

1. Назначение и функциональные возможности

Веб-приложение реализует интерактивный «цветовой конвертер» между тремя цветовыми моделями:

- **RGB** (Red, Green, Blue);
- **CMYK** (Cyan, Magenta, Yellow, Key/Black);
- **HSV** (Hue, Saturation, Value).

Дополнительно используется шестнадцатеричное представление цвета («HEX») как вспомогательное, через которое работает встроенная палитра браузера (`<input type="color">`).

Основные свойства системы:

1. Цвет можно задать **в любой из трёх моделей**, а также выбрать из стандартной **палитры** (HEX).
2. После изменения цвета в **любой** модели он автоматически **пересчитывается во все остальные**.
3. В каждой модели доступны числовые значения (компоненты цвета), которые всегда остаются согласованными между собой.
4. Значения копируются в буфер обмена по формуле:
 - `rgb(R, G, B)`
 - `cmk(C%, M%, Y%, K%)`
 - `hsv(H, S%, V%)`

При этом интерфейс использует следующую систему диапазонов:

- RGB:
 $R, G, B \in \{0, 1, \dots, 255\}$
- CMYK (в процентах):
 $C, M, Y, K \in [0, 100]$
- HSV:
 $H \in [0, 360], \quad S, V \in [0, 100]$
- HEX: `#RRGGBB` или `#RGB`.

2. Общий принцип конвертации

Внутреннее «базовое» представление цвета в приложении — это **RGB**-вектор:

$$(R, G, B), \quad R, G, B \in [0, 255] \cap \mathbb{Z}.$$

Все остальные модели рассматриваются как функции от RGB:

- из любой модели (CMYK, HSV, HEX) сначала выполняется перевод **в RGB**,
- затем из RGB вычисляются значения в остальных моделях.

Схематично:

$$\text{ввод пользователя} \xrightarrow{\text{нормализация}} RGB \xrightarrow{\text{формулы}} \begin{cases} \text{CMYK} \\ \text{HSV} \\ \text{HEX} \end{cases}$$

Во всех формулах для удобства сначала используется **нормализация** к диапазону $[0, 1]$, а затем результат масштабируется обратно к диапазонам $[0, 255]$, $[0, 100]$ и т.п., после чего значения **округляются до целых** (там, где это требуется) и **ограничиваются** (clamp) по допустимому диапазону.

3. Математика конвертации между моделями

3.1. Нормализация RGB

Задано:

$$R, G, B \in \{0, \dots, 255\}.$$

Вводим нормализованные величины:

$$r = \frac{R}{255}, \quad g = \frac{G}{255}, \quad b = \frac{B}{255}, \quad r, g, b \in [0, 1].$$

Все дальнейшие расчёты в CMYK и HSV ведутся через r, g, b .

3.2. Перевод RGB → CMYK

Используется классическая модель для полиграфии, в которой CMYK выражается как доля голубого, пурпурного, жёлтого и чёрного красителя.

1. Сначала вычисляется компонент **К** (key, «чёрный»):

$$K = 1 - \max(r, g, b).$$

2. Если $K = 1$ (т.е. $r = g = b = 0$, чистый чёрный), то:

$$C = 0, \quad M = 0, \quad Y = 0, \quad K = 1.$$

3. Во всех остальных случаях:

$$C = \frac{1 - r - K}{1 - K}, \quad M = \frac{1 - g - K}{1 - K}, \quad Y = \frac{1 - b - K}{1 - K}.$$

4. Для вывода используются проценты:

$$C_{\%} = \text{round}(100 \cdot C), \quad M_{\%} = \text{round}(100 \cdot M), \quad Y_{\%} = \text{round}(100 \cdot Y), \quad K_{\%} = \text{round}(100 \cdot K).$$

Здесь `round` обозначает привычное округление до ближайшего целого; затем результат дополнительно ограничивается диапазоном $[0, 100]$.

3.3. Перевод CMYK → RGB

На входе пользователь задаёт:

$$C_{\%}, M_{\%}, Y_{\%}, K_{\%} \in [0, 100].$$

Они переводятся в доли:

$$C = \frac{C_{\%}}{100}, \quad M = \frac{M_{\%}}{100}, \quad Y = \frac{Y_{\%}}{100}, \quad K = \frac{K_{\%}}{100}, \quad C, M, Y, K \in [0, 1].$$

Затем вычисляются нормализованные RGB-значения:

$$r = (1 - C)(1 - K), \quad g = (1 - M)(1 - K), \quad b = (1 - Y)(1 - K).$$

И далее:

$$R = \text{round}(255 \cdot r), \quad G = \text{round}(255 \cdot g), \quad B = \text{round}(255 \cdot b),$$

после чего значения ограничиваются диапазоном $[0, 255]$ и используются как базовое RGB-представление.

3.4. Перевод RGB → HSV

Снова берутся нормализованные компоненты $r, g, b \in [0, 1]$.

