

UI Development Without the Grind

It doesn't have to be this difficult





Sarah Padovani

Web Team Lead

- Maritimer
- Avid Baker
- Gong Show Master



Sarah Padovani

Web Team Lead

- Maritimer
- Avid Baker
- Gong Show Master



Sarah Padovani

Web Team Lead

- Maritimer
- Avid Baker
- Gong Show Master



Sarah Padovani

Web Team Lead

- Maritimer
- Avid Baker
- Gong Show Master

What is “The Grind”?

as told through



A Tale of Three Homepages



Once upon a time, there lived a humble trio of developers. They toiled each day to keep an aging website running, even as its magic slowly faded. Fewer travellers stopped to read its stories, and the kingdom's coffers began to shrink.



One day, the High Council of Business summoned the trio with a noble quest: to craft a brand-new homepage that would bring prosperity back to the land.

For many moons, they laboured,
crafting a homepage that sparkled with
promise. The big launch day was just
over the horizon.



But alas! Just one week before the unveiling, a decree came down from the tower — a product sorcerer declared that the new homepage did not honour the sacred tenets of the Brand.





The trio did not falter.

A new guild of designers arrived, bearing scrolls and symbols blessed by the branding sages. The trio, ever faithful, took up their tools once more and built anew.



The new homepage was launched to great fanfare. The people rejoiced, traffic returned, and the site was honoured with the prestigious Order of the Webby — a token of its greatness.



But peace is often fleeting in the lands of Product and Design.

From distant parts of the kingdom came new branding edicts—mysterious changes from beyond their walls. Once again, the homepage was no longer in alignment.



And so, the developers — still the same three — rebuilt it a third time.

This time, there was no celebration. Only quiet exhaustion. The trio had followed every command, answered every scroll, and met every deadline. But with each rebuild, their spark dimmed. The joy of creation had been replaced by the weariness of repetition.

They had done what was asked — but at great cost.



So, What is “The Grind”?

- It's needlessly difficult
- != progress
- It's demoralizing

“The needlessly difficult or monotonous tasks that prevent you from focusing on the technically interesting challenges”



WHAT IS “THE GRIND”?

Variables



WHAT IS “THE GRIND”?

Variables

CSS variables.css

```
1 :root {
2   --light: #ffffff;
3   --dark: #1A1E23;
4   --teal: #12F7D6;
5   --teal-light: #98FAEC;
6   --background: #292F36;
7 }
```



About Me

Hello!

My name is Sinan and I specialize in web development that utilizes **HTML**, **CSS**, **JS**, and **REACT** etc. I am a highly motivated individual and eternal optimist dedicated to writing clear, concise, robust code that works. Striving to never stop learning and improving. When I'm not coding, I am **writing bolgs**, reading, or picking up some new hands-on art project like **photography**. I like to have my perspective and belief systems challenged so that I see the world through new eyes.

WHAT IS “THE GRIND”?

Variables



CSS variables.css

```
1 :root {
2   --light: #ffffff;
3   --dark: #1A1E23;
4   --teal: #12F7D6;
5   --teal-light: #98FAEC;
6   --background: #292F36;
7 }
```

<C/> SinanTokmak

HomeBlogs

InstagramDiscordGithub

Sinan

Full-stack developer

✉ abdurrahman_sinan@hotmail.com

📍 Turkey

🏠 Full-time / Freelancer

🌐 www.sinantokmak.com

HTMLCSSJSREACT

Download CV

Hey

I'm Sinan,

Full-Stack Developer

I help business grow by crafting amazing web experiences. If you're looking for a developer that likes to get stuff done,

Let's Talk

4

Programming Language

6

Development Tools

8

Years of Experience

About Me

```
<p>
Hello!
My name is Sinan and I specialize in web development that utilizes HTML, CSS, JS, and REACT etc.
I am a highly motivated individual and eternal optimist dedicated to writing clear, concise, robust code that works. Striving to never stop learning and improving. When I'm not coding, I am writing bolgs, reading, or picking up some new hands-on art project like photography.
I like to have my perspective and belief systems challenged so that I see the world through new eyes.
</p>
```



WHAT IS “THE GRIND”?

Variables



CSS variables.css

```
1 :root {
2   --light: #ffffff;
3   --dark: #1A1E23;
4   --darker: #000000;
5   --teal: #12F7D6;
6   --teal-light: #98FAEC;
7   --background: #292F36;
8 }
```

<C/> SinanTokmak

Home Blogs

Instagram Discord Github

☰

👤

⏏

📄

✍

✉

Sinan

Full-stack developer

✉ abdurrahman_sinan@hotmail.com

📍 Turkey

🏠 Full-time / Freelancer

🌐 www.sinantokmak.com

HTML

CSS

JS

REACT

Download CV ⬇

<h1>

Hey I'm Sinan, Full-Stack Developer

</h1>

<p>

I help business grow by crafting amazing web experiences. If you're looking for a developer that likes to get stuff done,

</p>

Let's Talk ✉

4 Programming Language

6 Development Tools

8 Years of Experience

About Me

Hello!

My name is Sinan and I specialize in web development that utilizes HTML, CSS, JS, and REACT etc. I am a highly motivated individual and eternal optimist dedicated to writing clear, concise, robust code that works. Striving to never stop learning and improving. When I'm not coding, I am writing bolgs, reading, or picking up some new hands-on art project like photography. I like to have my perspective and belief systems challenged so that I see the world through new eyes.

WHAT IS “THE GRIND”?

