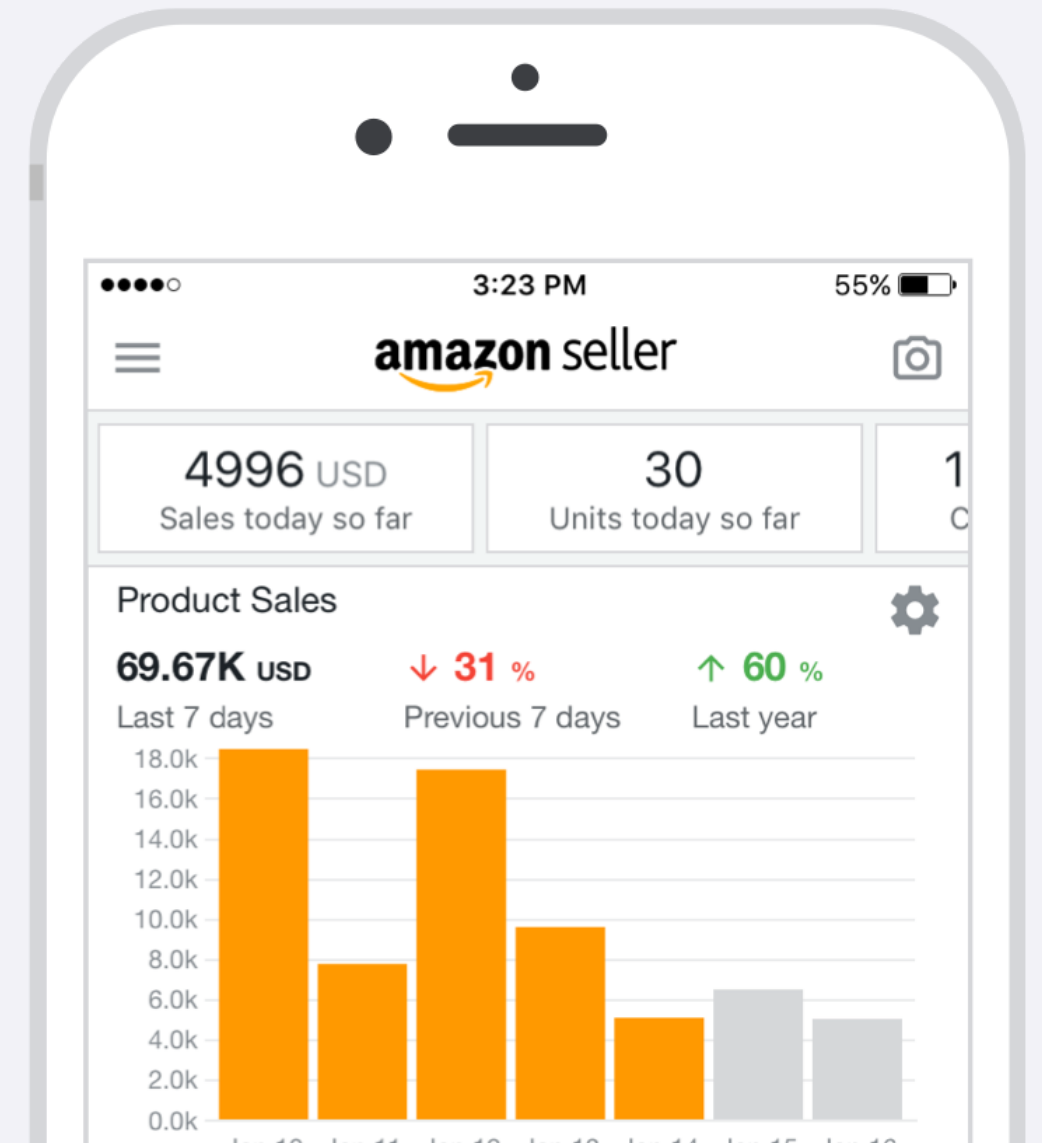
