

# Consistency in Theming System

through

# Style Dictionary

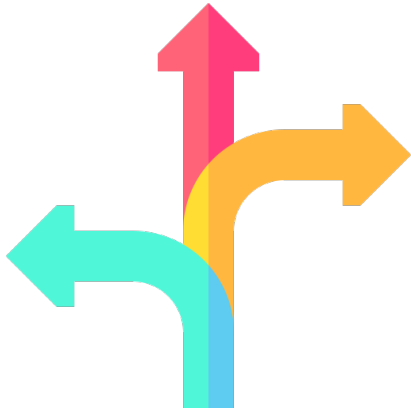


Apr 2025

Mohsen Madani

# Why?

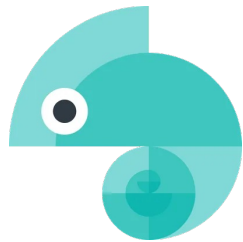
**Flexibility**



**Integrity**

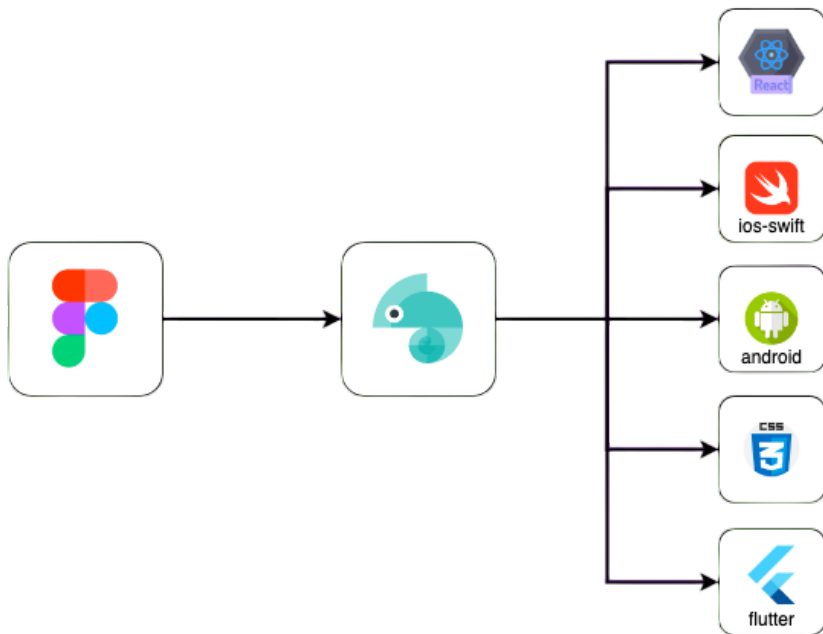
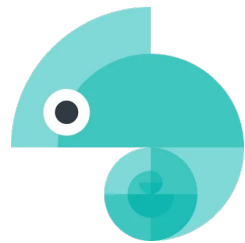


# Style Dictionary



- Export your Design Tokens to any platform - iOS, Android, **CSS**, **JS**, HTML, sketch files, style documentation, or anything you can think of. Forward-compatible with

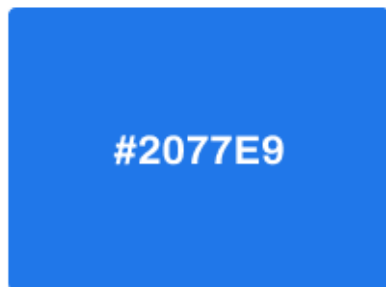
# Style Dictionary



# Design Token



Design Decision



"Our primary color is blue"