Variables



CSS variables.css

```
1 :root {
2   --light: #ffffff;
3   --dark: #1A1E23;
4   --darker: #000000;
5   --pink: #E33647;
6   --pink-light: #EC9D9C;
7   --background: #292F36;
8 }
```

<C/> SinanTokmak

HomeBlogs

InstagramDiscordGithub

☰

☹

<>

📄

✍

✉

Sinan

Full-stack developer

✉ abdurrahman_sinan@hotmail.com

📍 Turkey

🏠 Full-time / Freelancer

🌐 www.sinantokmak.com

HTMLCSSJSREACT

Download CV ⬇

<h1>

Hey I'm Sinan, Full-Stack Developer </h1>

<p>

I help business grow by crafting amazing web experiences. If you're looking for a developer that likes to get stuff done,

</p>

Let's Talk ✉

4 Programming Language

6 Development Tools

8 Years of Experience

About Me

<p>

Hello!

My name is Sinan and I specialize in web development that utilizes HTML, CSS, JS, and REACT etc.

I am a highly motivated individual and eternal optimist dedicated to writing clear, concise, robust code that works. Striving to never stop learning and improving.

When I'm not coding, I am writing bolgs, reading, or picking up some new hands-on art project like photography.

I like to have my perspective and belief systems challenged so that I see the world through new eyes.

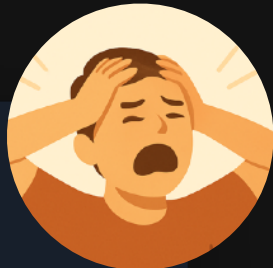
</p>

WHAT IS “THE GRIND”?

Variables

CSS variables.css

```
1 :root {
2   --light: #ffffff;
3   --dark: #1A1E23;
4   --darker: #000000;
5   --pink: #E33647;
6   --pink-light: #EC9D9C;
7   --background: #292F36;
8 }
```



<C/> SinanTokmak

Home Blogs



Instagram

Discord

Github

Developer

Hey
I'm Sinan,
Full-Stack Developer

crafting amazing web experiences. If you're
that likes to get stuff done,

41k



4 Programming
Language

6 Development
Tools

8 Years of
Experience

About Me

Hello!

My name is Sinan and I specialize in web development that utilizes HTML, CSS, JS, and REACT etc.
I am a highly motivated individual and eternal optimist dedicated to writing clear, concise, robust code that works. Striving to never stop learning and improving.
When I'm not coding, I am writing bolgs, reading, or picking up some new hands-on art project like photography.
I like to have my perspective and belief systems challenged so that I see the world through new eyes.



WHAT IS “THE GRIND”?

Monoliths



Donuts

A donut is more than just a pastry, it is a wish fulfilled

[Buy Now](#)

WHAT IS “THE GRIND”?

Monoliths

Card.tsx

```
1 <Card
2   title="Donuts"
3   description="A donut is more than..."
4   thumbnailUrl="https://unsplash.com/image.jpg"
5   link="https://buy.now"
6 />
```



Donuts

A donut is more than just a pastry, it is a wish fulfilled

Buy Now

WHAT IS “THE GRIND”?

Monoliths

🔗 Card.tsx

```
1 <Card
2   title="Donuts"
3   description="A donut is more than..."
4   thumbnailUrl="https://unsplash.com/image.jpg"
5   link="https://buy.now"
6 />
```



