

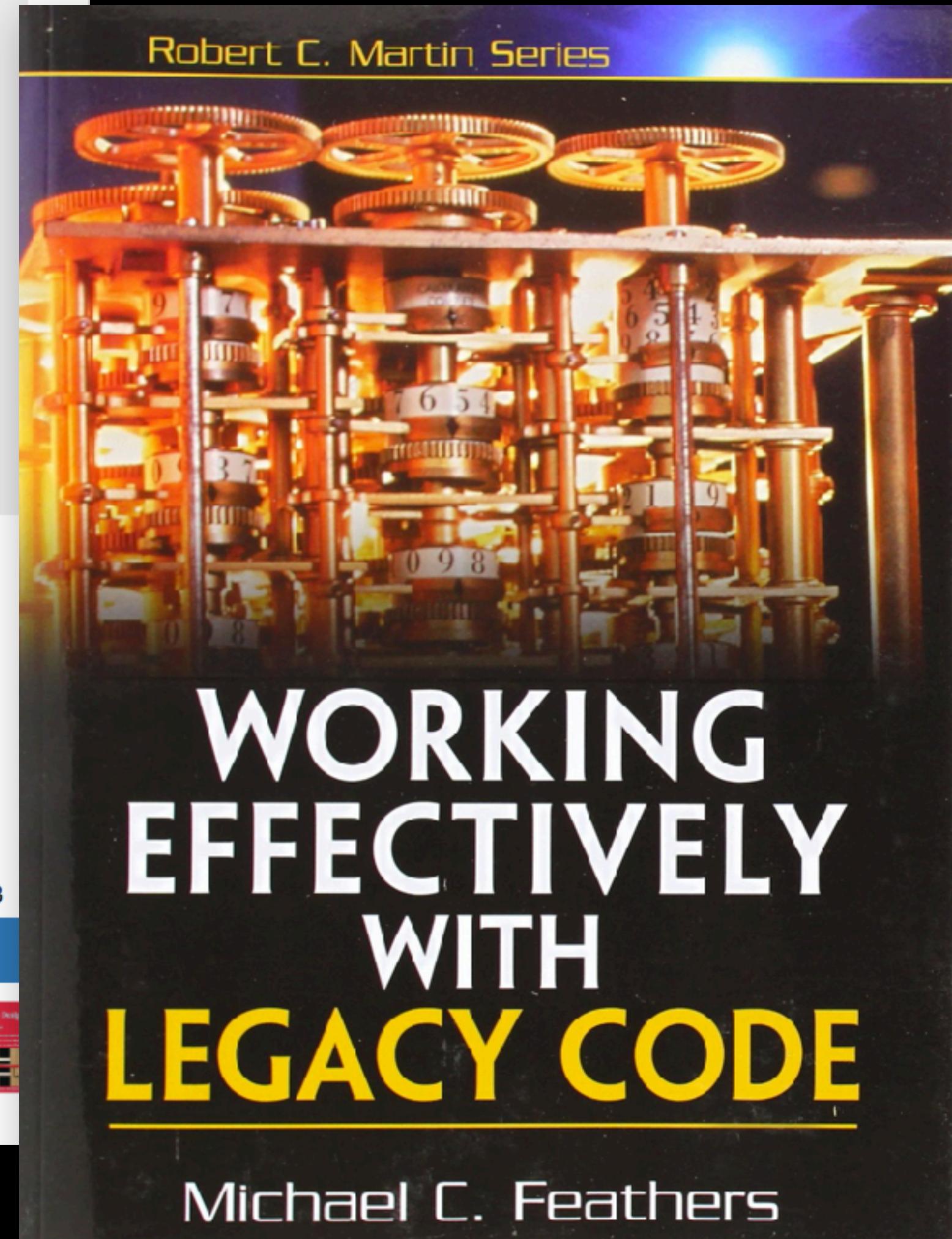
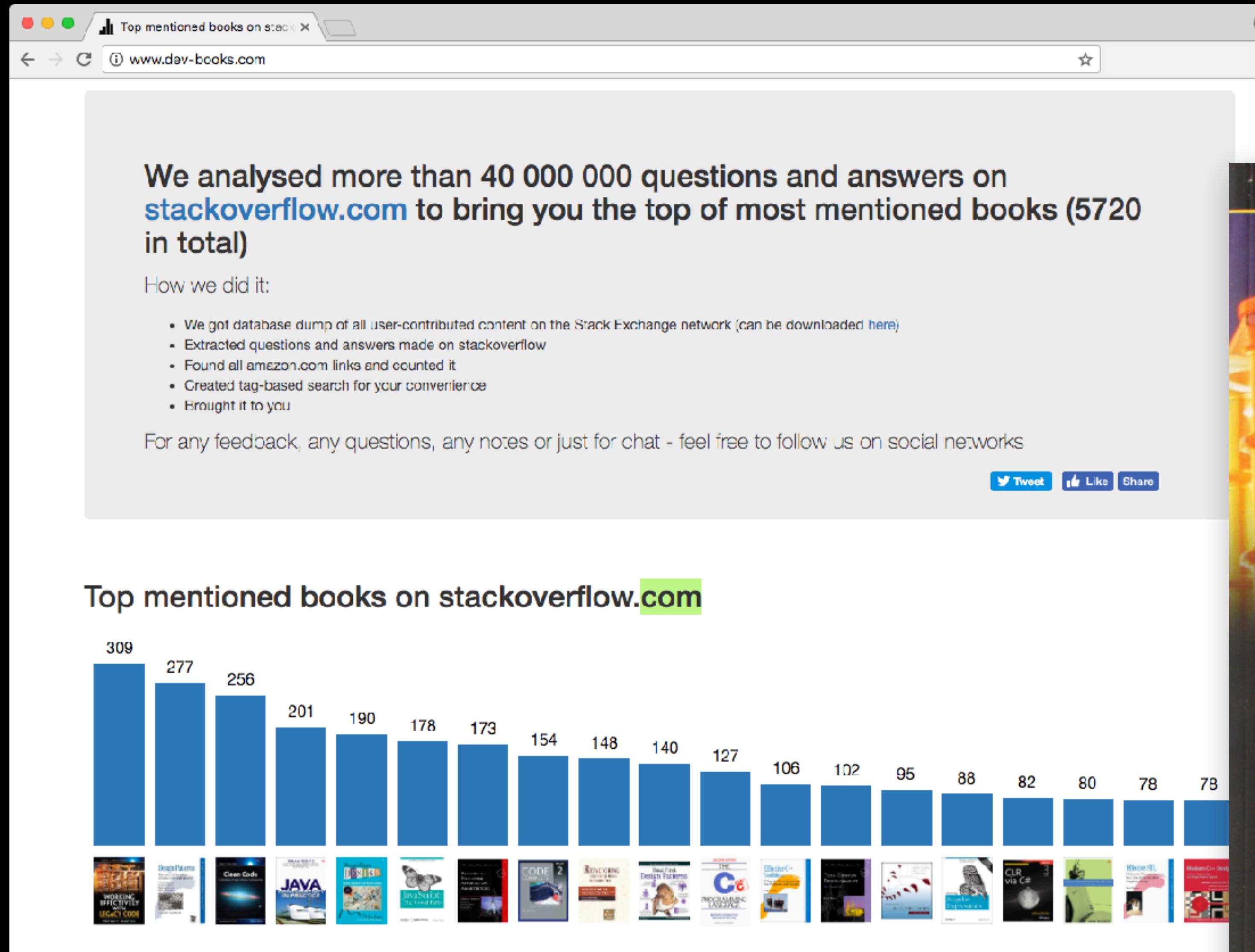
# **BUILD FOR DESTRUCTION**

*zehfernades* • @zehf • #frontinsampa2017



WORK  
& CO







Atenção : Registros de Sucesso serão automaticamente removidos em 30 (trinta) dias em relação a sua data de criação.

**2000237** Walmart Processado com erro Processado às 17:34 do dia 07/06/2016

**Ações**

- Buscar pelo Código
- Exportar para excel
- Remover Log

ORIGIN: cnova (293)

STATUS: sucesso (290), erro (3)

Buscar pelo Código: sucesso

Start: 21 Jul 2016 End: 21 Sep 2036 ACTIVE

Indexado: 1421

Total Indexado (1421)  
98,89% dos produtos

Ultima Atualização : 8/15/16

Walmart

Salvar Voltar Buscar Excluir Precisa de ajuda ? Limpar filtros de busca Voltar para Admin

Melhoramos a importação e exportação do Relatório de Skus! Clique aqui para ser direcionado para uma opção de Relatório de Skus mais completo e organizado.

Primeira Anterior 1 2 Próxima Última

Pesquisar categoria 1 de 1 🔍

### Profile System

Clients [+](#) [≡](#)

Addresses [+](#) [≡](#)

Buscar por nome do produto, categoria, marca, descrição... [🔍](#)

Desejo me autenticar Status: Ativa

Buscar por Nome, Descrição, UTM ou Política Comercial

Vendas > v1161401bdev-01

Taxa - frete

Archive

CARREGANDO...