Design Token

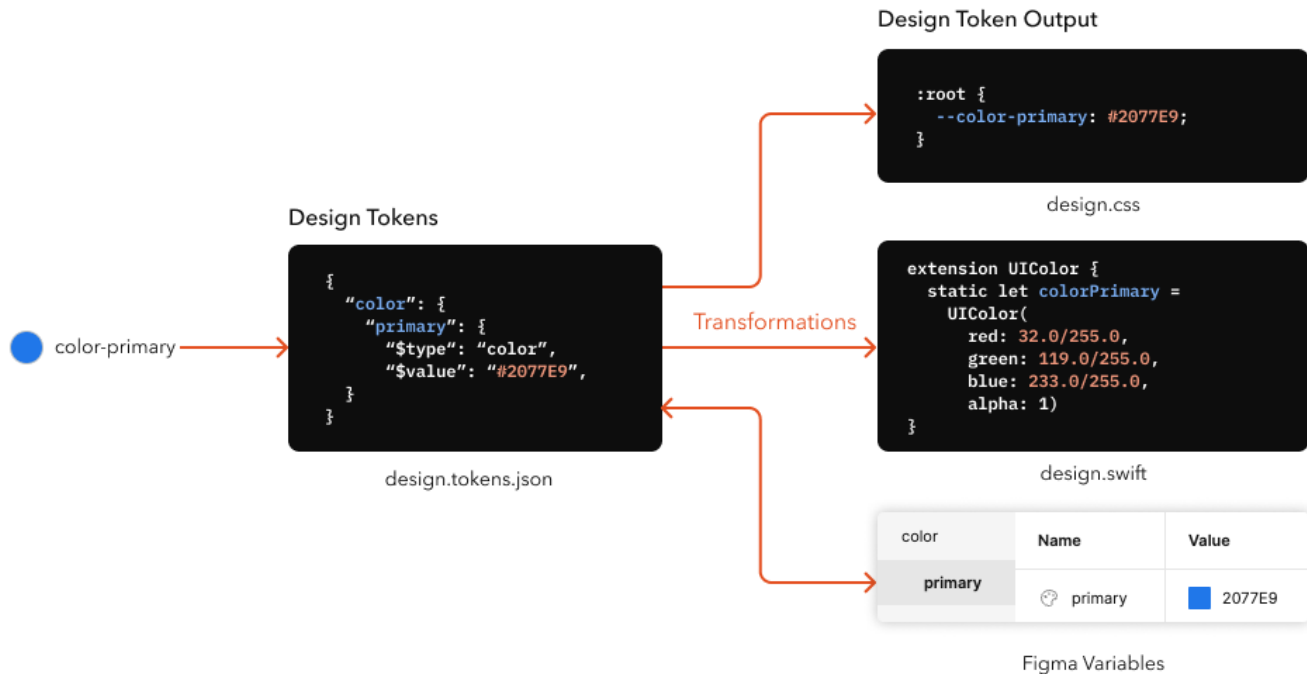


Design Token Code

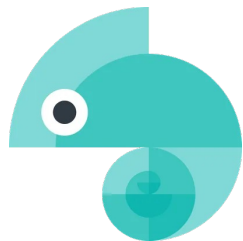
```
{  
  "color": {  
    "primary": {  
      "$type": "color",  
      "$value": "#2077E9",  
    }  
  }  
}
```

design.tokens.json

# Design Token



# Design Token



## Primitive

- color-white
- color-black
- color-grey-100
- color-grey-200
- color-grey-300
- color-grey-400
- color-grey-500
- color-grey-600
- color-grey-700
- color-grey-800