Donuts

A donut is more than just a pastry, it is a wish fulfilled

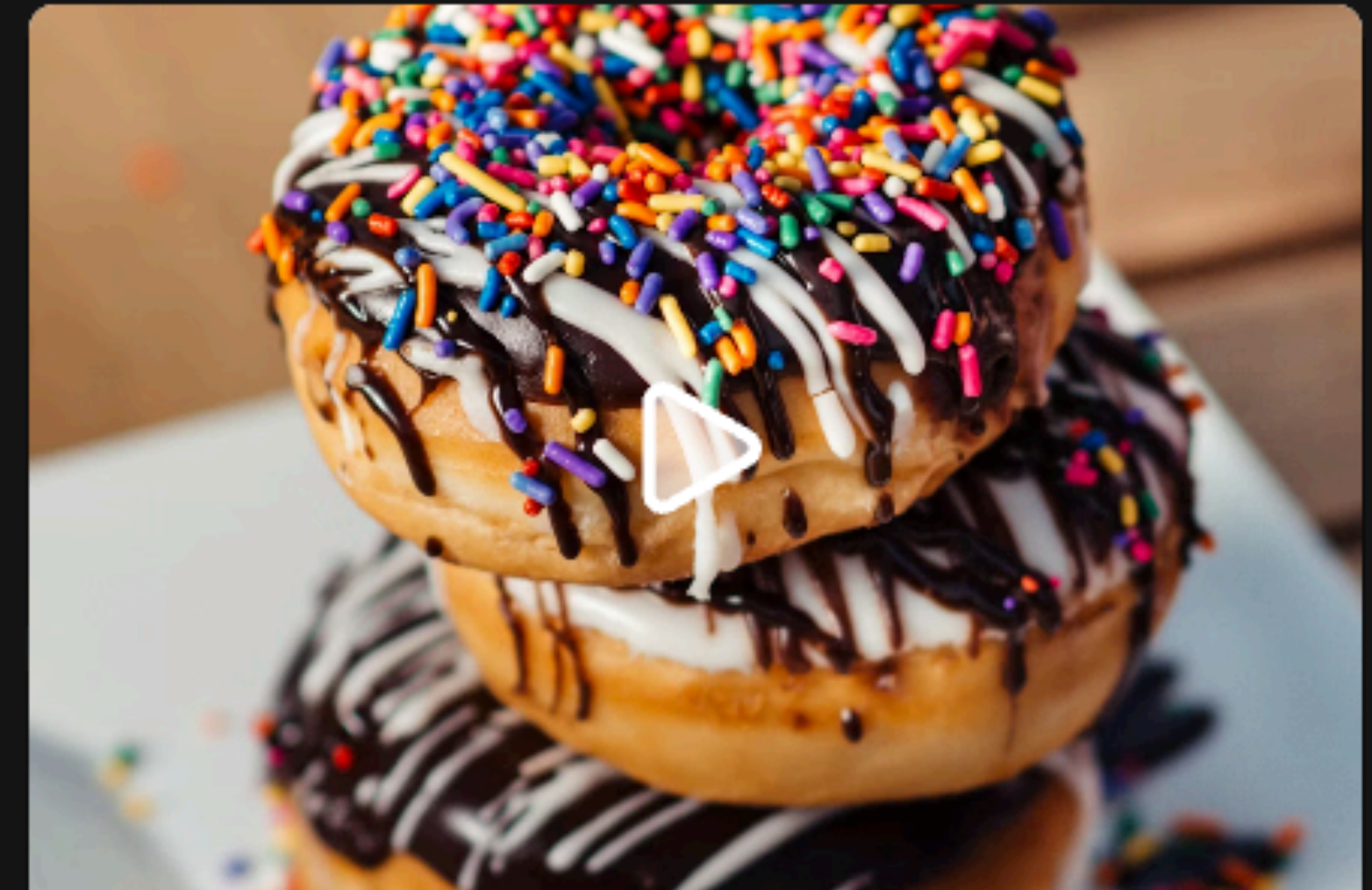
Buy Now

WHAT IS “THE GRIND”?

Monoliths

Card.tsx

```
1 <Card
2   title="Donuts"
3   description="A donut is more than..."
4   thumbnailUrl="https://unsplash.com/image.jpg"
5   link="https://buy.now"
6   videoUrl="https://unsplash.com/video.mp4"
7 />
```



Donuts

A donut is more than just a pastry, it is a wish fulfilled

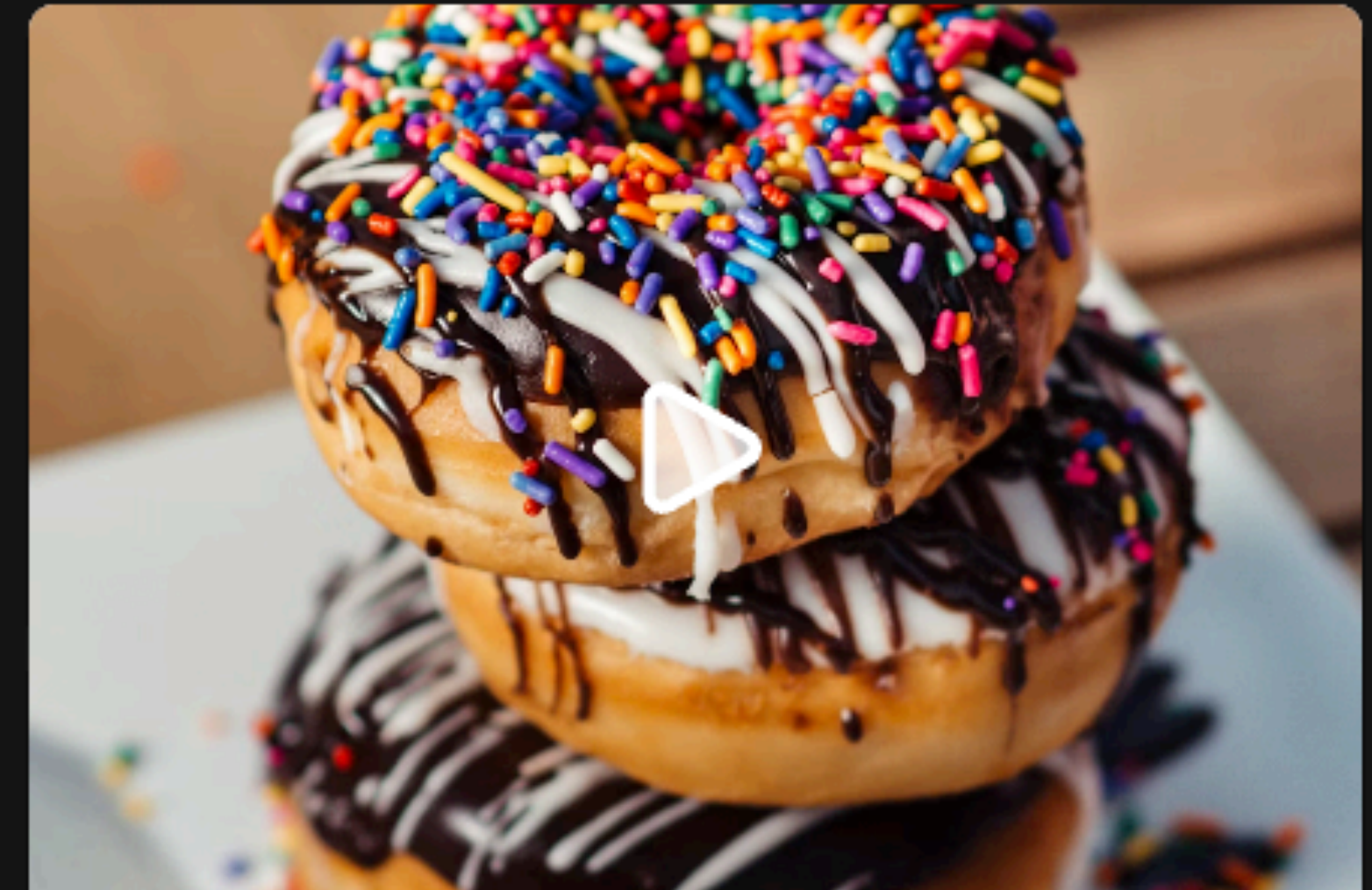
Buy Now

WHAT IS “THE GRIND”?

