

# Tvorba barevných ikon pro Delphi

---

Tip: Tlačítko ? v záhlaví aplikace otevře tento dokument (PDF) v novém okně prohlížeče.

## Účel nástroje

---

Nástroj slouží k převodu HTML/SVG barev a SVG/PNG obrázků do dvou odvozených variant:

- **Disabled** — světle šedá varianta barvy pro vypnuté nebo neaktivní prvky
- **Hot** — zvýrazněná varianta barvy pro aktivní, hover nebo zvýrazněný stav

Aplikace funguje jako samostatná HTML stránka s JavaScriptem. Nevyžaduje serverovou část. Pro hromadné ZIP exporty používá knihovnu **JSZip** načítanou z CDN.

## Podporované vstupy

---

### Textový vstup

Do textového pole lze vložit jednu barvu, více barev nebo celý SVG/HTML/CSS text.

Podporované tvary barev:

- zkrácený HEX s mřížkou: #09c
- zkrácený HEX s alfa kanálem a mřížkou: #09c8
- plný HEX s mřížkou: #336699
- plný HEX s alfa kanálem a mřížkou: #336699cc
- RGB: rgb(51, 102, 153)
- RGBA: rgba(51, 102, 153, 0.8)
- HSL: hsl(120, 100%, 50%) i moderní zápis hsl(120 100% 50%)
- HSLA: hsla(120, 100%, 50%, 0.8) i hsl(120 100% 50% / 0.8)

- pojmenované HTML barvy: `red`, `white`, `transparent`, `rebeccapurple`, ... (všech 148 standardních CSS jmen)

Detekce barev je úmyslně přísná — vyžaduje explicitní prefix (`#`, `rgb(`, `rgba(`, `hsl(`, `hsla(`) nebo přesný název pojmenované barvy. **Holý HEX bez # není rozpoznáván**, protože by se v SVG/CSS zákonitě chytaly číselné atributy typu `width="320"` nebo `cx="220"` (3-znakový validní HEX).

U HSL hodnoty hue lze použít jednotky `deg` (výchozí), `rad`, `grad`, `turn`. Vstupní formát se v tabulce zachovává — z RGB se vrátí RGB, z HSL HSL, z HEX HEX. Pojmenované barvy se převádí pouze ve směru `name` → **HEX** (zpět nikdy).

Pojmenovaná barva uvnitř SVG kódu se převede jen v typickém barevném atributu (`fill`, `stroke`, `stop-color` apod.) nebo uvnitř CSS hodnoty — aby se z `id="redCircle"` nestal pokus o převod barvy „red“. V SVG attributech se výstupní HEX zapisuje vždy s mřížkou (`#AABBCC`).

## Přetažení nebo otevření obrázků

Do drop zóny lze přetáhnout jeden nebo více souborů:

- `*.svg`
- `*.png`

Stejné typy souborů lze otevřít tlačítkem **Otevřít obrázky**.

## Výpočet disabled barvy

---

Disabled varianta převádí barvu na světle šedou, alfa kanál zachovává.

Postup:

- Barva se rozdělí na kanály `R`, `G`, `B` v rozsahu 0–255
- Spočítá se průměr:  $C = (R + G + B) \div 3$
- Spočítá se zesvětlená šedá:  $C2 = 255 - ((255 - C) \div 2)$
- Výsledek je `c2c2c2`

## Příklad:

R = 51  
G = 102  
B = 153  
C = (51 + 102 + 153) div 3 = 102  
C2 = 255 - ((255 - 102) div 2) = 179  
Výsledek: #B3B3B3

## Výpočet hot barvy

---

Hot varianta zvyšuje intenzitu barvy.