## Semantic

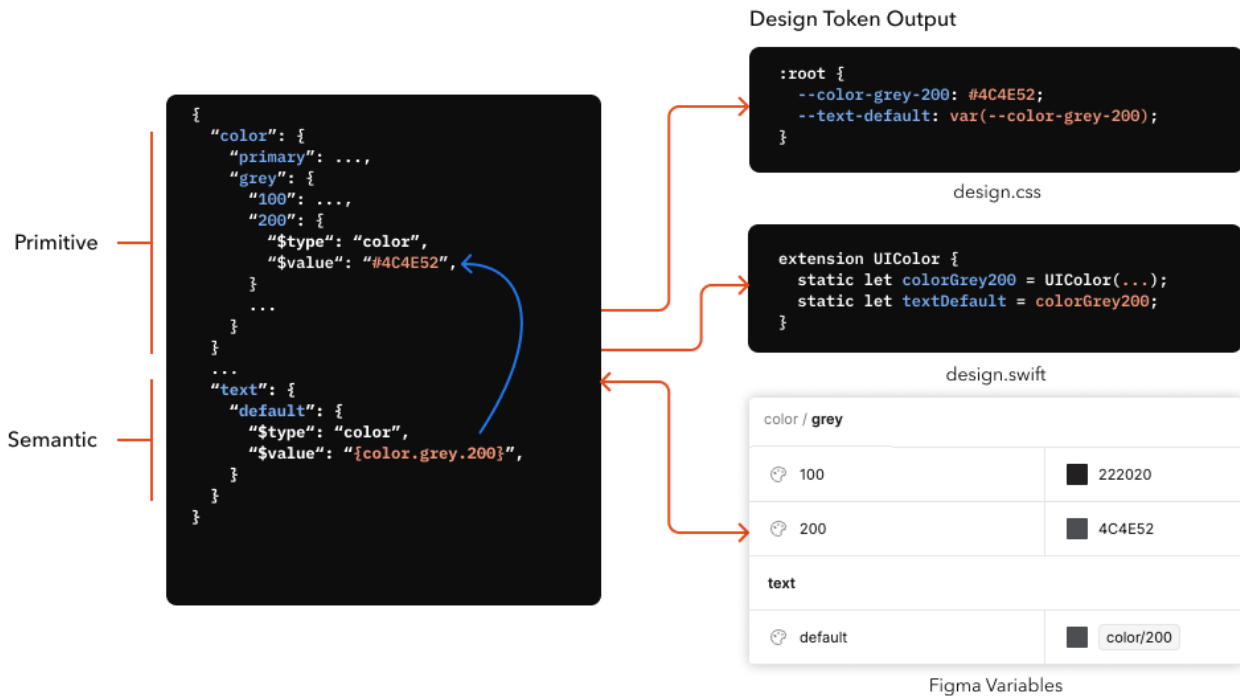
- text-default

## Web Page

**Organize and integrate your content, then easily reuse it**

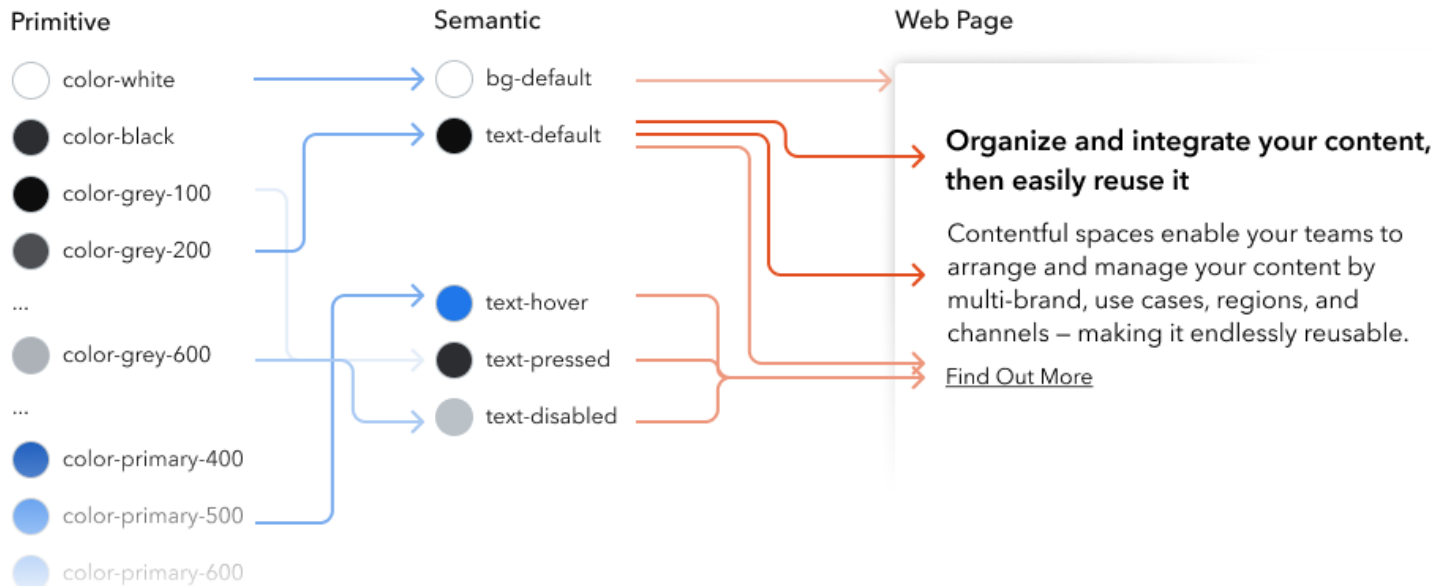
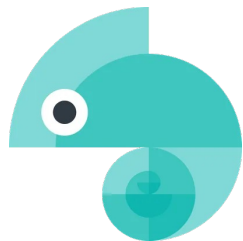
Contentful spaces enable your teams to arrange and manage your content by multi-brand, use cases, regions, and channels – making it endlessly reusable.

# Design Token





# Design Token





# Design Token



```
palette.json  JSON

{
  "palette": {
    "retail": {
      "primary": {
        "0": {
          "value": "#ffffff"
        },
        ...,
        "100": {
          "value": "#000000"
        }
      },
      "neutral": {...},
      "error": {...},
      "success": {...},
      "warning": {...},
      "informative": {...}
    }
  }
}
```

# Design Token



```
color.json  JSON

{
  "light": {
    "color": {
      "retail": {
        "primary": {
          "base": {
            "value": "{palette.retail.primary.50}"
          },
          "on": {
            "value": "{palette.retail.primary.0}"
          },
          "container": {
            "value": "{palette.retail.primary.5}"
          },
          "onContainer": {
            "value": "{palette.retail.primary.60}"
          }
        },
        "success": {...},
        "error": {...},
        "warning": {...},
        "informative": {...},
        "onSurface": {...},
        "surface": {...},
        ...
      },
      "op": {...}
    },
    "dark": {...}
  }
}
```