Обозначим:

$$M = \max(r, g, b), \quad m = \min(r, g, b), \quad \Delta = M - m.$$

1. **Value (яркость):**

$$V = M.$$

2. **Saturation (насыщенность):**

$$S = \begin{cases} 0, & \text{если } M = 0, \\ \frac{\Delta}{M}, & \text{иначе.} \end{cases}$$

3. **Hue (оттенок)** — угол на цветовом круге (в градусах):

Если $\Delta = 0$ (серый цвет), то оттенок считается неопределённым; в приложении в этом случае берётся

$$H = 0.$$

Иначе:

$$H = \begin{cases} 60^\circ \cdot \frac{g - b}{\Delta} \pmod{360}, & \text{если } M = r, \\ 60^\circ \cdot \left(\frac{b - r}{\Delta} + 2 \right), & \text{если } M = g, \\ 60^\circ \cdot \left(\frac{r - g}{\Delta} + 4 \right), & \text{если } M = b. \end{cases}$$

После вычисления H приводится к диапазону $[0, 360)$.

4. Для вывода используются проценты:

$$S_{\%} = \text{round}(100 \cdot S), \quad V_{\%} = \text{round}(100 \cdot V),$$

с последующим ограничением в $[0, 100]$. Итоговая тройка:

$$(H, S_{\%}, V_{\%}).$$

3.5. Перевод HSV → RGB

На входе заданы:

$$H \in [0, 360], \quad S_{\%}, V_{\%} \in [0, 100].$$

Нормализация:

$$S = \frac{S_{\%}}{100}, \quad V = \frac{V_{\%}}{100}.$$

Вычисляется:

$$h' = \frac{H \bmod 360}{60}, \quad i = \lfloor h' \rfloor, \quad f = h' - i.$$

Вводятся вспомогательные величины:

$$p = V(1 - S), \quad q = V(1 - fS), \quad t = V(1 - (1 - f)S).$$

Далее выбирается один из шести случаев (сектор цветового круга):

$$(r, g, b) = \begin{cases} (V, t, p), & i = 0, \\ (q, V, p), & i = 1, \\ (p, V, t), & i = 2, \\ (p, q, V), & i = 3, \\ (t, p, V), & i = 4, \\ (V, p, q), & i = 5 \text{ (или иначе, по умолчанию)}. \end{cases}$$

В результате $r, g, b \in [0, 1]$, из них получаем RGB:

$$R = \text{round}(255 \cdot r), \quad G = \text{round}(255 \cdot g), \quad B = \text{round}(255 \cdot b),$$

с ограничением $[0, 255]$. Эта тройка снова становится базовым состоянием цвета.

3.6. Перевод HEX ↔ RGB (для палитры)

Палитра браузера оперирует шестнадцатеричным представлением:

- #RRGGBB — по две шестнадцатеричные цифры на канал;
- #RGB — сокращённая форма, где каждая цифра дублируется (например, #F03 → #FF0033).

HEX → RGB:

1. Строка очищается от символа #.
2. Если длина равна 3, то каждая цифра удваивается: "abc" → "aabbcc".

- Получается строка из 6 символов `RRGGBB`, она интерпретируется как шестнадцатеричное число:

$$N = \text{int}(\text{"RRGGBB"}, 16).$$

- Компоненты:

$$R = (N \gg 16) \& 255, \quad G = (N \gg 8) \& 255, \quad B = N \& 255.$$

Далее (R, G, B) используется как базовый цвет.

RGB → HEX:

- Каждая компонента $R, G, B \in [0, 255]$ приводится к двум шестнадцатеричным цифрам с ведущим нулём при необходимости.
- Строки конкатенируются: `"#" + RR + GG + BB`.

HEX нужен только для работы палитры; все дальнейшие вычисления выполняются в RGB/CMYK/HSV.

4. Обработка данных и устойчивость

4.1. Ограничение диапазонов (clamping)

Во всех местах ввода значения ограничиваются:

- RGB: $0 \leq R, G, B \leq 255$,
- CMYK: $0 \leq C, M, Y, K \leq 100$,
- HSV: $0 \leq H \leq 360, 0 \leq S, V \leq 100$.

Это предотвращает:

- уход значения за диапазон из-за ручного ввода;
- накопление ошибок округления при многократных преобразованиях.

4.2. Округление и обратимость

Поскольку пользовательские значения представляются целыми (и для CMYK/HSV в процентах), точная взаимная обратимость невозможна; преобразования **обратимы с точностью до округления**.

Общий принцип:

- все промежуточные вычисления ведутся в плавающей точке (`double` / `float`);
- перед выводом выполняется `round` и последующий `clamping`.

Это гарантирует численно устойчивое поведение при многократных изменениях цвета.

5. Итог

Приложение реализует полный цикл конвертации цвета между моделями **RGB**, **CMYK** и **HSV**, используя RGB как внутреннее представление и стандартные математические формулы для перехода между моделями.

Выбор цвета возможен:

- числовым вводом (в любой из трёх моделей),
- «графически» через стандартную палитру (HEX-значение).

Каждое изменение приводит к пересчёту всех остальных представлений по описанным выше алгоритмам, что позволяет использовать конвертер как универсальный инструмент для работы с различными цветовыми системами.