Monoliths

Card.tsx

```
1 <Card
2   title="Donuts"
3   description="A donut is more than..."
4   thumbnailUrl="https://unsplash.com/image.jpg"
5   link="https://buy.now"
6   videoUrl="https://unsplash.com/video.mp4"
7   secondaryLink="https://nutritional.info"
8 />
```



Donuts

A donut is more than just a pastry, it is a wish fulfilled

Buy Now

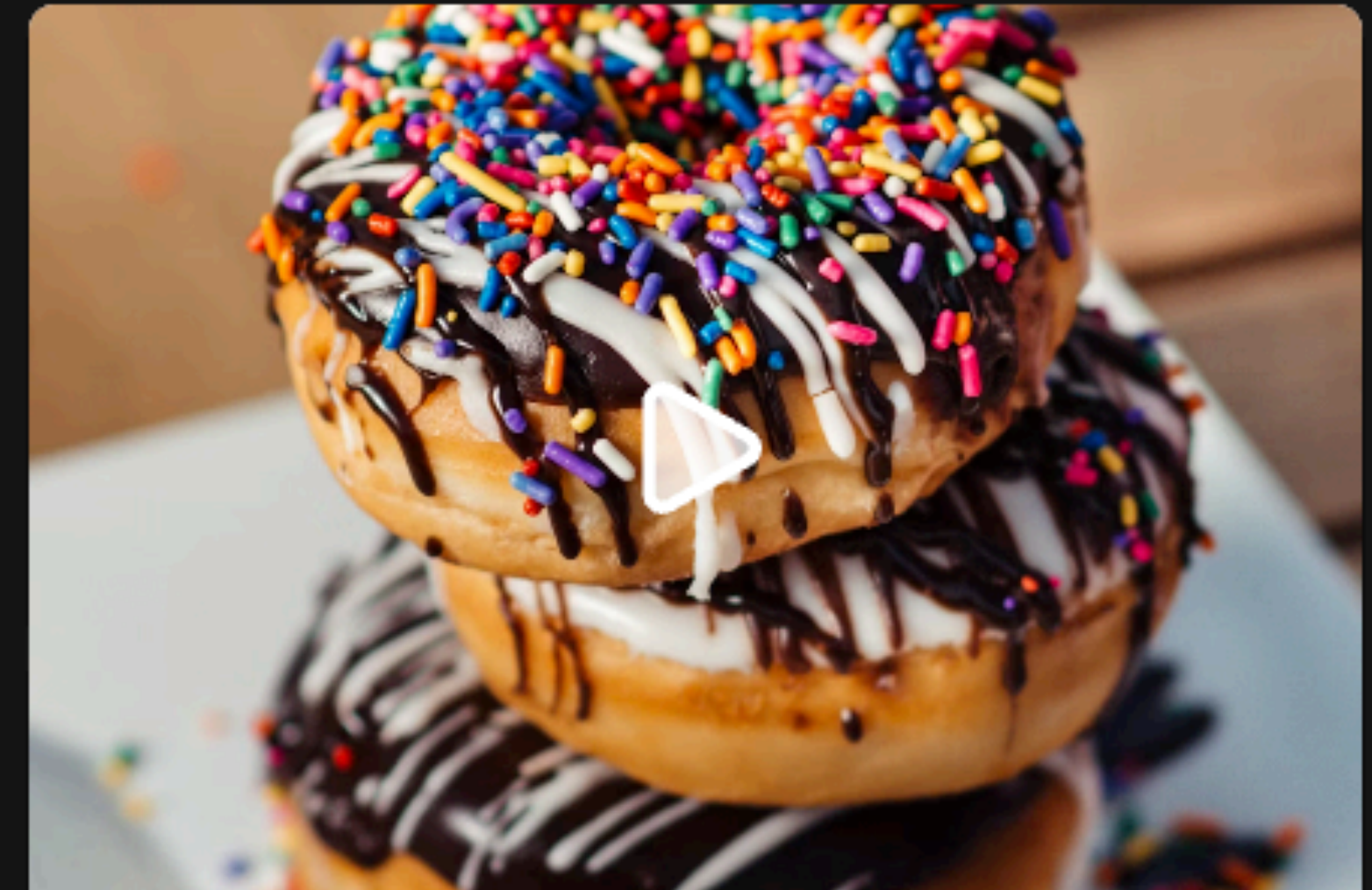
Nutritional Info

WHAT IS “THE GRIND”?

Monoliths

Card.tsx

```
1 <Card
2   title="Donuts"
3   description="A donut is more than..."
4   thumbnailUrl="https://unsplash.com/image.jpg"
5   link="https://buy.now"
6   videoUrl="https://unsplash.com/video.mp4"
7   secondaryLink="https://nutritional.info"
8   stars="4.5"
9 />
```