Total: 110457 Per Page: 15

Search by Name, Description, UTM or Trade Policy

New Carrier Show:  Not Archived  Archived

FLUXO

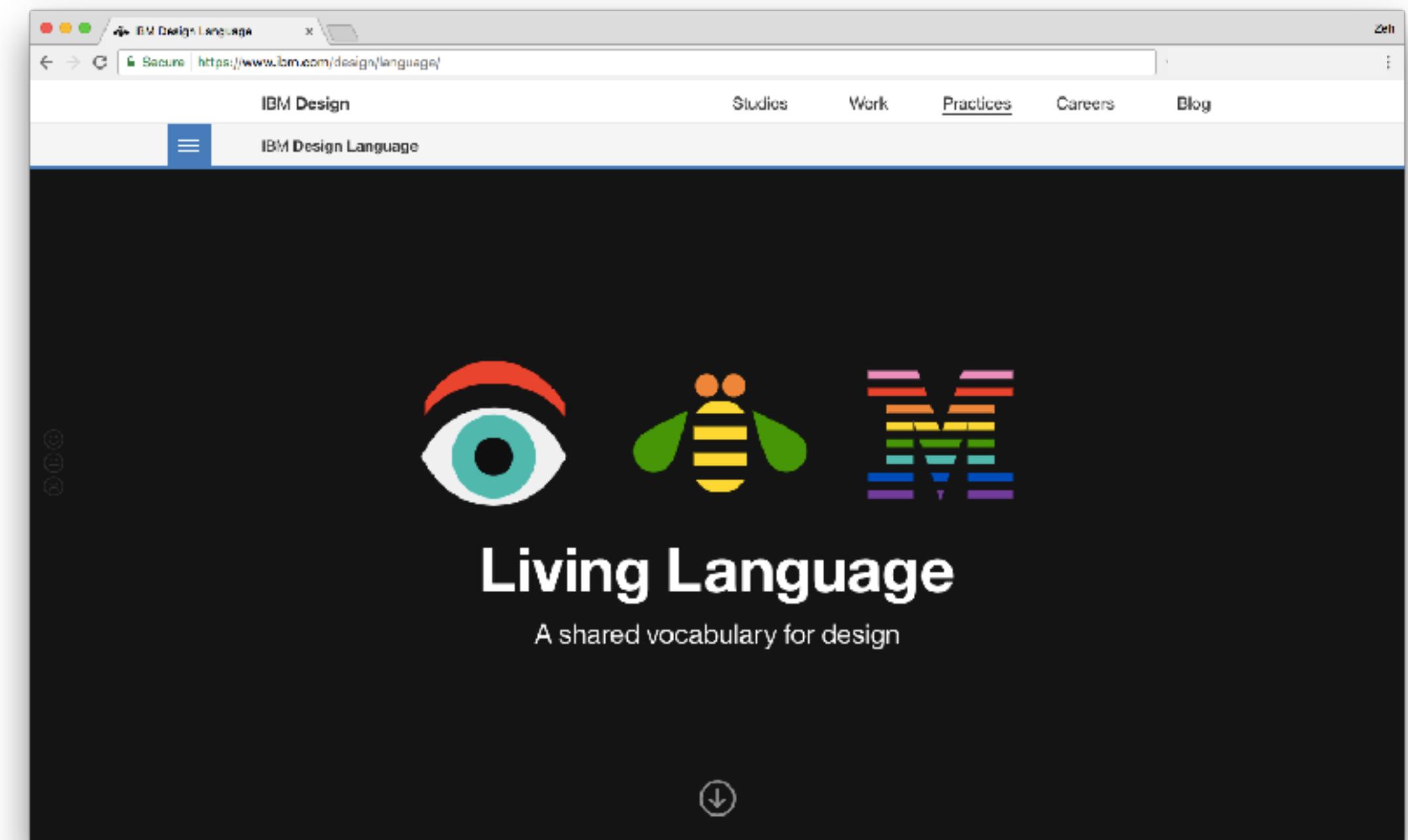
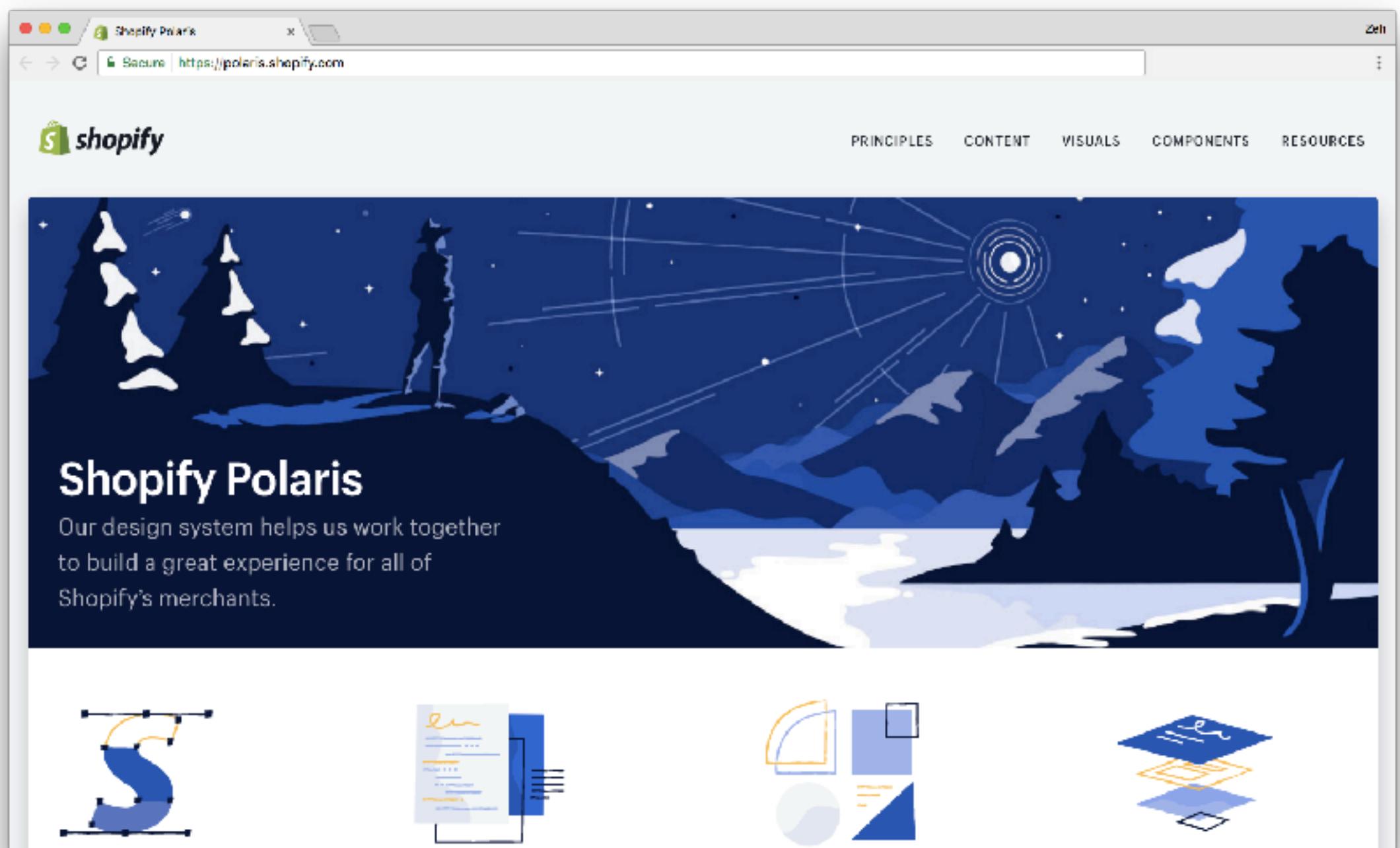
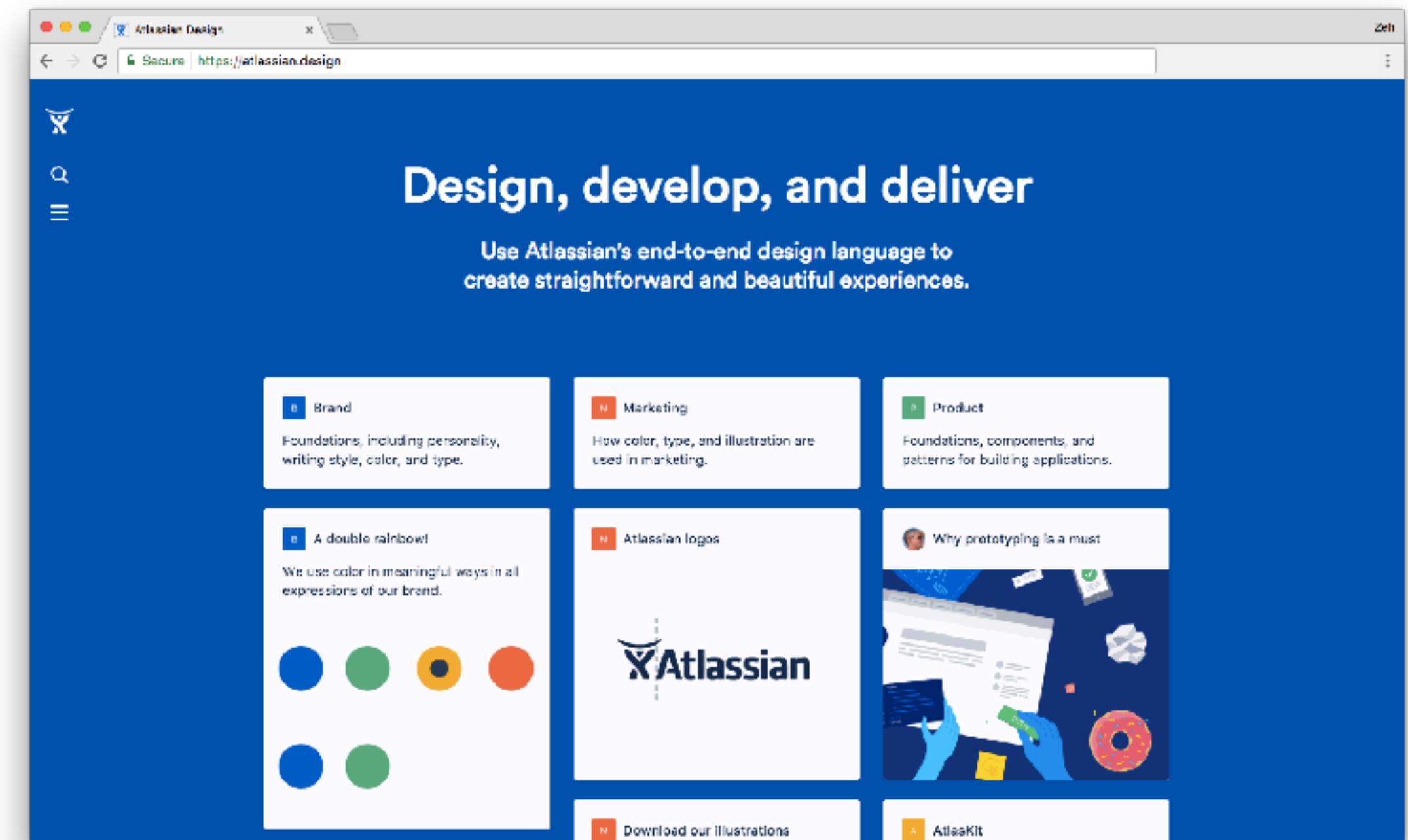
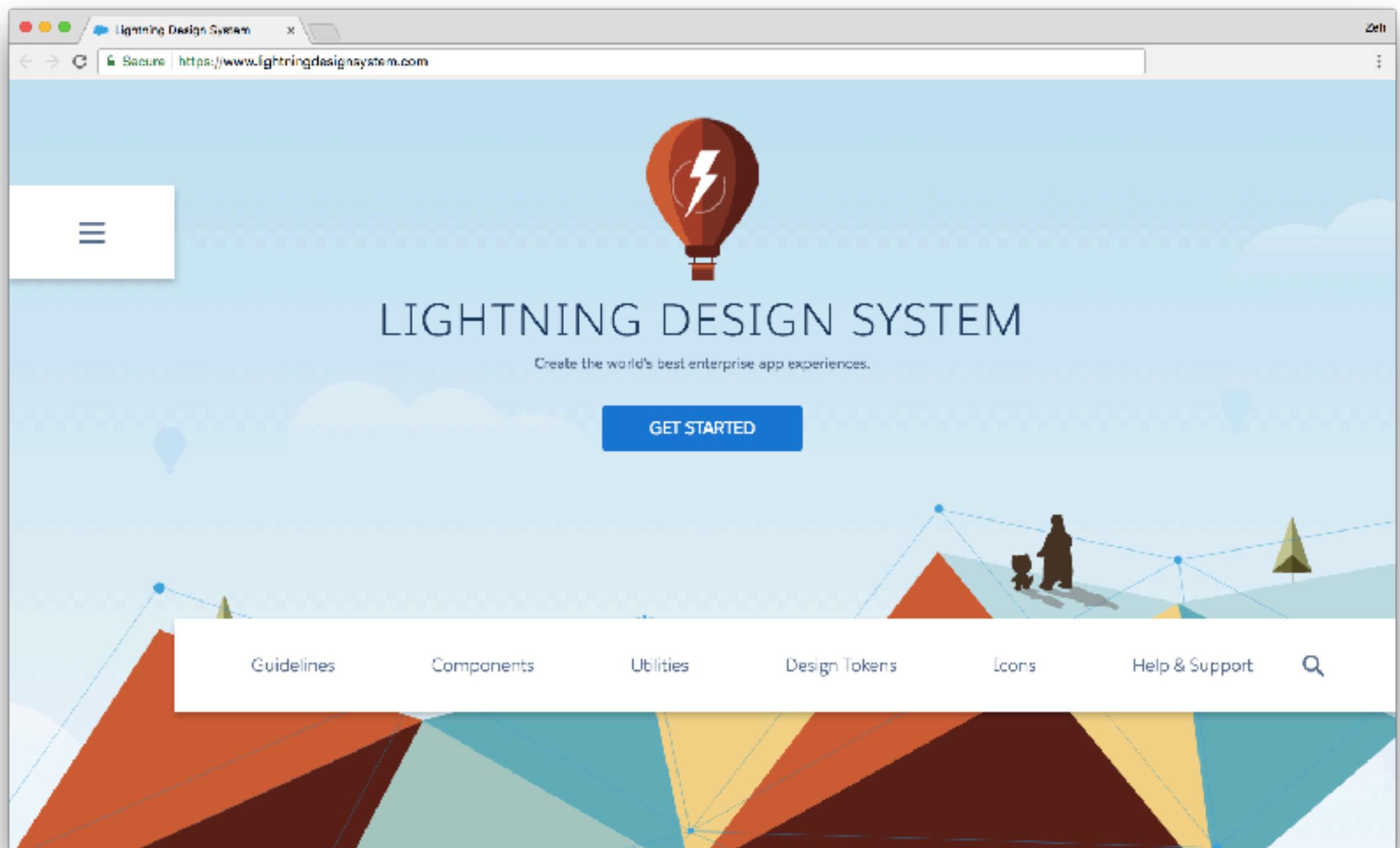
```

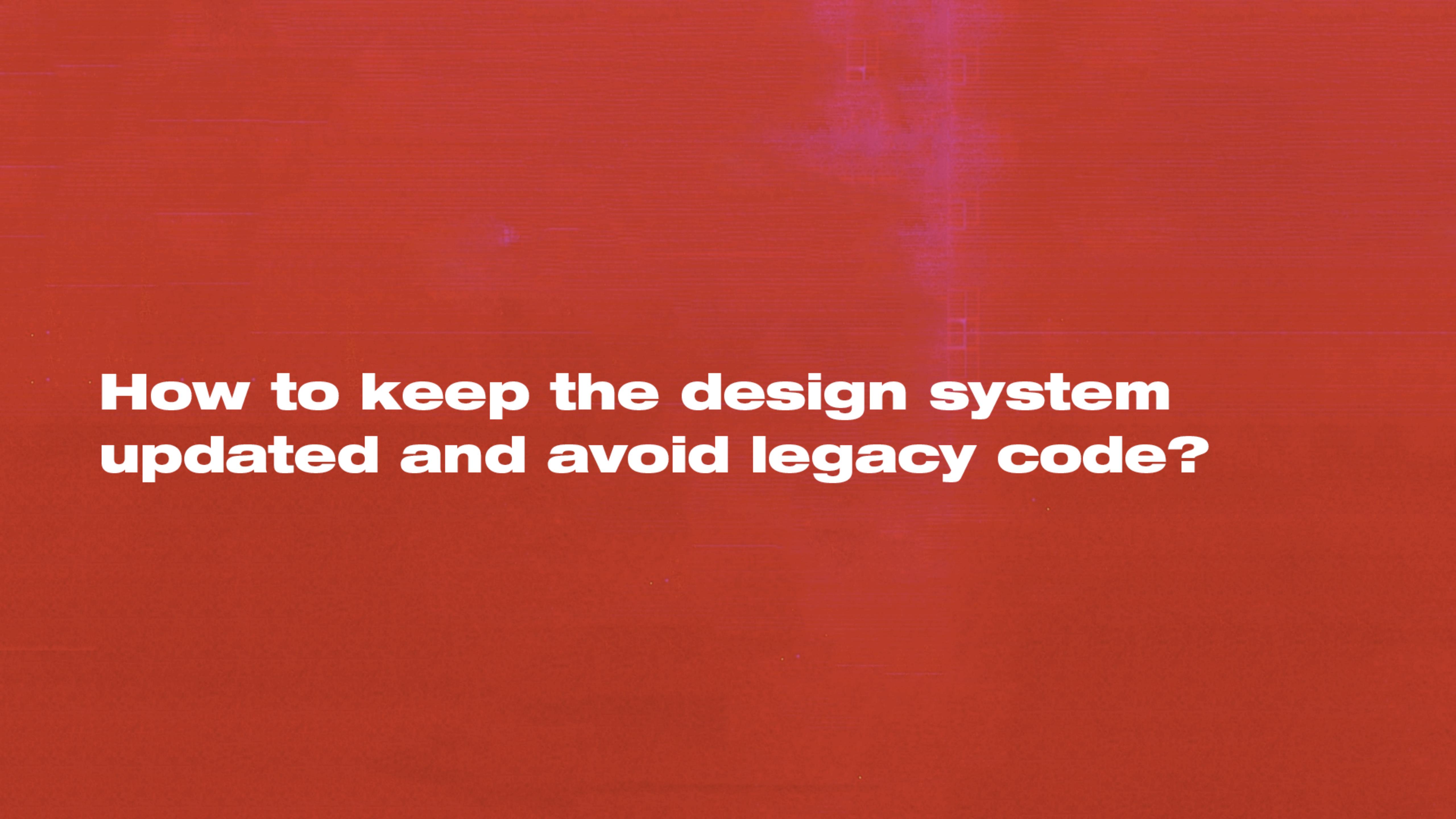
graph LR
    A[14:31 Aguardando confirmação do seller  
25/02/16] -- "Pedido aceito" --> B[14:31 Pagamento pendente  
25/02/16]
    B -- "Aprovar pagamento  
25/02/16" --> C[14:31 Pagamento aprovado  
25/02/16]
    C -- "Autorizar despacho  
25/02/16" --> D[14:31 Carência para cancelamento  
25/02/16]
    D -- "Fim da carência" --> E[15:01 Pronto para manuseio  
25/02/16]
    E -- "Iniciar manuseio  
09/06/16" --> F[11:05 Preparando entrega  
09/06/16]
    F -- "Todos os itens faturados" --> G[11:08 Faturado  
09/06/16]
    G -- "Negado" --> H[Solicitar cancelamento]
    H -- "Solicitar cancelamento" --> I[Cancelamento solicitado]
    I -- "Aceito" --> J[Iniciando cancelamento]
    J -- "Iniciando cancelamento" --> K[Cancelado]
    
```