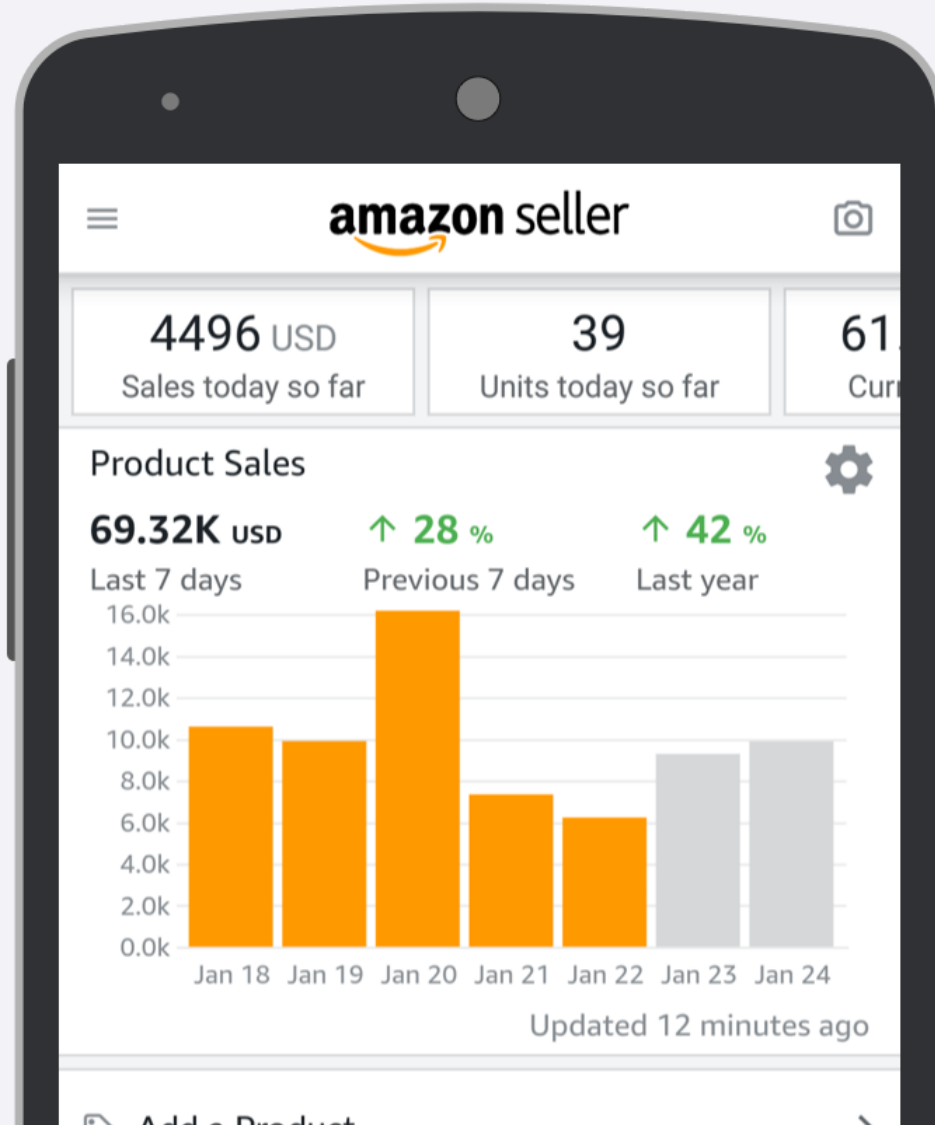




