



# gatsby-design-tokens

npm v4.22.0 downloads 58k/month downloads 1.9M

## Introduction

*Design tokens are the visual design atoms of the design system — specifically, they are named entities that store visual design attributes. We use them in place of hard-coded values (such as hex values for color or pixel values for spacing) in order to maintain a scalable and consistent visual system for UI development.*

This quote from the [Lightning Design System Design Tokens documentation](#) accurately describes the tokens contained in this package — to be a bit more specific: `gatsby-design-tokens` offers *plain objects or arrays of values for related **CSS properties***. Currently,

Gatsby's design tokens are following the [System UI Theme Specification](#) as well as the [Theme UI Theme Specification](#).

## Project state and versioning

`gatsby-design-tokens` is a work-in-progress, but versions *do* follow the [Semantic Versioning](#) specification:

- Minor fixes to tokens will be released as patch versions.
- Major design changes will be released as minor versions
- *Breaking* public API changes will be released in major versions only.

To prevent your site from breaking due to a breaking change or looking dramatically different due to a minor version bump, we recommend the `~` comparator when using this package.

## Installation

Using [npm](#):

```
npm install gatsby-design-tokens --save
```

Using [Yarn](#):

```
yarn add gatsby-design-tokens
```

## Tokens 🍒🍋🍏

All exports provide either plain objects or arrays of values for related CSS properties:

```
import {  
  borders,  
  // [ 0, `1px solid`, `2px solid` ]
```

```

breakpoints,
// { xs:`400px`, sm:`550px`, ...}
breakpointsArray,
// [ "400px", "550px", ...]
colors,
// { primary:`#639`, blackFade: { 5: `rgba(...)` , 10: ...}, ...}
fonts,
// { body: `-apple-system, ..., sans-serif`, monospace: {...} }
fontLists,
// { body: [`-apple-system`, ..., `sans-serif`], monospace: [...] }
fontSizes,
// [ `0.75rem`, ..., `5.75rem` ]
fontSizesPx,
// [ `12px`, ..., `92px` ]
fontSizesRaw,
// [ 12, 14, 16, 18, 20, 24, 28, 32, ..., 84, 92 ]
fontWeights,
// { body: 400, semiBold: 600, ..., heading: 700 }
letterSpacings,
// { normal: "normal", tracked: "0.075em", tight: "-0.015em" }
lineHeights,
// { solid: 1, dense: 1.25, ... }
mediaQueries,
// { xs: "@media (min-width: 400px)", ..., xxl: "@media (min-width: 1600px)" }
radii,
// [ 0, "2px", "4px", "8px", "16px", "9999px", "100%" ]
shadows,
// { raised: `0px 1px 2px rgba(46,...)` , floating: `0px 2px 4px...` }
space,
// [ "0rem", "0.25rem", "0.5rem", ..., "4.5rem"]
spacePx,
// [ "0px", "4px", "8px", ..., "72px"]
spaceRaw,
// [ 0, 4, 8, 12, 16, 20, 24, 32, 40, 48, 56, 64, 72]
transition,
// transition = {
//   default: `250ms cubic-bezier(0.4, 0, 0.2, 1)` ,
//   curve: { default: `cubic-bezier(0.4, 0, 0.2, 1)` , ... },
//   speed: { faster: `50ms`, ... },
// }
} from "gatsby-design-tokens"

```

- `rem` values are based on a `font-size` of `16px` for the root element.
- All tokens work in the context of [Theme UI's Theme Scales](#), with the exception of `breakpoints` : Use `breakpointsArray` for `theme-ui` and its [responsive styles](#) feature.

## theme-ui themes 🧠

theme

A basic `theme-ui` theme composed of unmodified tokens, with one exception: `colors` are modified to provide the basic set of variables described in <https://theme-ui.com/theme-spec#color>.