EXIBIR INTERAÇÕES

Website







# **How to keep the design system updated and avoid legacy code?**



"A classic and probably a work of genius" —JANE JACOBS, author of *The Death and Life of Great American Cities*

# HOW BUILDINGS LEARN

What happens after they're built



New Orleans, 1857

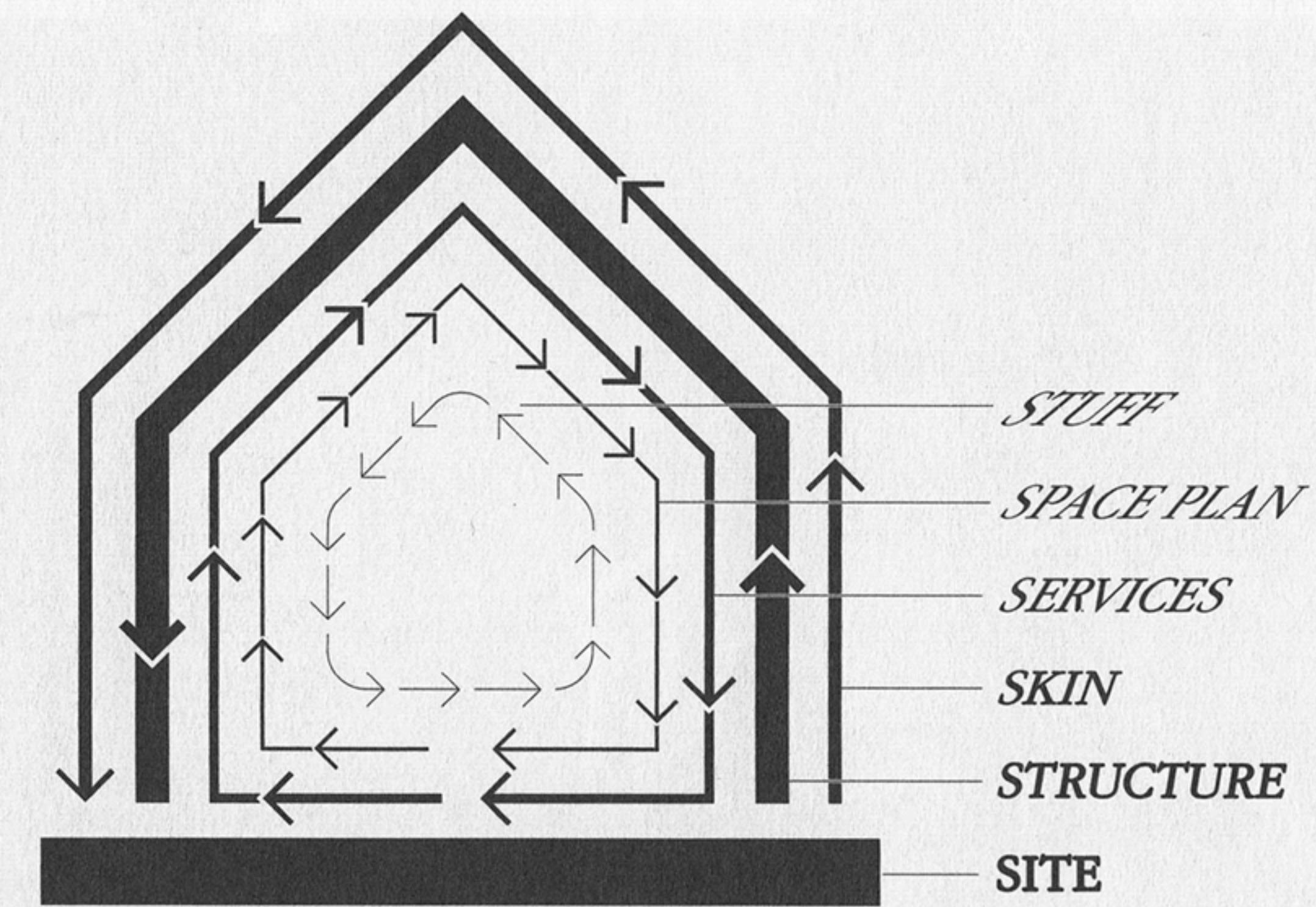


The same two buildings, 1993

STEWART BRAND

creator of THE WHOLE EARTH CATALOG





**“embrace time as a key factor  
in the development of the system”**

-- *Stewart Brand*

**BUILD FOR DESTRUCTION**

**BUILD FOR** Refactor

styleguide

styleguide.vtex.com

VTEX  
STYLEGUIDE

Design Principles

How-to

Instalação

Grid System

Elements

Button

Card

Heading

Icon

Input

Label

Loader

Select Box

Simple List

Table

Componentes

# PRINCÍPIOS DE DESIGN

## COMO FAZEMOS & POR QUE FAZEMOS

# Sólido

Pense modular.  
Seja extensível.  
Construa escalável.

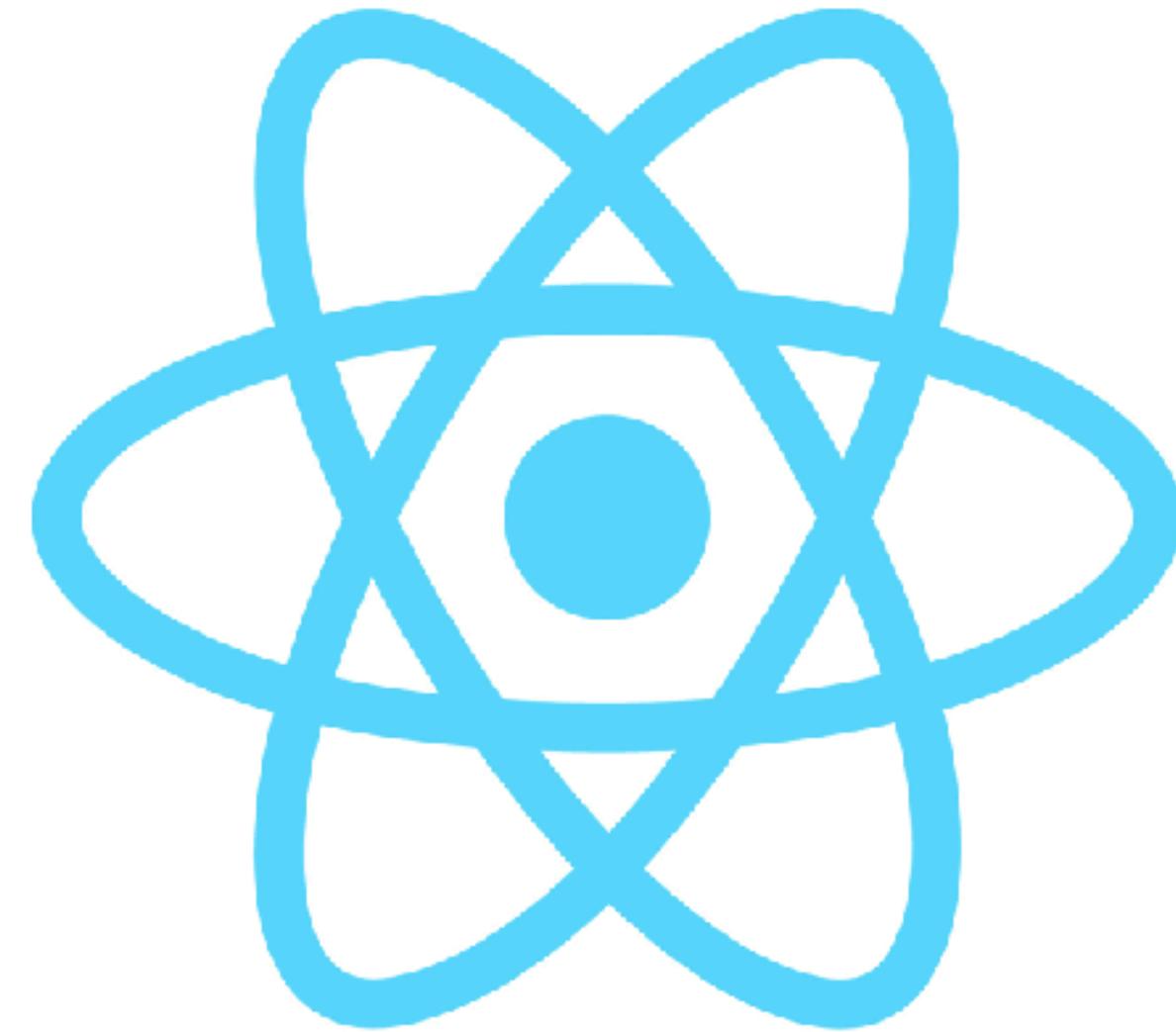
Crie sem limitações, pensando sempre em  
produtos escaláveis que cresçam com o  
ecossistema.