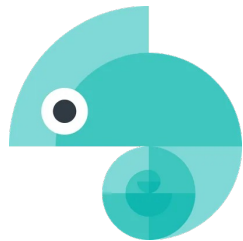
# Design Token



```
_variables.scss SCSS

$tokens: (
  "light": (
    "color": (
      "retail": (
        "primary": (
          "base": #e30613,
          "on": #fff,
          "container": #fce6e7,
          "onContainer": #cf0511
        ),
        ...
      ),
      "op": (...)
    ),
    "font": (
      "retail": (
        "weight": ("bold": 700, "medium": 500, "regular": 400, "light": 300)
      ),
    ),
    "typography": (
      "retail": (
        "display": ("large": ("font-size": 57px, "line-height": 88px), ...),
        ...
      )
    ),
  ),
  "dark": (...)
);
```

# Design Token

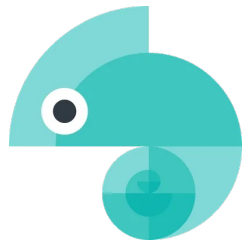


\_mixins.scss

SCSS

```
@mixin themify($themes: variables.$tokens) {  
  @each $themeName, $themeMap in $themes {  
    .theme-#{$themeName} & {  
      // Assign this theme's nested map globally so it's accessible to '@content'  
      $theme-map: $themeMap !global;  
  
      // Everything inside @content sees '$theme-map' for lookups  
      @content;  
  
      // Clean up the global variable after we're done  
      $theme-map: null !global;  
    }  
  }  
}
```

# Design Token



styles.scss

SCSS

```
* {  
  @include theme.themify {  
    // This block is generated once for each .theme-light & .theme-dark, etc.  
    color: theme.themed('color', 'retail', 'primary', 'base');  
    background-color: theme.themed('color', 'retail', 'primary', 'on');  
  }  
}
```

# Design Token

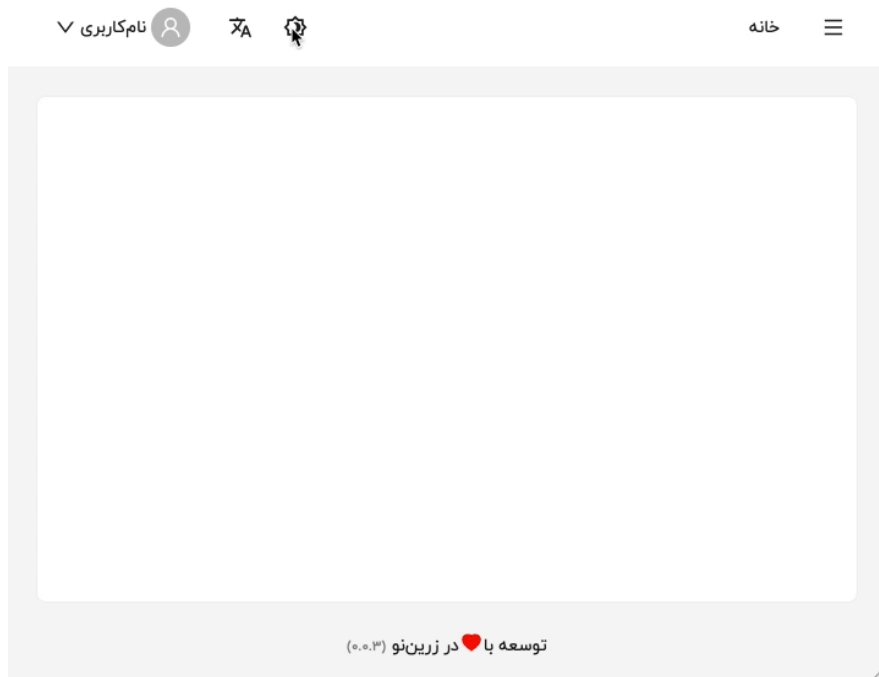
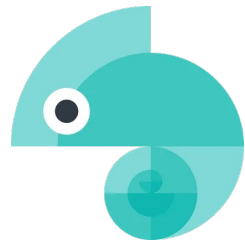


```
variables.ts                                TypeScript

export const TOKENS = {
  light: {
    color: {
      retail: {
        primary: {
          base: '#e30613',
          on: '#ffffff',
          container: '#fce6e7',
          onContainer: '#cf0511',
        },
        success: {...},
        error: {...},
        warning: {...},
        informative: {...},
        onSurface: {...},
        surface: {...},
        outline: {...},
        onBackground: {...},
      },
      op: {...},
    },
    font: {...},
    typography: {...},
  },
  dark: {...},
} as const;
```



# Tada!



# Q&A

Ask me question if you have any

# The END

Mohsen Madani



# References

- Style Dictionary ([link](#))
- Contentful ([link](#))