CSS Variables in Tiki

Example of CSS variable use in themes/base_files/_tiki-pagetop_colors.scss:

```css
// Topbar module zone and parent divs - dark color
.layout_basic .topbar_modules.bg-dark,       // Single Container layout -- Topbar, which
.probably includes another nav menu
.layout_classic .topbar.bg-dark-parent,       // Classic Tiki layout -- Parent of the
topbar module zone
topbar wrapper
{
  background: var(--tiki-topbar-bg-dark) !important;
  color: var(--tiki-topbar-dark-color);
  a:not(.dropdown-item),
  .btn.btn-link {
    color: var(--tiki-topbar-dark-link-color);
    &:hover {
      color: var(--tiki-topbar-dark-link-hover-color);
    }
  }
}
```

Background info on CSS variables

Custom properties (sometimes referred to as CSS variables or cascading variables) are entities defined by CSS authors that contain specific values to be reused throughout a document. They are set using custom property notation (e.g., `--main-color: black;`) and are accessed using the `var()` function (e.g., `color: var(--main-color);`).

See also: https://css-tricks.com/a-complete-guide-to-custom-properties/

CSS variables in Bootstrap

There were just 28 CSS variables in Bootstrap 4 (https://getbootstrap.com/docs/4.0/getting-started/theming/#css-variables). As of Bootstrap 5.2, this number, for root CSS variables, is up to approximately 70 (https://getbootstrap.com/docs/5.2/customize/css-variables/#root-variables). Root CSS variables have global scope. There are also component variables, with narrow scope, such as those used for tables and navbars. (I just recently became aware of these navbar CSS variables, so will look again at the Tiki CSS variables implementation to try to be as coherent as possible.

The Tiki CSS variable syntax

Just a short time ago we realized it would be useful to distinguish SCSS variables created specifically for Tiki from Bootstrap or other SCSS variables, so we decided on the `--tiki-` prefix, following luci's suggestion. I used the same pattern for our CSS variables, using `--tiki-` (The prepended two hyphens and appended single hyphen are the standard format for CSS variables.)

So far, for the pagetop colors, we have these Tiki CSS variables:

```css
:root {
  --tiki-top-bg-light
  --tiki-top-light-color
  --tiki-top-light-link-color
```
These are very explicit because they are used in the calculation of the site logo image; their scope is limited to header.fixed-top:

```css
header.navbar.fixed-top {
  --tiki-fixed-top-border-top-width
  --tiki-fixed-top-border-right-width
  --tiki-fixed-top-border-bottom-width
  --tiki-fixed-top-border-left-width
  --tiki-fixed-top-light-border-color
  --tiki-fixed-top-dark-border-color
}
```

Related, we also have these for site title and subtitle, in fixed-top position (Basic Bootstrap layout) and static position (Classic Tiki and Single Container layouts) and their default values:

```css
--tiki-fixed-top-site-title-font-size: 1.8rem;           // Fixed-top position (Basic Bootstrap layout)
--tiki-fixed-top-site-title-font-weight: 700;
--tiki-fixed-top-site-subtitle-font-size: 1.4rem;
--tiki-fixed-top-site-subtitle-font-weight: 400;
--tiki-site-title-font-size: 2rem;                     // Static position (Classic Tiki
As the docs on CSS variables show, they need to be presented in an array. This array, in a theme's files, can/should be placed at the bottom of the theme's variables file. This positions the `:root` array at the beginning of the theme CSS file, where it needs to be, and also after the SCSS variables are defined, which CSS variables often require. This example is from FiveAlive/Orange:

```css
// CSS variables / custom properties for base_files/scss/_tiki-pagetop_colors.scss
// Interpolation info from https://developersink.com/css/css-custom-properties-sass-variables/
// Background images (actual images or CSS gradients, etc.), if assigned, override background colors.
:root {
  --tiki-top-bg-light: linear-gradient(to right, #c44a23 0%,#c64b23 10%,#dc5823 34%,#e96023 58%,#f06423 87%,#f06423 100%);
  --tiki-top-light-color: ${navbar-light-color};
  --tiki-top-light-link-color: ${navbar-light-link-color};
  --tiki-top-light-link-hover-color: ${navbar-light-link-hover-color};

  --tiki-topbar-bg-light: ${topbar-light-color};
  --tiki-topbar-light-color: ${topbar-light-color};
  --tiki-topbar-light-link-color: ${navbar-light-link-color};
  --tiki-topbar-light-link-hover-color: ${navbar-dark-link-hover-color};

  --tiki-top-bg-dark: linear-gradient(to right, #c44a23 0%,#c64b23 10%,#dc5823 34%,#e96023 58%,#f06423 87%,#f06423 100%);
  --tiki-top-dark-color: ${top-dark-color};
  --tiki-top-dark-link-color: ${navbar-dark-link-color};
  --tiki-top-dark-link-hover-color: ${navbar-dark-link-hover-color};

  --tiki-topbar-bg-dark: ${topbar-dark-bg};
  --tiki-topbar-dark-color: ${navbar-dark-color};
  --tiki-topbar-dark-link-color: ${navbar-dark-link-color};
  --tiki-topbar-dark-link-hover-color: ${navbar-dark-link-hover-color};
}
```

**Look & Feel customization**

To redefine one or more of the CSS variables, currently it works to input the variable name and the new value in the Look & Feel admin customization text area, inside the `:root` array.

This could also be in a custom.css file.

**Future possibilities**

- A dedicated Look & Feel "CSS Variables" text area provided so that just the variables could be entered and the root array syntax would be added on the backend.
- New Tiki CSS variables in addition to those for pagetop colors — what other Tiki page elements should easy customization be enabled for?
- Site color controls (or other controls) available to site users on the public side, such as for dark mode, like switching themes but not needed a page refresh — [https://pedromarquez.dev/blog/2022/7/dark-mode-css](https://pedromarquez.dev/blog/2022/7/dark-mode-css)

**References:**

• https://css-tricks.com/a-complete-guide-to-custom-properties/
• https://blog.getbootstrap.com/2022/05/16/using-bootstrap-css-vars/
• https://getbootstrap.com/docs/5.2/customize/css-variables/
• https://getbootstrap.com/docs/4.0/getting-started/theming/#css-variables
• https://pedromarquez.dev/blog/2022/7/dark-mode-css