The page is a screenshot of a web browser showing a style guide for VTEX. The header includes the VTEX logo and the URL 'styleguide.vtex.com'. The main content area has a sidebar on the left listing various design components like 'Design Principles' and 'Elements'. The main section features a large title 'PRINCÍPIOS DE DESIGN' and a subtitle 'COMO FAZEMOS & POR QUE FAZEMOS'. Below this, a large heading 'Sólido' is followed by three bolded text blocks: 'Pense modular.', 'Seja extensível.', and 'Construa escalável.'. A final paragraph encourages creating products that grow with the ecosystem. To the right of the text is a graphic of concentric circles and a solid blue circle at the bottom left.

- *Separation of concerns*
- *Single source of truth*
- *Don't Repeat yourself*
- *Single Responsibility Principle*



*Separation of concerns*

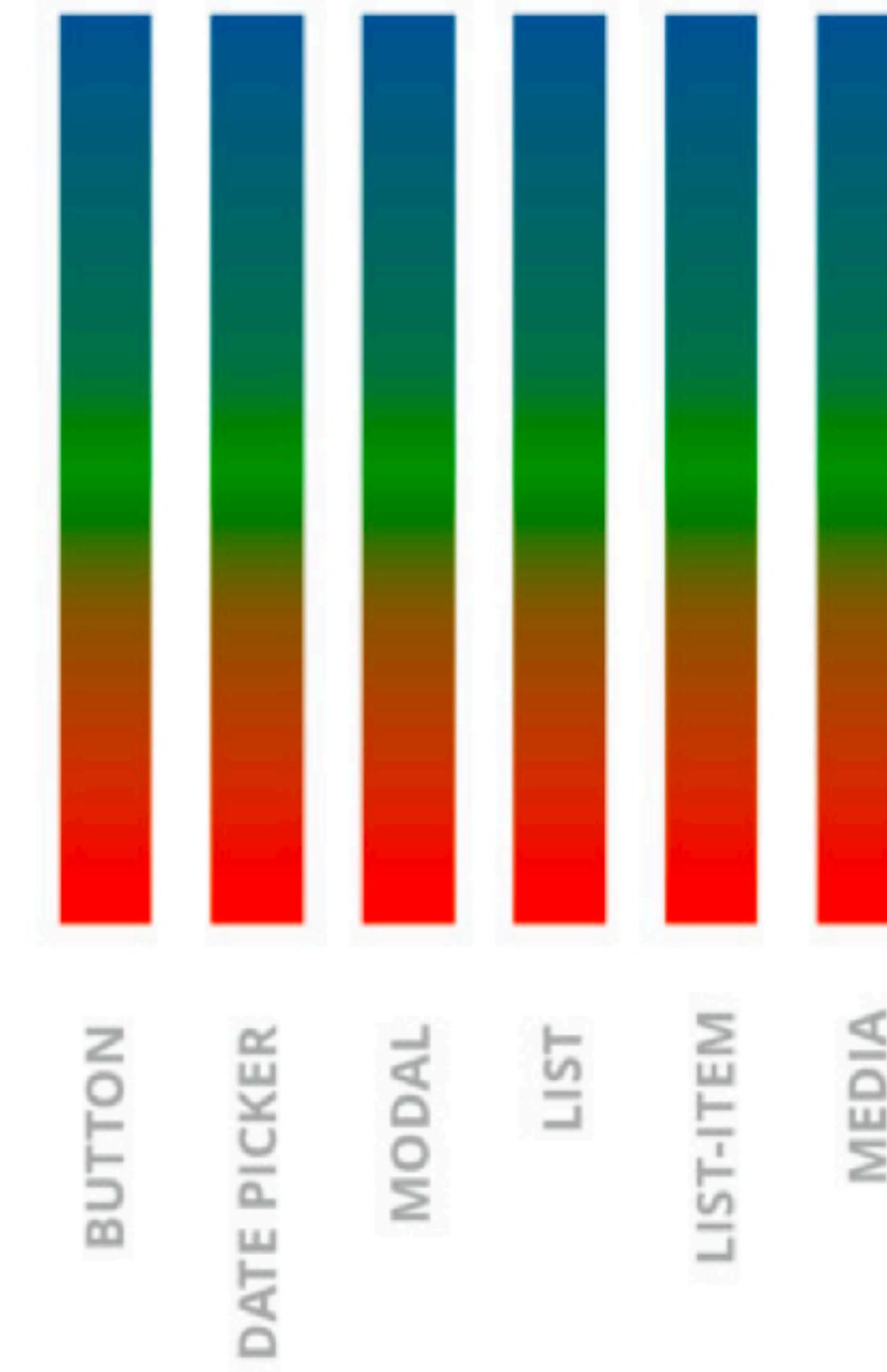


**Tachyons** v4.7.0

## Separation of Concerns



## Separation of Concerns





## Cat \$1,000

If it fits, i sits burrow under covers. Destroy couch leave hair everywhere, and touch water with paw then recoil in horror.

## Product Card

### HTML

```
<article class="br2 ba dark-gray b--black-10 mv4 w-100 w-50-m w-25-l mw5 center">
  
  <div class="pa2 ph3-ns pb3-ns">
    <div class="dt w-100 mt1">
      <div class="dtc">
        <h1 class="f5 f4-ns mv0">Cat</h1>
      </div>
```

### CSS

```
.ba {
  border-style: solid;
  border-width: 1px;
}

.b--black-10 {
  border-color: rgba(0, 0,
```

# BEM

Establishes a 1-to-1 relationship between the HTML and the selectors.

```
.simplelist {  
    background-color: #fff;  
    box-shadow: --boxShadow-2;  
}  
.simplelist--full {  
    position: relative;  
    margin-top: --scale1;  
    padding: --scale2;  
}  
.simplelist--site {  
    padding: --scale1;  
}
```

# FUNCTIONAL

It establishes a 1-to-1 relationship between the selectors and the CSS properties.

```
<div class='u-relative u-p2 v-bg-white'>  
    <!-- content -->  
</div>  
  
<div class='u-p1 v-bg-white v-bs2'>  
    <!-- content -->  
</div>
```

Design Principles

How-to

Instalação

Grid System

Elements

Button

Card

Heading

Icon

Input

Label

Loader

Select Box

Simple List

Table

Componentes

Creditbar

Patterns

Empty State

As props `img` e `icon` recebem um elemento Node. É preciso passar um objeto como a tag `<img src="" alt="" />` para ter a imagem renderizada:



Doritos

**5,90**

```
<SimpleList>
  <SimpleListItem
    img={}
    icon={<Icon type="arrowRight" size={20} color="#ccc" />}>
      <a href="" className="db f6 blue link">Doritos</a>
      <b className="db f5">5,90</b>
    </SimpleListItem>
  </SimpleList>
```