Donuts ★ ★ ★ ★ ★

A donut is more than just a pastry, it is a wish fulfilled

Buy Now

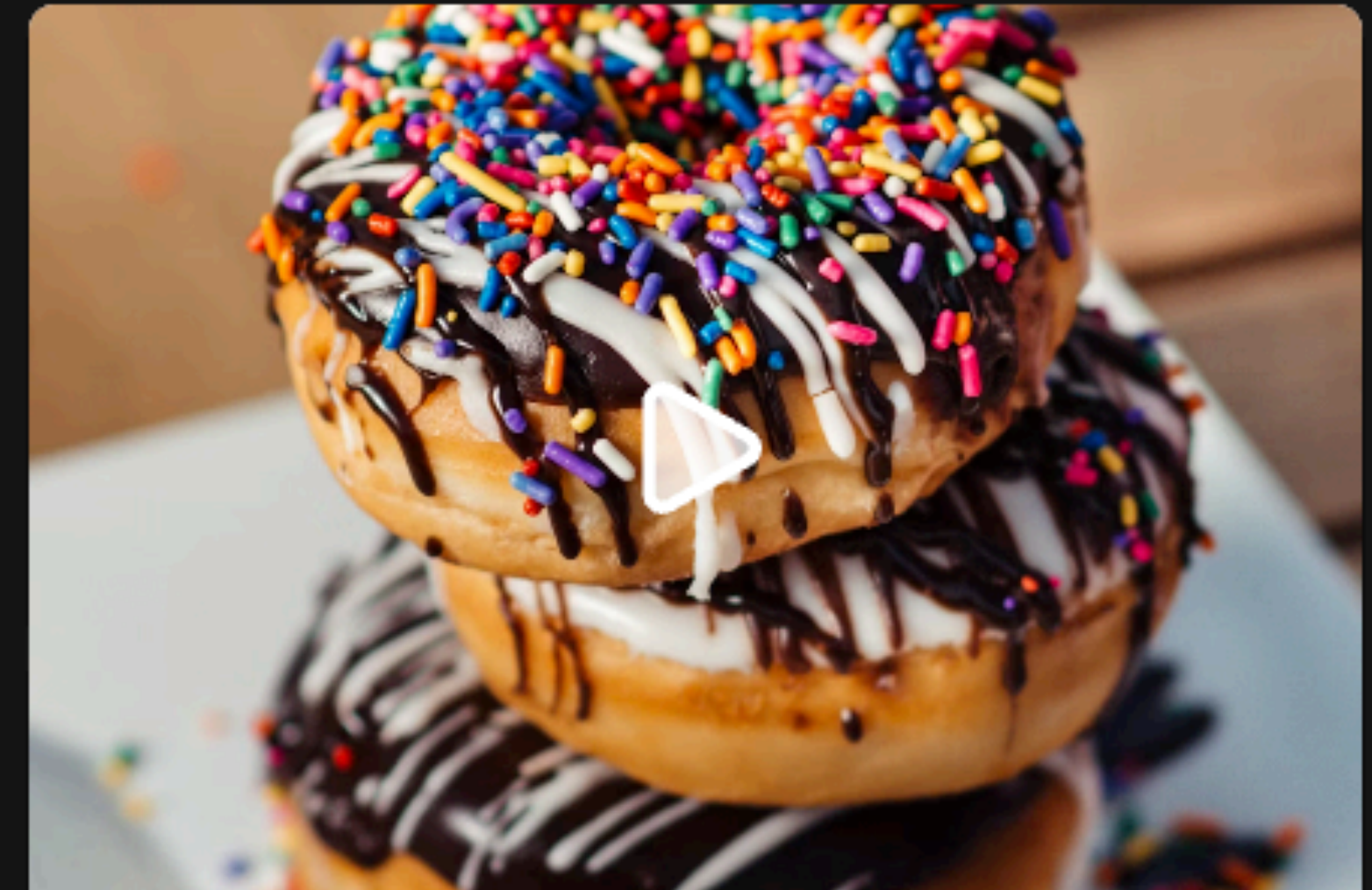
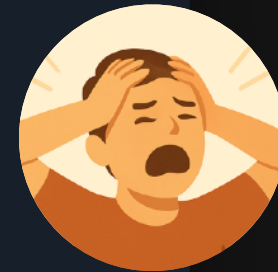
Nutritional Info

WHAT IS "THE GRIND"?

Monoliths

⚙ Card.tsx

```
1 <Card
2   title="Donuts"
3   description="A donut is more than..."
4   thumbnailUrl="https://unsplash.com/image.jpg"
5   link="https://buy.now"
6   videoUrl="https://unsplash.com/video.mp4"
7   secondaryLink="https://nutritional.info"
8   stars="4.5"
9   productId="001234"
10  trackingId="123456789"
11  promotionCode="abc-123-def-456"
12  rating="E"
13  backgroundColor="#ffffff"
14  isFavourite={true}
15  productInStock={true}
16  onTrackInteraction={track}
17  onSubmitRating={submit}
18 />
```



Donuts ★ ★ ★ ★ ★

A donut is more than just a pastry, it is a wish fulfilled

Buy Now

Nutritional Info

WHAT IS "THE GRIND"?

Monoliths

Card.tsx

```
1 <Card
2   title="Donuts"
3   description="A donut is more than."
4   thumbnailUrl="https://unsplash.co
5   link="https://buy.now"
6   videoUrl="https://unsplash.com/
7   secondaryLink="https://nutriti
8   stars="4.5"
9   productId="001234"
10  trackingId="123456789"
11  promotionCode="abc-123"
12  rating="E"
13  backgroundColor="#ffffff"
14  isFavourite={true}
15  productInStock={true}
16  onTrackInteraction={track}
17  onSubmitRating={submit}
18 />
```



Donuts ★ ★ ★ ★ ★

A donut is more than just a pastry, it is a wish fulfilled

Buy Now

Nutritional Info

WHAT IS “THE GRIND”?

HTML Semantics

 Button.tsx

```
1 export const Button = ({ label, onClick }) => {  
2   return (  
3     <button className="button-styles" onClick={onClick}>  
4       {label}  
5     </button>  
6   )  
7 }
```


WHAT IS “THE GRIND”?

HTML Semantics



 Button.tsx

```
1 export const Button = ({ label, onClick }) => {  
2   return (  
3     <button className="button-styles" onClick={onClick}>  
4       {label}  
5     </button>  
6   )  
7 }
```


WHAT IS "THE GRIND"?

HTML Semantics



🔗 Button.tsx

```
1 export const Button = ({ label, onClick, href }) => {
2   if (href) {
3     return <a className="button-style" href={href}>{label}</a>
4   }
5
6   return (
7     <button className="button-styles" onClick={onClick}>
8       {label}
9     </button>
10  )
11 }
```


WHAT IS “THE GRIND”?