Použitý postup:

- vypočítá se průměr RGB kanálů
- odchylka každého kanálu od průměru se zvýší koeficientem **1,35**
- výsledná barva se mírně přiblíží k bílé koeficientem **0,12**
- alfa kanál se nemění

## Tabulka variant

---

Po převodu se zobrazí tabulka se třemi sloupci:

| Normální barvy | Disabled barvy | Hot barvy |
|----------------|----------------|-----------|
|----------------|----------------|-----------|

V každém řádku je vzorek barvy a její textové zápisy ve **všech formátech**, v **jakých byla v textu nalezena**. Pokud se stejná barva ve vstupu objeví víckrát ve stejném formátu, vypíše se jen jednou (bez ohledu na počet výskytů).

Krátký HEX ( #RGB , #RGBA ) se vždy rozepíše na dlouhý ( #RRGGBB , #RRGGBBAA ) — preferujeme delší kanonický zápis.

Plně neprůhledné varianty stejné barvy v různých zápisech se slučují do **jednoho řádku** se všemi formáty na samostatných řádcích v buňce. Příklad: vstup #369 , #336699 a rgb(51, 102, 153) vytvoří jeden řádek, jehož „Normální barvy“ buňka obsahuje:

```
#336699  
rgb(51, 102, 153)
```

Sloupce „Disabled barvy“ a „Hot barvy“ zrcadlí tytéž formáty, jen s transformovanými hodnotami (například #B3B3B3 a rgb(179, 179, 179) pro disabled variantu).

Tabulka se používá jednotně pro jakýkoli textový vstup — od jednotlivých barev přes CSS úryvky až po celý SVG kód. Pro práci s obrázky (přetažený SVG/PNG) má aplikace samostatný „Obrázkový náhled“.

## Detail barvy (popup)

Klepnutí na vzorek otevře překryvné okno s velkým náhledem a textovými zápisy v různých formátech:

- pokud zdrojová barva **nemá** alfa kanál (nebo je plný FF / 255 / 1.0): **HEX** ( #RRGGBB ), **RGB** ( rgb(r, g, b) ), **HSL** ( hsl(h, s%, l%) )
- pokud zdrojová barva **má** alfa kanál pod 1.0: **HEX** s alfou ( #RRGGBBAA ), **RGBA** ( rgba(r, g, b, a) ), **HSLA** ( hsla(h, s%, l%, a) )

Klepnutí na řádek zápisu zkopíruje hodnotu do schránky (vizuálně potvrzené krátkým zeleným zvýrazněním). Okno se zavře křížkem, klepnutím na pozadí, nebo klávesou **Esc**.

## Režim Ukázka

---

Tlačítko **Ukázka** přepne stránku do vizuálního režimu. V režimu ukázky se tlačítko změní na **Zpět**.

## Chování obrázku

Zobrazuje se:

- SVG ze vstupu
- nebo generovaná geometrie
- nebo načtené PNG/SVG

Ovládání:

- normální stav = původní
- hover = hot
- klik = disabled
- klik z disabled = hot (pokud je kurzor nad objektem)
- odjetí myši = návrat na normální (pokud není disabled)

Při více obrázcích se zobrazí mřížka.

## Pozadí ukázky

---

### HLS výběr

- osa X = odstín (H)
- osa Y = světlost (L)
- saturace samostatně pod čtvercem

Hover = stínové kolečko Klik = fixace

### Saturace

- vlevo = nulová saturace
- vpravo = plná saturace

### Editace barvy pozadí

Pole pro barvu pozadí ukázky je benevolentnější než hlavní konvertor — automaticky doplní # , pokud chybí, takže lze napsat:

- #RRGGBB nebo RRGGBB
- #RRGGBBAA nebo RRGGBBAA
- #RGB nebo RGB
- #RGBA nebo RGBA
- rgb(r,g,b) / rgba(r,g,b,a)

- `hsl(h,s%,l%) / hsla(h,s%,l%,a)`
- pojmenované barvy: `red` , `white` , `transparent` , ...

Příklad:

```
rgba(175,255,77,0.5) -> #AFFF4D80
```

- alfa se při vykreslení pozadí ignoruje (pozadí je vždy plně neprůhledné)
- normalizace na kanonický `#RRGGBB[AA]` zápis proběhne až při `blur/Enter`

## Předdefinovaná pozadí

- Word-like paleta
- standardní barvy
- patterny
- SVG scény (krajina, les, město...)