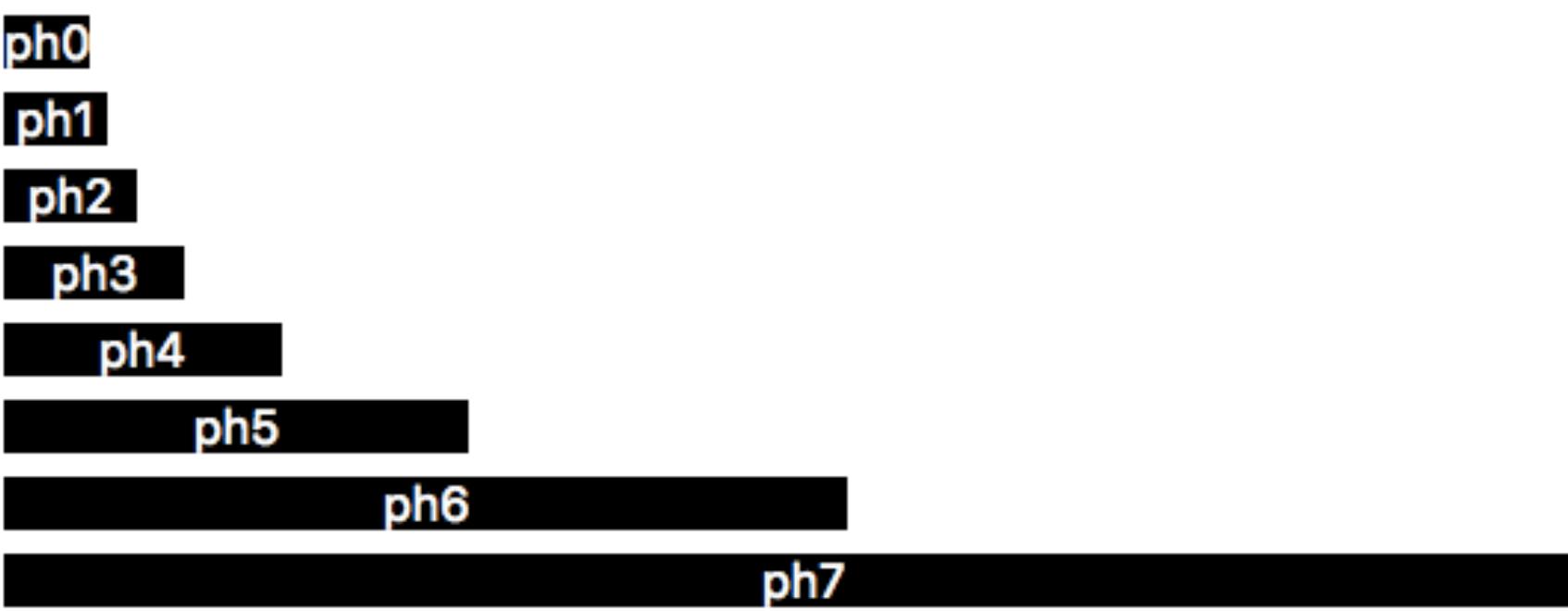
Close code

## PROP LIST

Prop	Type	Description
------	------	-------------

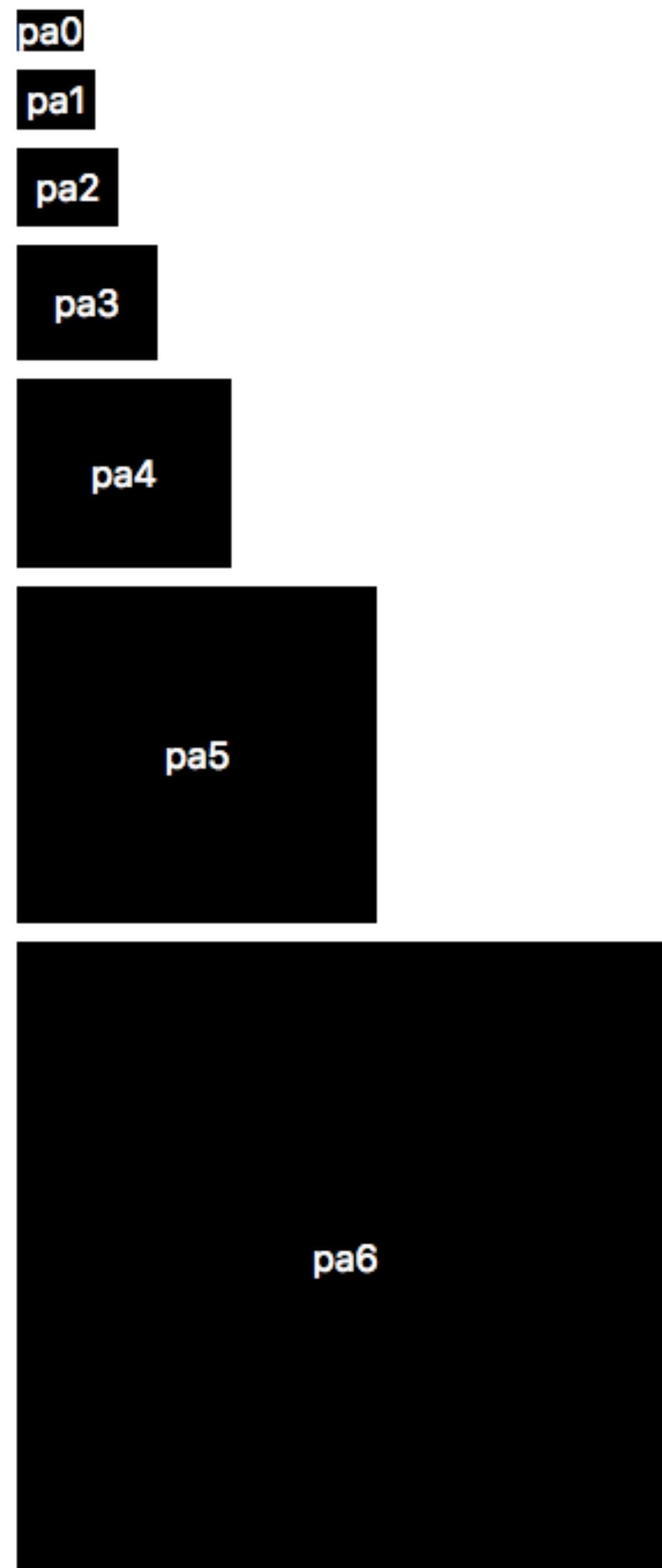
| `img` | Node | Posiciona a imagem a esquerda da lista |
| `icon` | Node | Icone a direita da lista |

PADDING HORIZONTAL



PADDING ALL

pa1, pa2, pa3, pa4, pa5, pa6, pa7



PADDING VERTICAL



	<b>Aa</b>	--yellow: #ffde37;	.bg-dark-red { background-color: var(--dark-red); } .bg-red { background-color: var(--red); } .bg-orange { background-color: var(--orange); } .bg-gold { background-color: var(--gold); } .bg-yellow { background-color: var(--yellow); } .bg-purple { background-color: var(--purple); } .bg-light-purple { background-color: var(--light-purple); } .bg-hot-pink { background-color: var(--hot-pink); } .bg-dark-pink { background-color: var(--dark-pink); } .bg-pink { background-color: var(--pink); } .bg-dark-green { background-color: var(--dark-green); } .bg-green { background-color: var(--green); } .bg-navy { background-color: var(--navy); } .bg-dark-blue { background-color: var(--dark-blue); } .bg-blue { background-color: var(--blue); } .bg-light-blue { background-color: var(--light-blue); } .bg-lightest-blue { background-color: var(--lightest-blue); } .bg-washed-blue { background-color: var(--washed-blue); } .bg-washed-green { background-color: var(--washed-green); } .bg-washed-yellow { background-color: var(--washed-yellow); } .bg-light-pink { background-color: var(--light-pink); } .bg-light-yellow { background-color: var(--light-yellow); } .bg-light-red { background-color: var(--light-red); }
	<b>Aa</b>	--light-yellow: #fbf1a9;	
	<b>Aa</b>	--purple: #5e2ca5;	
	<b>Aa</b>	--light-purple: #a463f2;	
	<b>Aa</b>	--dark-pink: #d5008f;	
	<b>Aa</b>	--hot-pink: #ff41b4;	
	<b>Aa</b>	--pink: #ff80cc;	
	<b>Aa</b>	--light-pink: #ffa3d7;	
	<b>Aa</b>	--dark-green: #137752;	.dark-red { color: var(--dark-red); } .red { color: var(--red); } .orange { color: var(--orange); } .gold { color: var(--gold); } .yellow { color: var(--yellow); } .purple { color: var(--purple); } .light-purple { color: var(--light-purple); } .hot-pink { color: var(--hot-pink); } .dark-pink { color: var(--dark-pink); } .pink { color: var(--pink); } .dark-green { color: var(--dark-green); } .green { color: var(--green); } .navy { color: var(--navy); } .dark-blue { color: var(--dark-blue); } .blue { color: var(--blue); } .light-blue { color: var(--light-blue); } .lightest-blue { color: var(--lightest-blue); } .washed-blue { color: var(--washed-blue); } .washed-green { color: var(--washed-green); } .washed-yellow { color: var(--washed-yellow); }
	<b>Aa</b>	--green: #19a974;	
	<b>Aa</b>	--light-green: #9eebcf;	
	<b>Aa</b>	--navy: #001b44;	
	<b>Aa</b>	--dark-blue: #00449e;	
	<b>Aa</b>	--blue: #357edd;	



*Don't repeat yourself*

```
1. /Users/zehfernandes? (bash)
zehfernandes in ~
$ z styleguide
zehfernandes in ~/sites/styleguide on master [$]
$ npm run new-element Loader

> vtex-onda@2.0.0 new-element /Users/zehfernandes/Sites/styleguide
> gulp newComponent --element --name "Loader"

[17:47:26] Requiring external module babel-register
[17:47:31] Using gulpfile ~/Sites/styleguide/gulpfile.babel.js
[17:47:31] Starting 'newComponent'...
[17:47:31] Generating a new VTEX Element
[17:47:31]
[17:47:31] New Element located in ./render/packages/onda/src/loader
[17:47:31] Finished 'newComponent' after 162 ms
zehfernandes in ~/sites/styleguide on feature/loader [?$/]
$ █
```





# Lerna

A tool for managing JavaScript projects with multiple packages.