Building Consistent Cross-Platform Interfaces



<https://bit.ly/ato2018-cross-platform>
<https://bit.ly/ato2018-cross-platform-pdf>
@dbanksdesign #AllThingsOpen





Consistent experiences help users build trust with the organization. Each interaction is part of the overall user experience with a company. If the user experience isn't consistent **across channels**, users will question the organization's credibility.

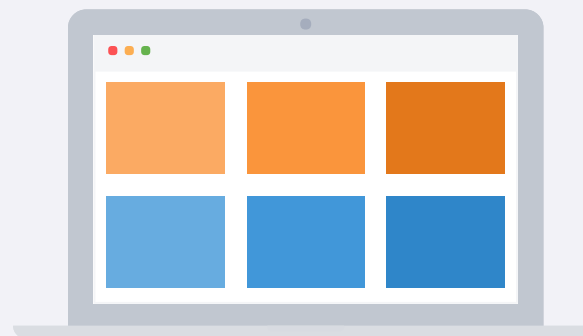
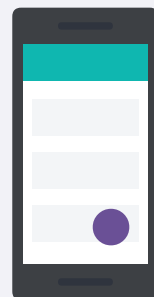
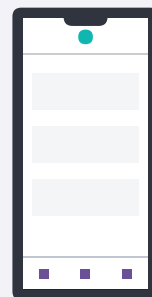
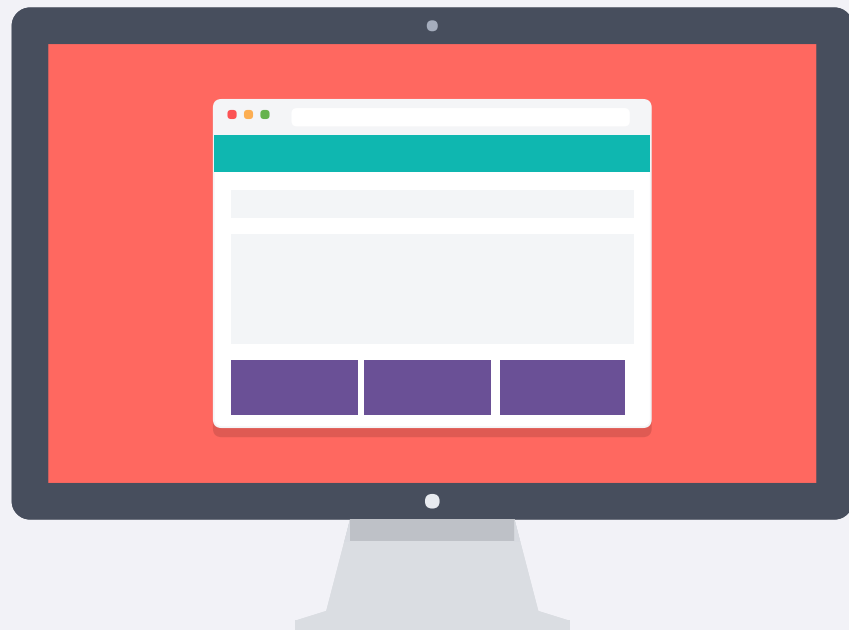
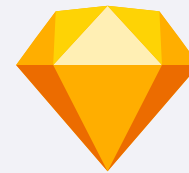
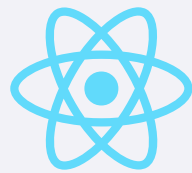
- Jakob Nielsen

Cross Channel Consistency

Consistent \neq Exact

Be a good platform citizen







Design Tokens



Cr

Color



Tp

Typography



Ic

Iconography



Mn

Motion



Dm

Dimension



Ev

Elevation

01. Canonical

primary brand color

SCSS

```
$color-brand-primary-base: #0fb7b0;
```

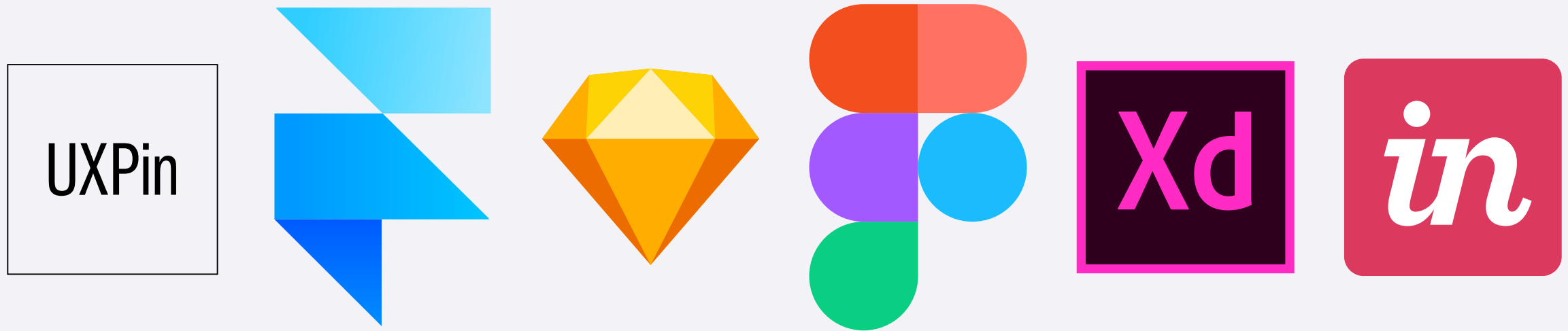
iOS/Objective-C

```
UIColor *brandPrimaryBase = [UIColor colorWithRed:0.06f green:0.72f blue:0.69f alpha:1.00f];
```

Android Resource XML

```
<color name="color_brand_primary_base">#ff0fb7b0</color>
```

