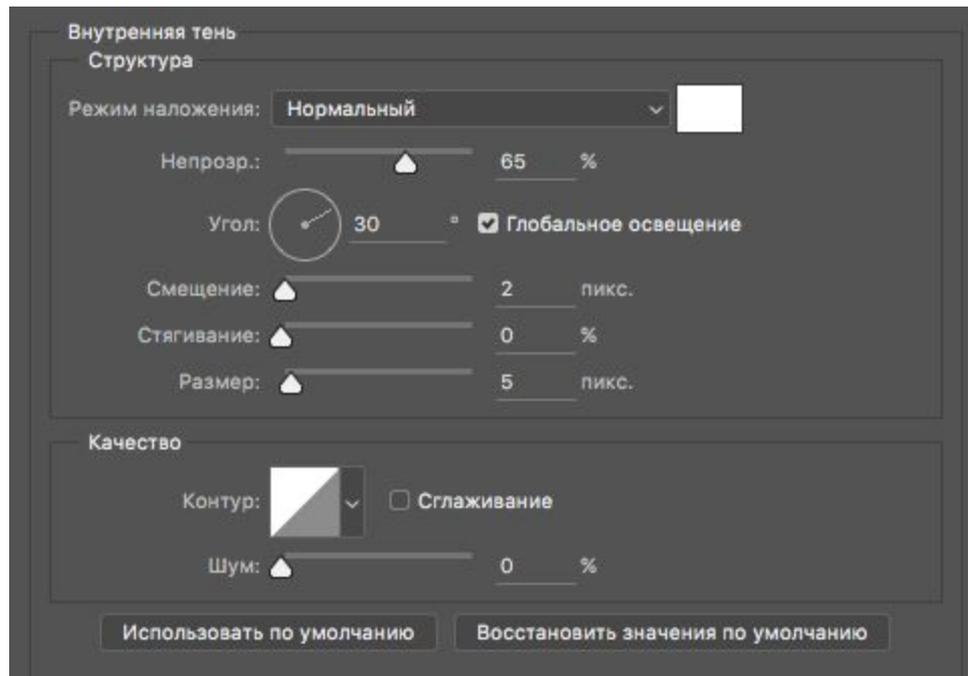
// Прим. “Spread” = 2%; Size = 7px;

// $R = 0.14px$; $BR = 6.86px$; `box-shadow: 0 0 6.86px 0.14px rgba(...);`

// Если значение в поле “Размах | Spread” равно нулю, Третий параметр <радиус размытия> выставляем тот, что указан в поле “Размер | Size”. А <растяжение> можно либо просто опустить, либо передать равным нулю

// С цветом ничего нового, смотрим параметр непрозрачности, и если он менее 100%, записываем в формате `rgba()`.

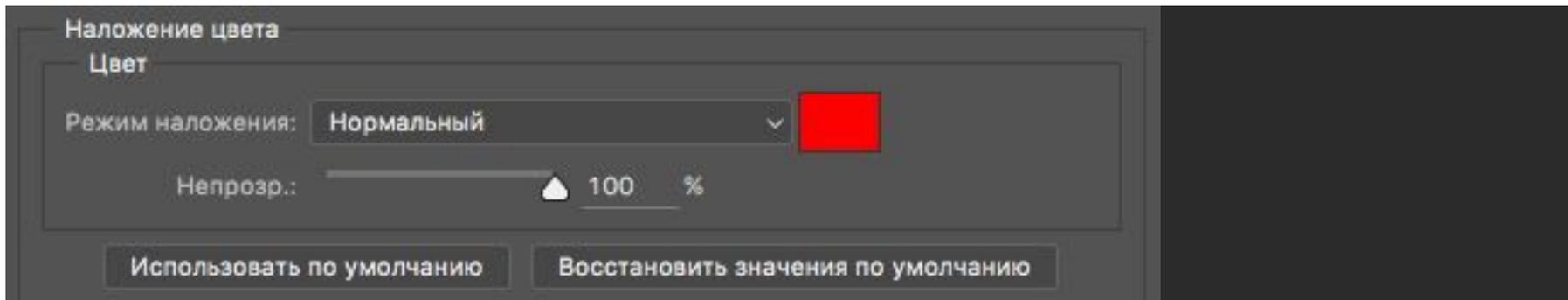
// Не забываем поставить `inset` в конце свойства, тк речь идет о внутренней тени



// Пример использования:

```
box-shadow: 4px 4px 2px 0px rgba(255, 255, 255, 0.65) inset;
```

“Наложение Цвета” или “Color Overlay”



// Наложение цвета - это обычная заливка: для блока это свойство `background-color`; для `svg` элемента может быть `fill`.