HTML Semantics

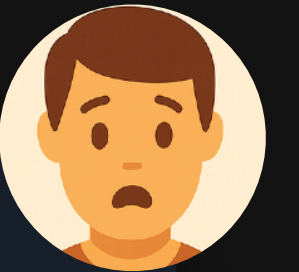
🔗 Button.tsx

```
1 export const Button = ({ label, onClick }) => {  
2   return (  
3     <button className="button-styles" onClick={onClick}>  
4       {label}  
5     </button>  
6   )  
7 }
```



🔗 Link.tsx

```
1 export const link = ({ label, href, display }) => {  
2   const className = display === "button"  
3     ? "button-styles"  
4     : "link-styles"  
5  
6   return (  
7     <a className={className} href={href}>  
8       {label}  
9     </a>  
10  )  
11 }
```

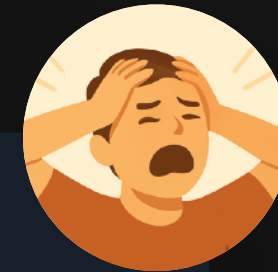


WHAT IS “THE GRIND”?

HTML Semantics

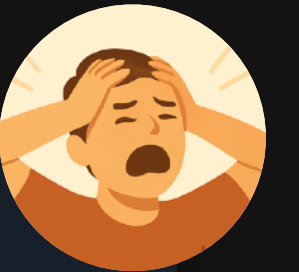
Button.tsx

```
1 export const Button = ({ label, onClick, disabled }) => {
2   return (
3     <button
4       className="button-styles"
5       onClick={onClick}
6       disabled={disabled}
7     >
8       {label}
9     </button>
10  )
11 }
```



Link.tsx

```
1 export const link = ({ label, href, display, disabled }) => {
2   const className = display === "button"
3     ? `button-styles ${disabled ? "disabled-styles" : ""}`
4     : "link-styles"
5
6   return (
7     <a className={className} href={href}>
8       {label}
9     </a>
10  )
11 }
```



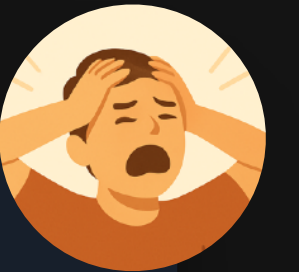
WHAT IS "THE GRIND"?

HTML Semantics

Button.tsx

```
1 export const Button = ({ label, onClick, disabled }) => {
2   return (
3     <button
4       className="button-styles"
5       onClick={onClick}
6       disabled={disabled}
7     >
8       {label}
9     </button>
10  )
11 }
```

```
link = ({ label, href, display, disabled }) => {
  display === "button"
  ? { disabled ? "disabled-styles" : "" }
  : {}
  {name} href={href}>
```



WHAT IS “THE GRIND”?

Patterns

 View.tsx

```
1 <Button color="primary" />  
2 <Icon color="secondary" />
```


WHAT IS “THE GRIND”?

Patterns

 View.tsx

```
1 <Button color="primary" />  
2 <Icon color="secondary" />
```

 View.tsx

```
1 <Typography color="#484848" />
```


WHAT IS “THE GRIND”?

Patterns

 View.tsx

```
1 <Button color="primary" />
2 <Icon color="secondary" />
```

 View.tsx

```
1 <Typography color="#484848" />
```

 View.tsx

```
1 <Label color="red" />
```


WHAT IS “THE GRIND”?

Patterns

 View.tsx

```
1 <Button className="self-end" />
2 <Icon className="self-end" />
3 <Typography className="self-end" />
```

 View.tsx

```
1 <div className="self-end">
2   <Label color="red" />
3 </div>
```


WHAT IS “THE GRIND”?

Patterns

 View.tsx

```
1 <Button onClick={ (event: MouseEvent<HTMLElement>) => {} } />
```

 View.tsx

```
1 <Icon onClick={ (event: string) => {} } />
```


WHAT IS “THE GRIND”?

Patterns

🔗 View.tsx

```
1 <Typography color="#484848" />
```



🔗 View.tsx

```
1 <Label color="red" />
```



🔗 View.tsx

```
1 <div className="self-end">
2   <Label color="red" />
3 </div>
```



🔗 View.tsx

```
1 <Icon onClick={(event: string) => {}} />
```



WHAT IS “THE GRIND”?

Patterns

🔗 View.tsx

```
1 <Typography color="#484848" />
```



🔗 View.tsx

```
1 <Label color="red" />
```



🔗 View.tsx

```
1 <div className="self-end">
2   <Label color="red" />
3 </div>
```



🔗 View.tsx

```
1 <Icon onClick={(event: string) => {}} />
```



WHAT IS "THE GRIND"?

Patterns

🔗 View.tsx

```
1 <Typography color="#484848" />
```



🔗 View.tsx

```
1 <Label color="red" />
```



🔗 View.tsx

```
1 <div className="self-eating" />
2   <Label color="red" />
3 </div>
```



🔗 View.tsx

```
1 <Icon onClick={(event: string) => {}} />
```



WHAT IS “THE GRIND”?

Generalization

Make sure this is a valid email

Email

Fredsmith123@gmail.com

WHAT IS “THE GRIND”?