02. Agnostic



03. **Structured**

Kingdom

Phylum

Class

Order

Family



04. Relational



color.core.teal.100

#0fb7b0



color.brand.primary.base

color.core.teal.100



color.background.button

color.brand.primary.base



color.font.link

color.brand.primary.base



color.core.orange.100

#fa953c



color.brand.primary.base

color.core.orange.100



color.background.button

color.brand.primary.base



color.font.link

color.brand.primary.base



color.core.teal.100

#0fb7b0



color.core.purple.100

#6a5096



color.brand.primary.base

color.core.teal.100



color.brand.secondary.base

color.core.purple.100



color.background.button

color.brand.primary.base



color.font.link

color.brand.secondary.base

Core color palette

grey 140	#191926
grey 120	#252535
grey 100	#2e2e46
grey 80	#48485e
grey 60	#81818e
grey 40	#c2c2cc
grey 20	#dadae6
grey 10	#f2f2f7

Text color palette

base	grey-120
secondary	grey-80
tertiary	grey-60

Specific uses

input	text-base
heading-1	text-secondary
nav-item	text-secondary
disabled-input	text-tertiary

Consistent

Resilient

Flexible

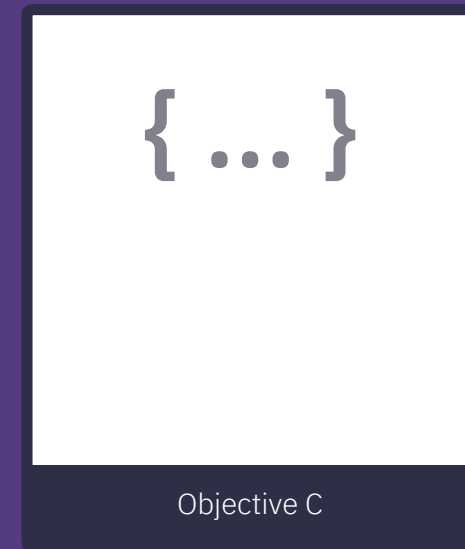
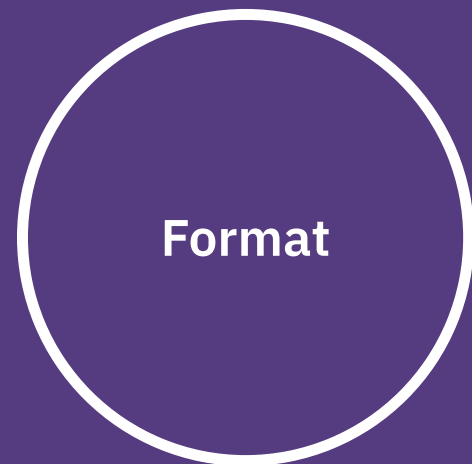
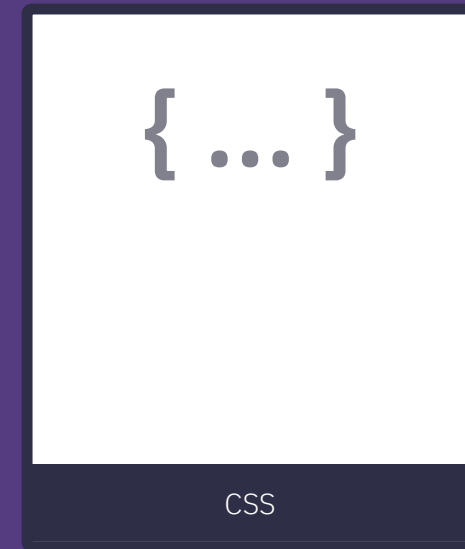
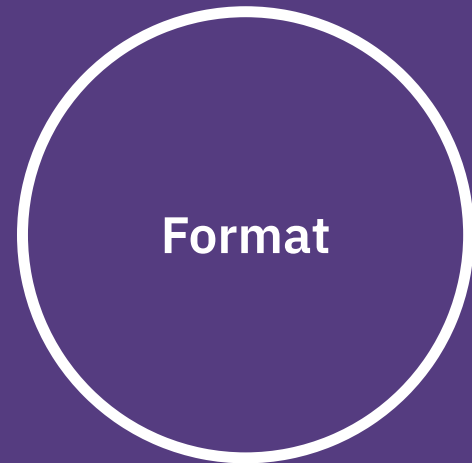