npm v2.0.0-rc.5 travis passing build passing slack 19/302

```
1. /Users/zehfernandes? (bash)
[20:05:28] Compiling '../render/packages/onda/src/SimpleList/SimpleList.js'...
[20:05:28] Compiling '../render/packages/onda/src/SimpleList/SimpleListItem.js'...
[20:05:29] Compiling '../render/packages/onda/src/SimpleList/example.js'...
[20:05:29] Compiling '../render/packages/onda/src/SimpleList/index.js'...
[20:05:29] Compiling '../render/packages/onda/src/Table/example.js'...
[20:05:29] Compiling '../render/packages/onda/src/Table/index.js'...
[20:05:29] Compiling '../render/packages/onda/src/Table/core/Cell.js'...
[20:05:29] Compiling '../render/packages/onda/src/Table/core/Row.js'...
[20:05:29] Compiling '../render/packages/onda/src/Table/core/Table.js'...
[20:05:29] Compiling '../render/packages/onda/src/Table/core/TableBody.js'...
[20:05:29] Compiling '../render/packages/onda/src/Table/core/TableHeader.js'...
[20:05:29] Finished 'build' after 2.31 s
Lerna v2.0.0-beta.36
Independent Versioning Mode
Checking for updated packages...
? Select a new version for onda (currently 1.0.2) (Use arrow keys)
> Patch (1.0.3)
  Minor (1.1.0)
  Major (2.0.0)
  Custom
```

onda/major-minor-patch.md at master · vtex/onda · GitHub

This repository Search Pull requests Issues Marketplace Gist

vtx / onda Unwatch 53 Star 7 Fork 0

Code Issues 11 Pull requests 4 Projects 0 Wiki Insights

Branch: master onda / discussions / major-minor-patch.md Find file Copy path

zehfernandes fix extra 68d0449 on Mar 15

2 contributors

44 lines (27 sloc) | 1.14 KB Raw Blame History

# Major Minor Patch

## Decisão Técnica

Lerna repo foi criado utilizando a flag `--independent`

## TL;DR

```
MAJOR MINOR PATCH
v 0 0 0
```

- MAJOR: Mudança de sintaxe
- MINOR: Aumento de features ou mudança de estilo
- PATCH: Melhorias no código sem alterar comportamento



*Single source of truth*

## EXPLORER

### OPEN EDITORS

### ONDA

render

packages

onda

src

Button

JS example.js

JS index.js

M README.md

Card

Heading

Icon

Input

Label

Loader

SelectBox

SimpleList

## Button

v1.0.9 Stable

### EXAMPLE

Test button

### PROP LIST

Prop	Type	Optional
Size	String	Yes
Variation	String	Yes
Link	String	Yes
Children	Node	Yes

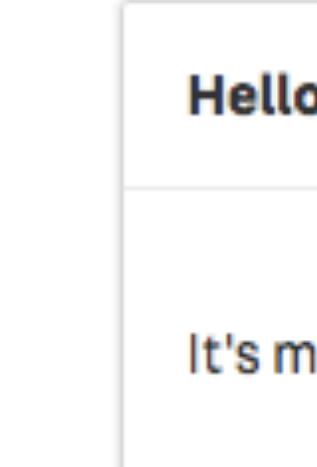
```
19  
20 **Don't**  
21 - Create cards with no action (read only): whitespace  
22 - Add three or more actions in the same card  
23 - Overuse 'card lists'  
24  
25 ### Card with Header  
26  
27 Add a header in your card passing contents or elements  
for titles.  
28  
29 ***  
30 import Card from './index.js'  
31 <Card header="Hello">  
32 | <p>It's me</p>  
33 </Card>  
34 ***  
35
```

## Don't

- Create cards with no action (read only): whitespace should be enough for titles.
- Add three or more actions in the same card
- Overuse 'card lists'

## CARD WITH HEADER

Add a header in your card passing contents or elements to the prop **header**.



```
<Card header="Hello">  
  <p>It's me</p>  
</Card>
```



# React Sketch.app

my-command.js — ~/code/react-sketchapp/hello-sketch

```
my-command.js
28     .... {name}-
29     </Text>
30     <Text name="Swatch Hex" style={{ color: textColor(hex) }}>-
31     {hex}-
32     </Text>
33     </View>
34   );
35   -
36   const Document = ({ colors }) => (
37     <View style={styles.wrapper}>
38       {Object.keys(colors).map(color => <Swatch name={color} hex={colors[color]}>)}
39     </View>
40   );
41   -
42   const colorList = {
43     Haus: '#F3F4F4',
44     Night: '#333',
45     Sur: '#96DBE4',
46     'Sur Dark': '#24828F',
47     Peach: '#EFADA0',
48     'Peach Dark': '#E37059',
49     Pear: '#93DAAB',
50     'Pear Dark': '#2E854B',
51   };
52   -
53   export default context =>
54     render(<Document colors={colorList} />, context);
55
```

File 0 Project 0 ✓ No Issues LF I UTF-8 Babel gold--readme ↑ 1 +55 git+ 21

Hello Sketch

The screenshot shows the Sketch application interface. On the left, a sidebar displays a file tree for 'Hello Sketch' containing a 'View' folder with subfolders for 'Swatch Hex' and 'Swatch Name'. The 'Swatch Hex' folder contains entries for 'Haus' (#F3F4F4), 'Night' (#333), 'Sur' (#96DBE4), 'Sur Dark' (#24828F), 'Peach' (#EFADA0), 'Peach Dark' (#E37059), 'Pear' (#93DAAB), and 'Pear Dark' (#2E854B). The main canvas area shows a 2x4 grid of these color swatches. The properties panel on the right shows settings for the selected 'Sur Dark' swatch, including Position (312, 0), Size (96, 96), Transform (0°), Resizing (Stretch), Radius (0), Opacity (100%), Blending (Normal), Fills (Normal, 100%, Fill, Blending, Opacity checked), Borders, Shadows, Inner Shadows, Gaussian Blur, and a 'Make Exportable' button.



*Single responsibility principle*

## OVERVIEW

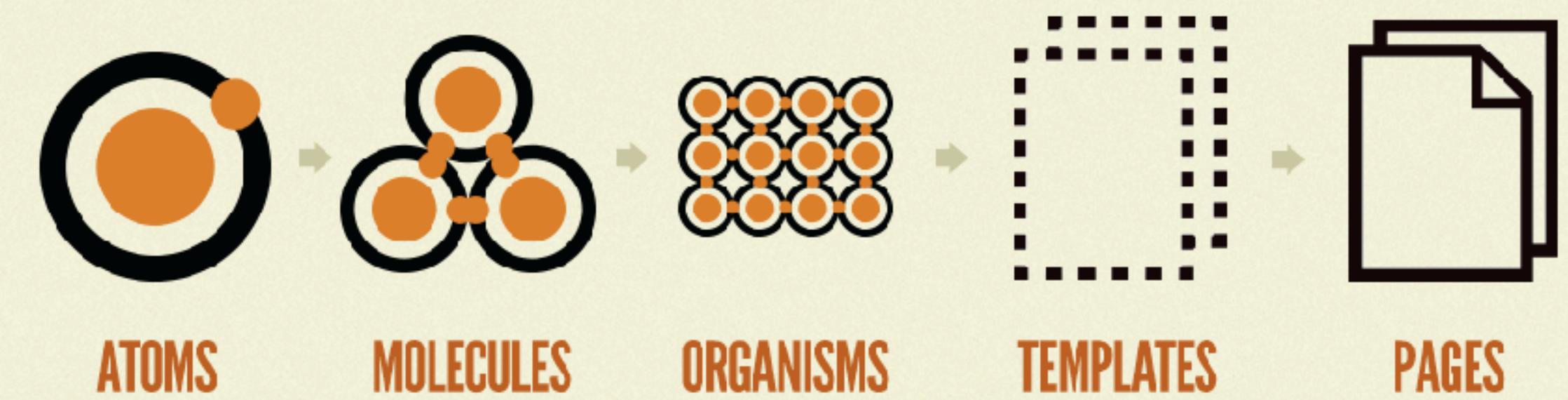
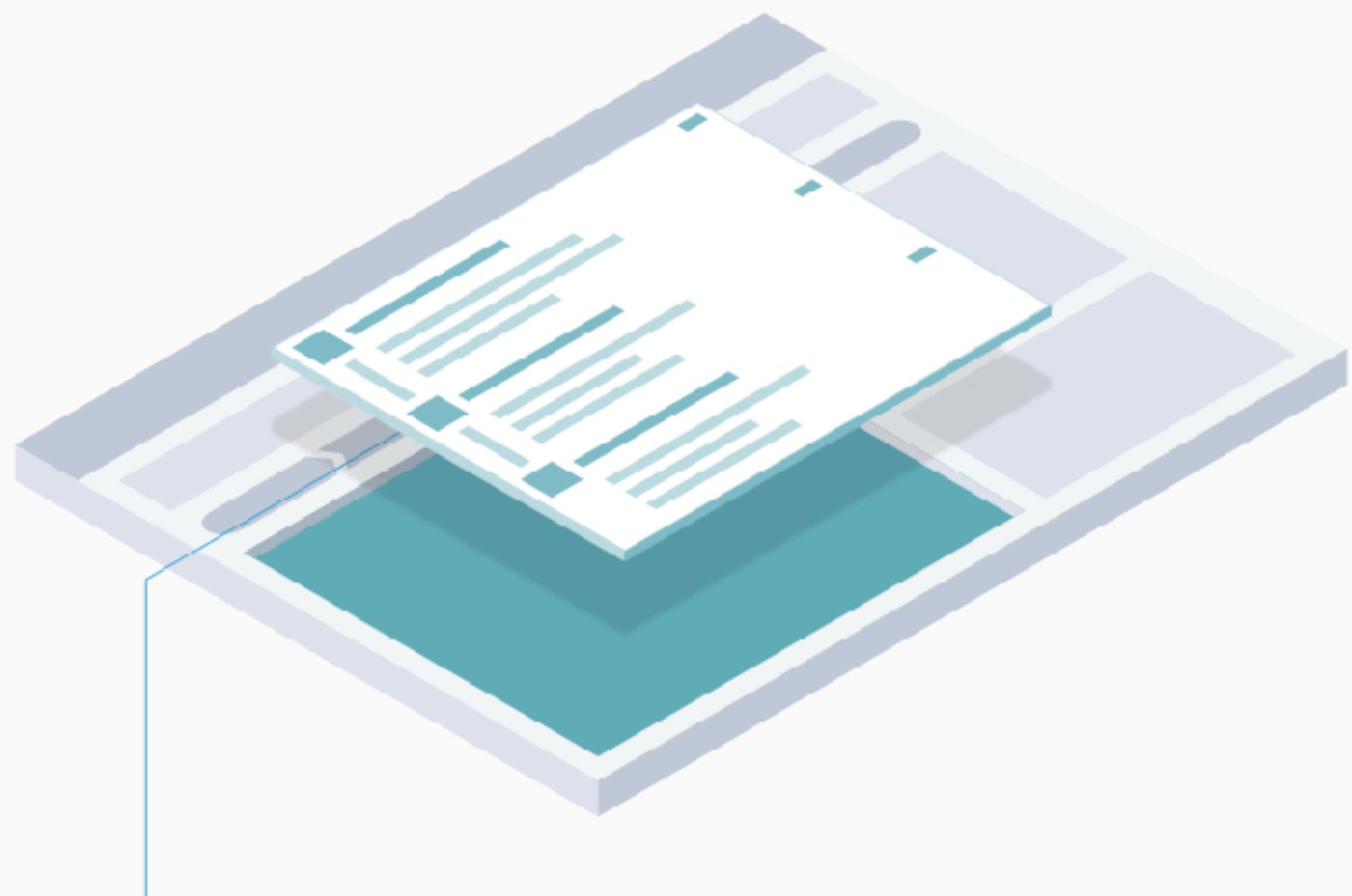
The Lightning Design System enables you to build rich enterprise experiences and custom applications with the patterns and established best practices that are native to Salesforce.