Vybrané pozadí má tučný rámeček.

## Obrázkový náhled

---

Po načtení souborů:

- původní
- disabled
- hot

## PNG vstup

- fyzická úprava pixelů
- alfa zachována

Tlačítko:

- [Stáhnout PNG](#)

## SVG vstup

Tlačítka:

- [Stáhnout SVG](#)
- [Stáhnout PNG](#)

PNG je generováno přes canvas (transparentní).

## Export SVG jako PNG

---

Možnosti:

- šířka
- výška

Chování:

- zachování poměru stran
- vystředění
- průhledné pozadí
- žádné opakování

## Suffixy názvů

---

- velikost: `_32` , `_80x60`
- normal: *(prázdné)*
- disabled: `_d`
- hot: `_h`

Příklad:

ikona.svg  
ikona\_d.svg  
ikona\_h.svg  
ikona\_32.png  
ikona\_32\_d.png  
ikona\_32\_h.png

## Hromadné stahování

---

### PNG dávka

- Původní PNG
- Disabled PNG
- Hot PNG
- Vše PNG

### SVG dávka

- Původní SVG
- Disabled SVG
- Hot SVG
- Vše SVG
- Vše PNG

Volba **Rozdělit do složek:**

normal/  
disabled/  
hot/

## Technické poznámky

---

### SVG

- textová transformace
- regex-based parsing
- chráněné atributy

Podporované:

- fill
- stroke
- color
- stop-color
- flood-color
- lighting-color
- background-color

## Izolace SVG identifikátorů v náhledech

Při zobrazení více variant stejného SVG ve stránce může dojít ke kolizi identifikátorů, například u gradientů, masek, filtrů nebo symbolů. Prohlížeč pak může použít první nalezený `id`, typicky z původní verze, takže disabled/hot varianta nemusí vizuálně odpovídat upravenému SVG.

Stejný problém vzniká i u **CSS tříd** — pokud má SVG vlastní `<style>` blok s pravidly typu `.cls-1 { fill: ... }`, prohlížeč aplikuje pravidla globálně, takže styly z hot varianty by ovlivnily i zobrazení normální nebo disabled.

Nástroj proto při zobrazení v náhledu a v režimu Ukázka dočasně přepisuje:

- SVG `id` na unikátní hodnoty + aktualizuje odkazy `url(#id)`, `href="#id"`, `xlink:href="#id"`
- CSS třídy v atributu `class` + selektory `.classname` uvnitř `<style>` bloků

Tato úprava se používá pouze pro zobrazení ve stránce. Exportované SVG soubory zůstávají beze změny — uložené ikony mají původní `id` a názvy tříd.

## PNG

- FileReader
- canvas
- getImageData
- zachování alfa

## ZIP

```
<script src="https://cdn.jsdelivr.net/npm/jszip@3.10.1/dist/jszip.min.
```



## Omezení

---

- není plný CSS parser, parsování je regex-based
- holý HEX bez # není rozpoznáván jako barva (záměrně, viz výše)
- SVG → PNG závisí na canvas API prohlížeče
- externí zdroje v SVG (CSS, fonty, obrázky) se při převodu na PNG nemusí načíst
- izolace SVG `id` a názvů tříd se aplikuje pouze pro interní náhled; exportované SVG soubory zůstávají v původní struktuře

## Doporučené použití

---

### SVG

1. Přetáhnout
2. Zkontrolovat varianty
3. Ověřit na pozadí
4. Nastavit velikost
5. Stáhnout

### PNG

1. Přetáhnout
2. Zkontrolovat
3. Stáhnout

## Údržba

---

Hlídní:

- nepřepisovat nebarevné atributy v SVG
- zachovat alfa u PNG
- správné suffixy
- konzistence ZIP exportu
- nepřepisovat hodnotu při psaní