JSON

```
{
  "color": {
    "core": {
      "grey": {
        "20": "#f3f1f1",
        "40": "#888888",
        "60": "#4d4d4d",
        "80": "#323232",
        "100": "#262626"
      }
    }
  }
  "text": {
    "primary": "{color.core.grey.100}",
    "secondary": "{color.core.grey.80}",
    "tertiary": "{color.core.grey.60}"
  }
}
```

YAML

```
color:
  core:
    grey:
      20: &color-core-grey-20 "#f3f1f1"
      40: &color-core-grey-40 "#888888"
      60: &color-core-grey-60 "#4d4d4d"
      80: &color-core-grey-80 "#323232"
      100: &color-core-grey-100 "#262626"
  text:
    primary: *color-core-grey-100
    secondary: *color-core-grey-80
    tertiary: *color-core-grey-60
```



Custom Build Script

Eight Shapes

YAML + custom build script

FirefoxUX Design Tokens

Nested JSON objects + custom build script

✓ JSON, JSON5, or YAML based token files

✓ CLI as well as node module based

✓ Transforms and formats with customization

✓ Has a great community

```
# aliases.yml
aliases:
  primary_color:
    value: "rgb(125, 75, 32)"

# buttons.yml
props:
  button_background:
    value: "{!primary_color}"
imports:
  - ./aliases.yml
global:
  type: color
  category: buttons
```

```
const theo = require('theo');

theo
  .convert({
    transform: {
      type: 'web',
      file: 'buttons.yml'
    },
    format: {
      type: 'scss'
    }
  })
  .then(scss => {
    // $button-background: rgb(0, 112, 210);
  });
```



Style Dictionary

amzn.github.io/style-dictionary

Structure

1. MyStyleDictionary
2. |— assets
3. | |— fonts
4. | |└─ images
5. |— properties
6. | |└─ asset
7. | | |└─ font.json
8. | | |└─ color
9. | | | |— background.json
10. | | | |— border.json
11. | | | |— brand.json
12. | | | |— core.json
13. | | | |└─ font.json
14. | | |└─ size
15. | | | |— border.json
16. | | | |— font.json
17. | | | |— icon.json
18. | | | |└─ padding.json
19. |— config.json

config.json

```
1. {
2.   "source": ["properties/**/*.json"],
3.   "platforms": {
4.     "scss": {
5.       "transforms": ["attribute/cti", "name/cti/kebab", "color/hex"],
6.       "buildPath": "dist/web/",
7.       "files": [{
8.         "destination": "_variables.scss",
9.         "format": "scss/variables"
10.      }]
11.    },
12.    "android": {
13.      "transformGroup": "android",
14.      "buildPath": "dist/android/",
15.      "files": [{
16.        "destination": "style_dictionary_colors.xml",
17.        "format": "android/colors"
18.      }],
19.      "actions": ["copyImages"]
20.    }
21.  }
22. }
```

Design Token Files

```
1. {
2.   "size": {
3.     "font": {
4.       "small" : { "value": 0.75 },
5.       "medium": { "value": 1 },
6.       "large" : { "value": 1.25 },
7.       "xl"     : { "value": 1.75 },
8.       "xxl"    : { "value": 2.5 },
9.       "base"   : {
10.        "value": "{size.font.medium.value}",
11.        "comment": "All about that base"
12.      },
13.       "heading": {
14.         "1": { "value": "{size.font.xx1.value}" },
15.         "2": { "value": "{size.font.xl.value}" }
16.       }
17.     }
18.   }
19. }
```

Building the Style Dictionary

```
1. const StyleDictionary = require('style-dictionary');
2.
3. // If you want to add custom transforms or formats
4. StyleDictionary.registerTransform({
5.   // ...
6. });
7.
8. // Same as running style-dictionary build --config config.json
9. StyleDictionary.extend('config.json').buildAllPlatforms()
10.
11. // You can also pass in an object rather than a JSON file
12. StyleDictionary.extend({
13.   // ...
14. }).buildAllPlatforms()
```