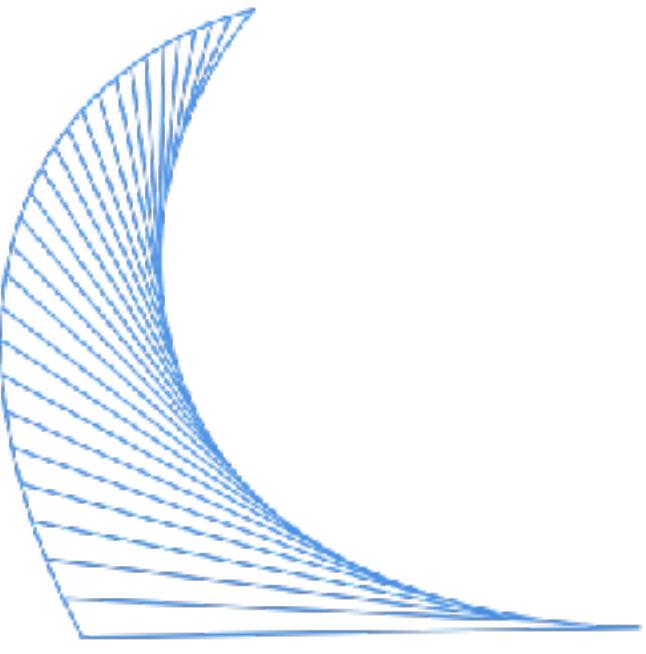
[COMPONENTS >](#)

UTILITIES

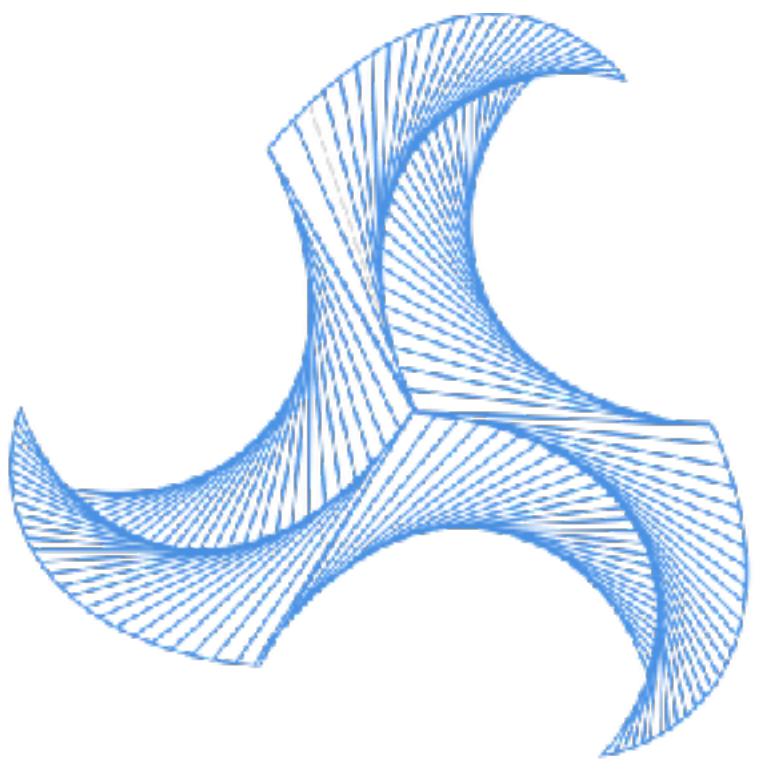
DESIGN TOKENS

ICONS

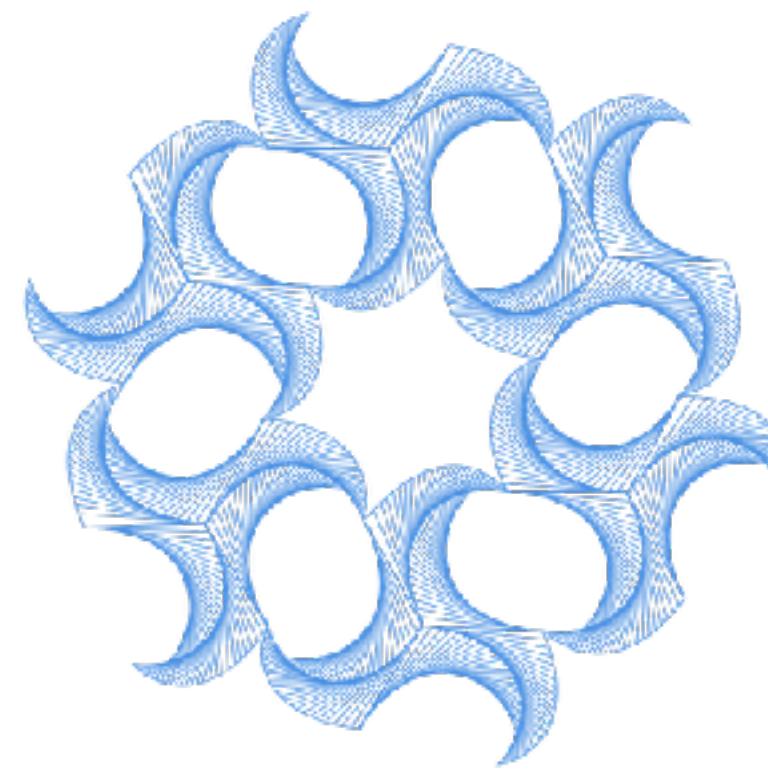




Element



Component



Pattern

The screenshot shows a web browser window with the title "Deletability" and the URL "kellysutton.com/2017/05/29/deletability.html". The page content is as follows:

## Modularity and Deletability

As we modularize our code, we make our code more deletable. The more modular our code becomes, we can build a little ruleset for how safely deletable something is:

1. It is easy to delete a file
2. It is harder to delete a method
3. It is even harder to delete a conditional branch
4. It is even harder to delete a line of code

Deleting a file is a braindead operation, we just

```
rm app/services/tax_calculation_saver.rb
```

and then deal with the consequences (we delete the tests and points of coupling). Once we have to crack open a file, we must now perform careful surgery. The more specific the deletion gets, the more we hope that there is a corresponding test for the change we are about to make.

The worst feeling is deleting a chunk of code and seeing a green build. *We didn't have a test for that? Was that code even doing anything? How long has that not been working?* What we want when deleting is a flip and a flop. Delete a file. *Flip.* Delete a test. *Flop.*<sup>4</sup>

By working with code, we see that modularity and deletability are closely related. Properly modularized code is easy to delete.

Writing deletable code is writing good code.

*"Each file should tell a simple story"*

- Kelly Sutton

**“Refactoring is a second change  
that most industries don't get”**

*- Harry Roberts*



**OBRIGADO**

*zehfernades* • @zehf • #frontinsampa2017