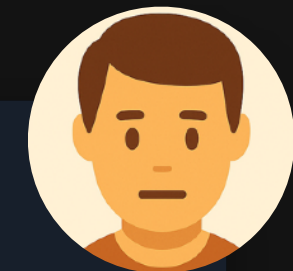
Generalizations

Engine.ts

```
1 class ValidationEngine {
2   constructor() {
3     this.rules = new Map();
4     this.transformers = new Map();
5     this.validators = new Map();
6   }
7
8   registerRule(name, config) {
9     this.rules.set(name, {
10      type: config.type,
11      params: config.params,
12      errorMessage: config.errorMessage,
13      priority: config.priority || 0,
14    });
15  }
16
17  registerTransformer(name, fn) {
18    this.transformers.set(name, fn);
19  }
20
21  registerValidator(type, fn) {
22    this.validators.set(type, fn);
23  }
24
25  async validate(data, ruleNames) {
26    const results = [];
27    const sortedRules = ruleNames
28      .map(name => ({ name, ...this.rules.get(name) }))
29      .sort((a, b) => b.priority - a.priority);
30
31    for (const rule of sortedRules) {
32      const validator = this.validators.get(rule.type);
33      const result = await validator(data, rule.params);
34      results.push({ rule: rule.name, passed: result, message: rule.errorMessage });
35    }
36    return results;
37  }
38 }
```

View.tsx

```
1 const [view, setView] = useState('current');
2
3 // Initialize the complex validation engine
4 const [engine] = useState(() => {
5   const e = new ValidationEngine();
6
7   // Register validators
8   e.registerValidator('minLength', (value, params) => {
9     return value.length >= params.min;
10  });
11
12   e.registerValidator('pattern', (value, params) => {
13     return new RegExp(params.regex).test(value);
14  });
15
16   // Register rules
17   e.registerRule('emailMinLength', {
18     type: 'minLength',
19     params: { min: 5 },
20     errorMessage: 'Email too short',
21     priority: 1
22  });
23
24   e.registerRule('emailPattern', {
25     type: 'pattern',
26     params: { regex: '^[^@]+@[^@]+\.[^@]+$' },
27     errorMessage: 'Invalid email format',
28     priority: 2
29  });
30
31   return e;
32 });
```



Make sure the user's password doesn't
include their email address

Password

Fredsmith123

Hide

WHAT IS “THE GRIND”?

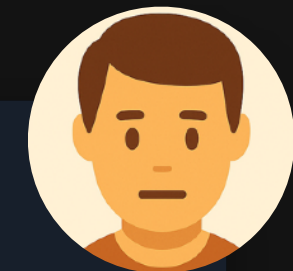
Generalizations

ts Engine.ts

```
1 class ValidationEngine {
2   constructor() {
3     this.rules = new Map();
4     this.transformers = new Map();
5     this.validators = new Map();
6   }
7
8   registerRule(name, config) {
9     this.rules.set(name, {
10      type: config.type,
11      params: config.params,
12      errorMessage: config.errorMessage,
13      priority: config.priority || 0,
14    });
15  }
16
17  registerTransformer(name, fn) {
18    this.transformers.set(name, fn);
19  }
20
21  registerValidator(type, fn) {
22    this.validators.set(type, fn);
23  }
24
25  async validate(data, ruleNames) {
26    const results = [];
27    const sortedRules = ruleNames
28      .map(name => ({ name, ...this.rules.get(name) }))
29      .sort((a, b) => b.priority - a.priority);
30
31    for (const rule of sortedRules) {
32      const validator = this.validators.get(rule.type);
33      const result = await validator(data, rule.params);
34      results.push({ rule: rule.name, passed: result, message: rule.errorMessage });
35    }
36    return results;
37  }
38 }
```

View.tsx

```
1 const [view, setView] = useState('current');
2
3 // Initialize the complex validation engine
4 const [engine] = useState(() => {
5   const e = new ValidationEngine();
6
7   // Register validators
8   e.registerValidator('minLength', (value, params) => {
9     return value.length >= params.min;
10  });
11
12   e.registerValidator('pattern', (value, params) => {
13     return new RegExp(params.regex).test(value);
14  });
15
16   // Register rules
17   e.registerRule('emailMinLength', {
18     type: 'minLength',
19     params: { min: 5 },
20     errorMessage: 'Email too short',
21     priority: 1
22  });
23
24   e.registerRule('emailPattern', {
25     type: 'pattern',
26     params: { regex: '^[^@]+@[^@]+\.[^@]+$' },
27     errorMessage: 'Invalid email format',
28     priority: 2
29  });
30
31   return e;
32 });
```



WHAT IS “THE GRIND”?

Generalizations

Engine.ts

```
1 class ValidationEngine {
2   constructor() {
3     this.rules = new Map();
4     this.transformers = new Map();
5     this.validators = new Map();
6   }
7
8   registerRule(name, config) {
9     this.rules.set(name, {
10      type: config.type,
11      params: config.params,
12      errorMessage: config.errorMessage,
13      priority: config.priority || 0,
14    });
15  }
16
17   registerTransformer(name, fn) {
18     this.transformers.set(name, fn);
19   }
20
21   registerValidator(type, fn) {
22     this.validators.set(type, fn);
23   }
24
25   async validate(data, ruleNames) {
26     const results = [];
27     const sortedRules = ruleNames
28       .map(name => ({ name, ...this.rules.get(name) }))
29       .sort((a, b) => b.priority - a.priority);
30
31     for (const rule of sortedRules) {
32       const validator = this.validators.get(rule.type);
33       const result = await validator(data, rule.params);
34       results.push({ rule: rule.name, passed: result, message: rule.errorMessage });
35     }
36     return results;
37   }
38 }
```