Output Web: SCSS

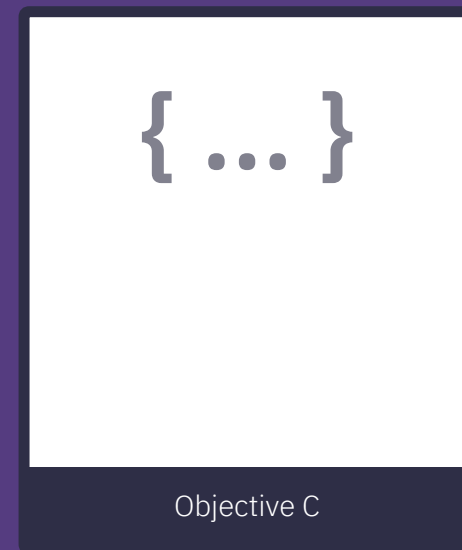
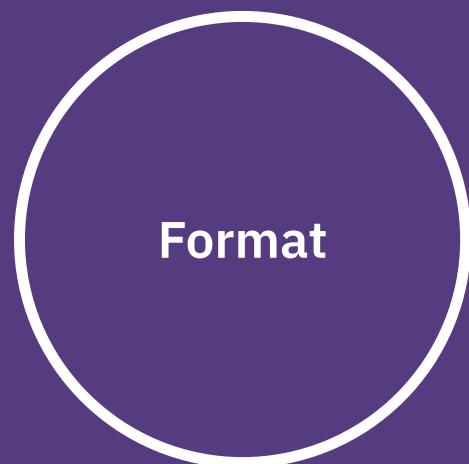
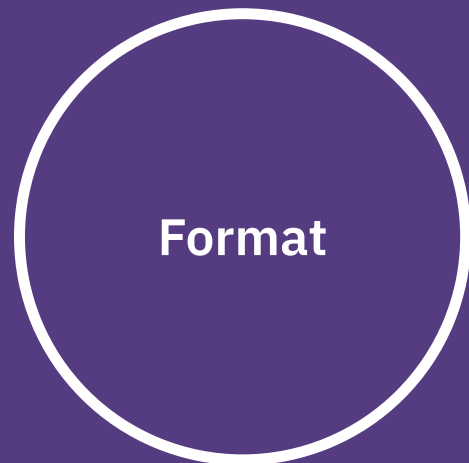
```
1. // variables.scss
2.
3. $size-font-small: 0.75rem;
4. $size-font-base: 1rem;
5. $size-font-large: 1.25rem;
6. $size-font-xl: 1.75rem;
7. $size-font-xxl: 2.5rem;
8. $size-font-base: 1rem // All about that base
9. $size-font-heading-1: 2.5rem;
10. $size-font-heading-2: 1.75rem;
```

Output iOS: Static Constants

```
1. // StyleDictionarySize.m
2.
3. float const SizeFontSmall = 12.0f;
4. float const SizeFontMedium = 16.0f;
5. float const SizeFontLarge = 20.0f;
6. float const SizeFontXl = 28.0f;
7. float const SizeFontXxl = 40.0f;
8. float const SizeFontBase = 16.0f; // All about that base
9. float const SizeFontHeading1 = 40.0f;
10. float const SizeFontHeading2 = 28.0f;
```

Output Android: Resource XML

```
1. // dims.xml
2.
3. <dimen name="size_font_small">12.00sp</dimen>
4. <dimen name="size_font_base">16.00sp</dimen>
5. <dimen name="size_font_large">20.00sp</dimen>
6. <dimen name="size_font_xl">28.00sp</dimen>
7. <dimen name="size_font_xx1">40.00sp</dimen>
8. <dimen name="size_font_base">16.00sp</dimen> // All about that base
9. <dimen name="size_font_heading_1">40.00sp</dimen>
10. <dimen name="size_font_heading_2">28.00sp</dimen>
```



```
const StyleDictionary = require('style-dictionary');
const Color = require('tinycolor2');
const _ = require('lodash');

// #06a39c => rgb(6, 163, 156)
StyleDictionary.registerTransform({
  name: 'color/rgbString',
  type: 'value',
  matcher: (prop) => prop.attributes.category === 'color',
  transformer: (prop) => Color(prop.value).toRgbString()
});

// size_font_base
StyleDictionary.registerTransform({
  name: 'name/snake',
  type: 'name',
  transformer: (prop) => _.snakeCase(prop.path.join(' '))
});
```


02. Formats



```
const StyleDictionary = require('style-dictionary');
const Handlebars = require('handlebars');
const fs = require('fs');
```

```
StyleDictionary.registerFormat({
  name: 'scss/variables',
  formatter: (dictionary) => (
    dictionary.allProperties.map((prop) => (
      `$$${prop.name}: ${prop.value}`
    ))
  )
});
```

```
StyleDictionary.registerFormat({
  name: 'scss/variables',
  formatter: Handlebars.compile(
    fs.readFileSync('./myTemplate.hbs')
  )
});
```