TBD: Adopt the `theme-ui` definitions for the basic `colors` tokens.

```
import { theme } from "gatsby-design-tokens/dist/theme"

// when used with `gatsby-plugin-theme-ui`, export the theme
// as default from `src/gatsby-plugin-theme-ui/index.js`
export { theme as default } from "gatsby-design-tokens/dist/theme"

// in case you need theme tokens outside of the `emotion` context
import {
  breakpoints,
  colors,
  fonts,
  fontSizes,
  fontWeights,
  letterSpacings,
  lineHeights,
  mediaQueries,
  radii,
  shadows,
  space,
  transition,
} from "gatsby-design-tokens/dist/theme"
```

### **theme-gatsbyjs-org**

The theme currently in use on gatsbyjs.com via `gatsby-plugin-theme-ui`:

- Extends the base theme's `colors` with a couple .org-specific things, and provides a `dark` mode (ref. <https://theme-ui.com/color-modes>).
- Adds .org-specific `sizes` and `zIndices`.
- Adds a couple of `variants`.
- Uses `hex2rgba` to create rgba colors.

```
import { theme } from "gatsby-design-tokens/dist/theme-gatsbyjs-org"

// when used with `gatsby-plugin-theme-ui`, export the theme
// as default from `src/gatsby-plugin-theme-ui/index.js`
export { theme as default } from "gatsby-design-tokens/dist/theme-gatsbyjs-org"

// in case you need theme tokens outside of the `emotion` context
import {
  breakpoints,
  colors,
  fonts,
  fontSizes,
  fontWeights,
  letterSpacings,
```

```
    lineHeights,  
    mediaQueries,  
    radii,  
    shadows,  
    sizes,  
    space,  
    transition,  
    zIndices,  
  } from "gatsby-design-tokens/dist/theme-gatsbyjs-org"
```

## Local development

The Gatsby monorepo, which hosts this package, also contains the source for gatsbyjs.com, aka `www` — where `theme-gatsbyjs-org` is in use. Using a little helper called `gatsby-dev` we can develop and test both of them locally.

### 1. Clone the `gatsby` monorepo and set it up for local dev

Follow the [official guide](#) to clone/fork and set up the Gatsby monorepo. This will roughly look like this:

```
# clone the repo/your fork  
git clone https://github.com/gatsbyjs/gatsby.git  
cd gatsby  
  
# set up the repo, install dependencies for `packages`, and build the latter  
yarn run bootstrap  
  
# make sure tests are passing  
yarn test  
  
# create a new feature branch  
git checkout -b topics/new-feature-name
```

#### Install `gatsby-dev-cli` to ease testing your local changes to `packages`

Assuming `gatsby-cli` [is installed](#), let's install [gatsby-dev-cli](#) with

```
yarn global add gatsby-dev-cli
```

`gatsby-dev-cli` needs to know where your local Gatsby repository lives; navigate to its root folder, get the absolute path to it via `pwd`, and tell `gatsby-dev-cli` about it with

```
gatsby-dev --set-path-to-repo /path-to-local-gatsby/gatsby
```

### 2. Get `www` running on `localhost`

```
cd /path-to-local-gatsby/gatsby/www
```

```
yarn # to install dependencies
```

Follow the [README instructions](#) to [add a .env.development file](#) in `www`, and add

```
GATSBY_SCREENSHOT_PLACEHOLDER=true to
```

*[...] skip downloading screenshots and generating responsive images for all screenshots and replace them with a placeholder image*

Run `yarn develop`, which thanks to the previous step shouldn't take ~40 minutes but way less. Hopefully you should be able to browse a local version of gatsbyjs.com after this. Let's leave `yarn develop` running!

### 3. Rebuild `gatsby-design-tokens` when it changes, and use `gatsby-dev` to copy the changed-and-compiled package over to `www/node_modules`

- Open a new terminal window, go to the root of your local `gatsby` repo, and run `yarn run watch` to watch for changes in `packages`, and recompile the modified package.
  - To watch only certain packages, use `yarn run watch --scope=gatsby-design-tokens` or `--scope={gatsby,gatsby-cli}`.
- Last, in another new terminal window, go to `gatsby/www`, and run `gatsby-dev --packages=gatsby-design-tokens` to copy the latest local version of the `gatsby-design-tokens` package over to `www/node_modules`.
  - Alternatively pass an array to watch multiple packages, or use plain `gatsby-dev` to copy the latest versions of all local packages that are used in your project's `package.json`.

### 4. Finally!

Let's try if everything is running and watching and recompiling! Let's open `packages/gatsby-design-tokens/src/fonts.js`, and replace

```
const header = [Futura PT, ...system]
```

with

```
const header = system
```

and you should see `Futura PT` turning into `system-ui` within a few moments on your `http://localhost:8000`.