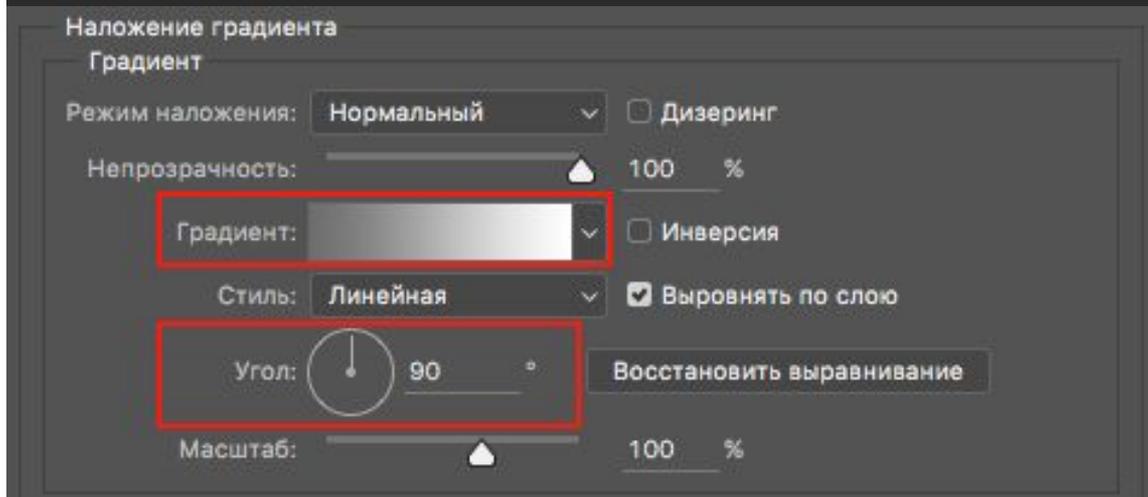
//Если непрозрачность менее 100% необходимо указывать цвет в формате `rgba()`

```
background-color: red | #f00 | #ff0000 | rgb(255, 0, 0);
```

“Наложение Градиента” или “Gradient Overlay”

// На этом свойстве слоя можно найти только обозначение и направление для градиента (обычно оно вертикальное, горизонтальное или диагональное)

// Для перехода в окно настроек градиента нам необходимо нажать на поле “градиент” (выделено красным сверху)



Настройка градиента

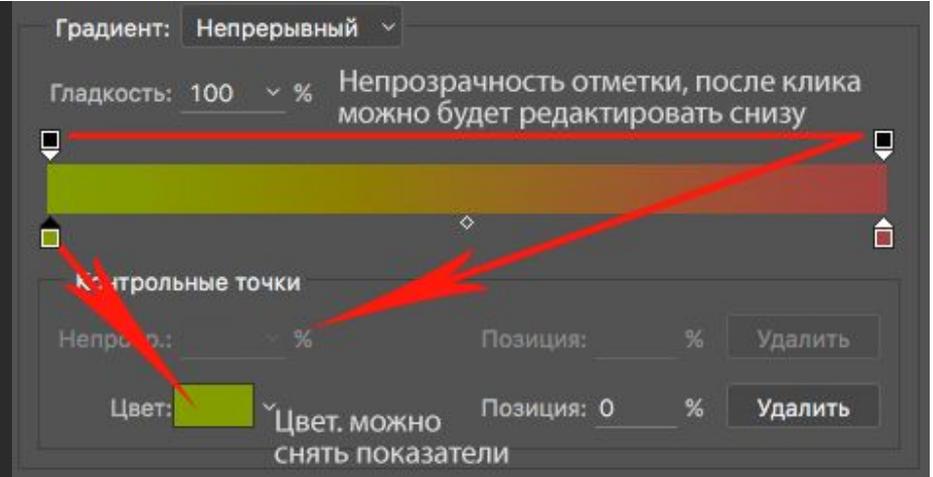
// В окне редактора градиента нам важно знать сколько есть точек, какой у каждой из них цвет и прозрачность.

// Точек в теории может быть много, но обычно получается две.

// Заходим с этими данными на сайт <http://www.colorzilla.com/> выбираем "Gradient Generator".

// Интерфейс приложения очень сильно похож на интерфейс окна редактора градиентов photoshop, поэтому сложностей вызвать не должен. В окне справа мы увидим кроссбраузерные CSS стили этого градиента.

// Рекомендую убрать галку "Comments" и удалить последнюю строку (начиная от `filter: ...` и до конца) если, конечно нет необходимости поддерживать IE6 - IE9.



“Тень” или “Drop Shadow”

// Для решения задачи с наложением теней смотри слайд “Выполнить Обводку” или “Stroke”. Все идентично, за исключением того, что inset указывать не надо.