03. Actions



```
const StyleDictionary = require('style-dictionary');
const spawn = require('child_process').spawn;

StyleDictionary.registerAction({
  name: 'generate_pngs',
  do: (dictionary, config) => {
    Object.keys(dictionary.properties.asset.svg)
      .forEach((key) => {
        let prop = dictionary.properties.asset.svg[key];
        let inkscape = spawn('inkscape', [
          prop.value,
          `--export-png=./dist/images/${key}.png`,
          `--export-width=${prop.attributes.size}`,
          `--export-background=#ffffff`
        ]);
      });
  });
});
```

Demo Time!

light
#40c7c2

base
#0fb7b0

dark
#06a39c

light
#8f6ccc

base
#6a5096

dark
#553c80



Button

Button

Heading 1

Heading 2

Heading 3

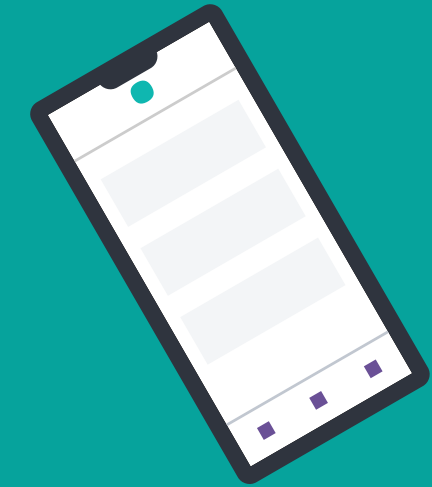
Resources

 [Awesome Design Tokens](#)

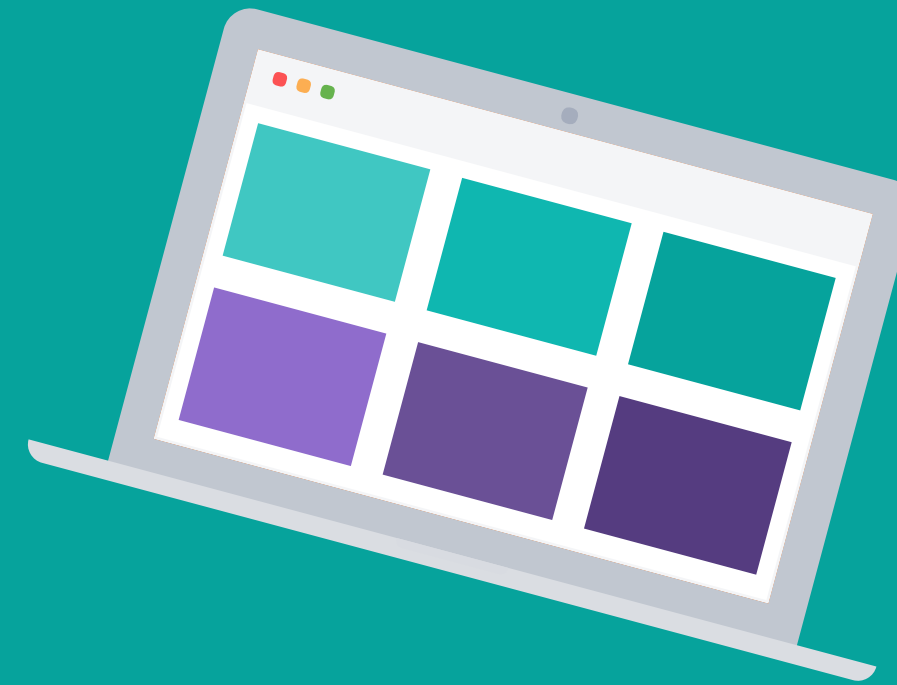
 [Design Systems Handbook](#)

M [How to manage your Design Tokens with Style Dictionary](#)

 [design.systems](#) [#tool-style-dictionary](#) [#tool-theo](#) [#topic-design-tokens](#)



Thank You!



Style Dictionary

 amzn.github.io/style-dictionary

 github.com/amzn/style-dictionary

 style-dictionary@amazon.com

My Info

 dbanksdesign.com

 [@dbanksdesign](https://github.com/dbanksdesign)

 [@dbanksdesign](https://twitter.com/dbanksdesign)

Slides

<https://bit.ly/ato2018-cross-platform>

<https://bit.ly/ato2018-cross-platform-pdf>