Конвертируем иконки из psd макета в svg

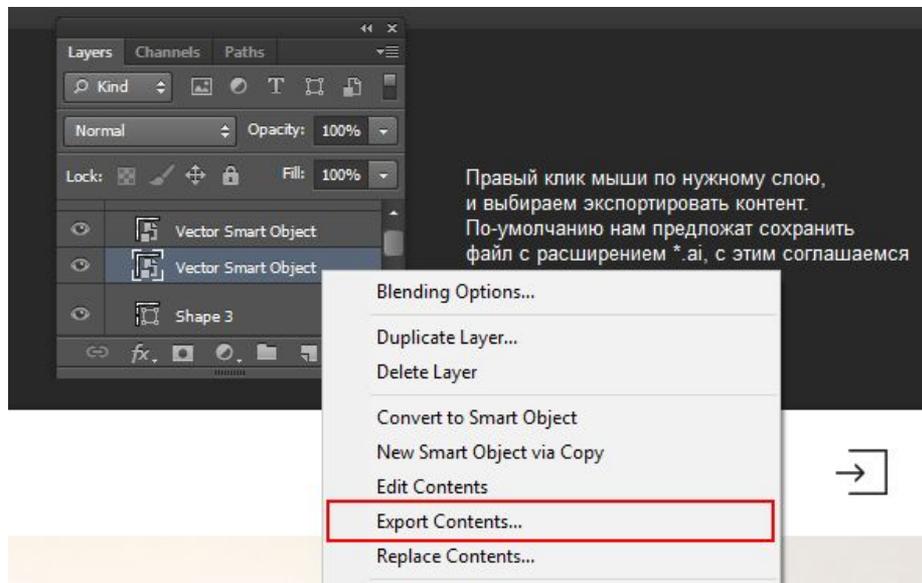
Для этого нам понадобятся adobe photoshop и adobe illustrator.

100% гарантировать успех конвертации никто не может, все напрямую зависит от дизайнера, который делал макет и немного от удачи).

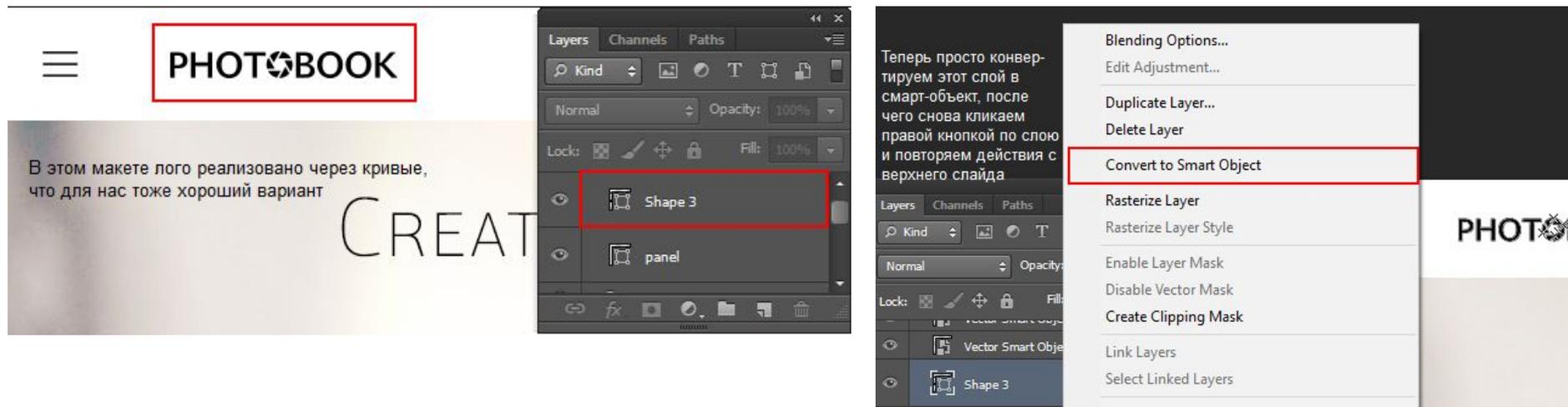
Если нам повезло, и слой уже реализован как vector smart object, то все получится просто.



Затем нам останется только открыть файл
в illustrator и сохранить его как *.svg.



Но что же делать, если слой не является vector smart object? рассмотрим на примере



После сохранения файла в формате *.svg его нужно будет открыть в любом текстовом документе. И да, там будет много символов, возможно очень много, не пугайтесь - это нормально.

Копируем все, начиная от открывающего тега <svg> и вставляем в ту часть html документа, где должна отображаться иконка.

Удаляем лишнее: для тега svg в этом документе используется больше атрибутов, чем нам необходимо, поэтому оставляем только viewBox, enable-background и fill (если он есть), также можно оставить атрибут version.

Далее в стилях указываем нужную ширину и высоту, обновляем страницу, смотрим на результат.

Если же слой сделан как обычный растровый, к сожалению мы его конвертировать не сможем, тк даже после сохранения в svg он останется растровым рисунком.