View.tsx

```
1 const [email, setEmail] = useState('');
2 const [validationResults, setValidationResults] = useState([]);
3
4 // CURRENT USE CASE: Just validate an email
5 const validateEmail = async () => {
6   const results = await engine.validate(email, ['emailMinLength', 'emailPattern']);
7   setValidationResults(results);
8 };
9
10 // NEW USE CASE Y: Validate password strength with contextual rules
11 // This DOESN'T FIT our validation engine at all!
12 const [password, setPassword] = useState('');
13 const [passwordIssues, setPasswordIssues] = useState([]);
14
15 const validatePassword = () => {
16   const issues = [];
17
18   // Password needs to be checked against the email
19   if (email && password.toLowerCase().includes(email.split('@')[0].toLowerCase())) {
20     issues.push('Password contains part of your email');
21   }
22
23   // Password strength depends on multiple fields together
24   if (password.length < 12 && ![A-Z]/.test(password) && ![0-9]/.test(password)) {
25     issues.push('Password is too weak - needs length OR complexity');
26   }
27
28   // Needs to check against a blacklist (async operation with external data)
29   const commonPasswords = ['password123', '123456', 'qwerty'];
30   if (commonPasswords.includes(password.toLowerCase())) {
31     issues.push('Password is too common');
32   }
33
34   setPasswordIssues(issues);
35 };
```

```
29 });
30
31 return e;
32 });
```

```
tate('current');
```

```
validation engine
=> {
engine();
```

```
ngth', (value, params) => {
arams.min;
```

```
rn', (value, params) => {
.regex).test(value);
```

```
ngth', {
```

```
short',
```

```
rn', {
```

```
@[^@]+\.[^@]+$' },
mail format',
```



WHAT IS “THE GRIND”?

Generalizations

Engine.ts

```
1 class ValidationEngine {
2   constructor() {
3     this.rules = new Map();
4     this.transformers = new Map();
5     this.validators = new Map();
6   }
7
8   registerRule(name, config) {
9     this.rules.set(name, {
10      type: config.type,
11      params: config.params,
12      errorMessage: config.errorMessage,
13      priority: config.priority || 0,
14    });
15  }
16
17   registerTransformer(name, fn) {
18     this.transformers.set(name, fn);
19   }
20
21   registerValidator(type, fn) {
22     this.validators.set(type, fn);
23   }
24
25   async validate(data, ruleNames) {
26     const results = [];
27     const sortedRules = ruleNames
28       .map(name => ({ name, ...this.rules.get(name) }))
29       .sort((a, b) => b.priority - a.priority);
30
31     for (const rule of sortedRules) {
32       const validator = this.validators.get(rule.type);
33       const result = await validator(data, rule.params);
34       results.push({ rule: rule.name, passed: result, message: rule.errorMessage });
35     }
36     return results;
37   }
38 }
```

View.tsx

```
1 const [email, setEmail] = useState('');
2 const [validationResults, setValidationResults] = useState([]);
3
4 // CURRENT USE CASE: validate email
5 const validateEmail = async (email) => {
6   const results = await engine.validate(email, ['email']);
7   setValidationResults(results);
8 };
9
10 // NEW USE CASE: validate password
11 // This DOES NOT require a transformer
12 const [password, setPassword] = useState('');
13 const [passwordIssues, setPasswordIssues] = useState([]);
14
15 const validatePassword = async (password) => {
16   const results = await engine.validate(password, ['password']);
17   const issues = results.filter(r => !r.passed).map(r => r.message);
18   setPasswordIssues(issues);
19 };
20
21 // Needs to check against a list of common passwords (external data)
22 const commonPasswords = ['password', '123456', 'qwerty', '1qaz!@WSX'];
23
24 if (commonPasswords.includes(password.toLowerCase())) {
25   issues.push('Password is too common');
26 }
27
28 // Needs to check against a list of common passwords (external data)
29 const commonPasswords = ['password', '123456', 'qwerty', '1qaz!@WSX'];
30 if (commonPasswords.includes(password.toLowerCase())) {
31   issues.push('Password is too common');
32 }
33
34 setPasswordIssues(issues);
35 };
```



```
state('current');
```

```
validation engine
=> {
  engine();
}
```

```
length', (value, params) => {
  params.min;
```

```
length', (value, params) => {
  regex).test(value);
```

```
length', {
```

```
short',
```

```
length', {
```

```
@['@']+\.[^@]+$' },
mail format',
```

```
29 });
```

```
30
```

```
31 return e;
```

```
32 });
```





== Assumptions

Avoiding  == Avoiding Assumptions

How do we avoid assumptions?

- ☐ Prevent assumptions
- ☐ Manage broken assumptions
- ☐ Resist assumptions

How do we prevent assumptions?



How do we prevent assumptions?

We get more information



What is a Design System?

A set of rules that dictate the
behaviour of your UI



PREVENT ASSUMPTIONS

Component

Space 100

Space 200

Space 300

Space 400

Space 600

Red circle

Yellow circle

Green circle

Grey circle

White circle

Black circle

Button

Button

Button

Title

Body text

Info icon

Refresh icon

Search icon

Trash icon

Send icon

Shopping cart icon

Name

Value

Surname

Value

Email

Value

Mobile Number

Value

Message

Value

Submit

Heading 1

Heading 2

Heading 3

Heading 4

Body Base

Tag x

Title

Body text for whatever you'd like to say.

Button

Title

Body text.

Button

Products

Solutions

Contact

Link

Sign in

Register

Label

Label

Avatar 1

Avatar 2

Avatar 3

Title

Answer the frequently asked question in a simple sentence.

PREVENT ASSUMPTIONS

Semantic Colors		Search			
All variables	10	Name	Light	Dark	+
Colors					
Foreground	Colors / Foreground				
Surface					

PREVENT ASSUMPTIONS

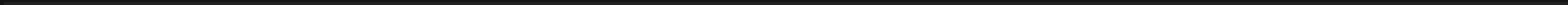
Token Layers

Primitive Token

green-600

blue-600

pink-600



PREVENT ASSUMPTIONS

Token Layers

Primitive Token

green-600

blue-600

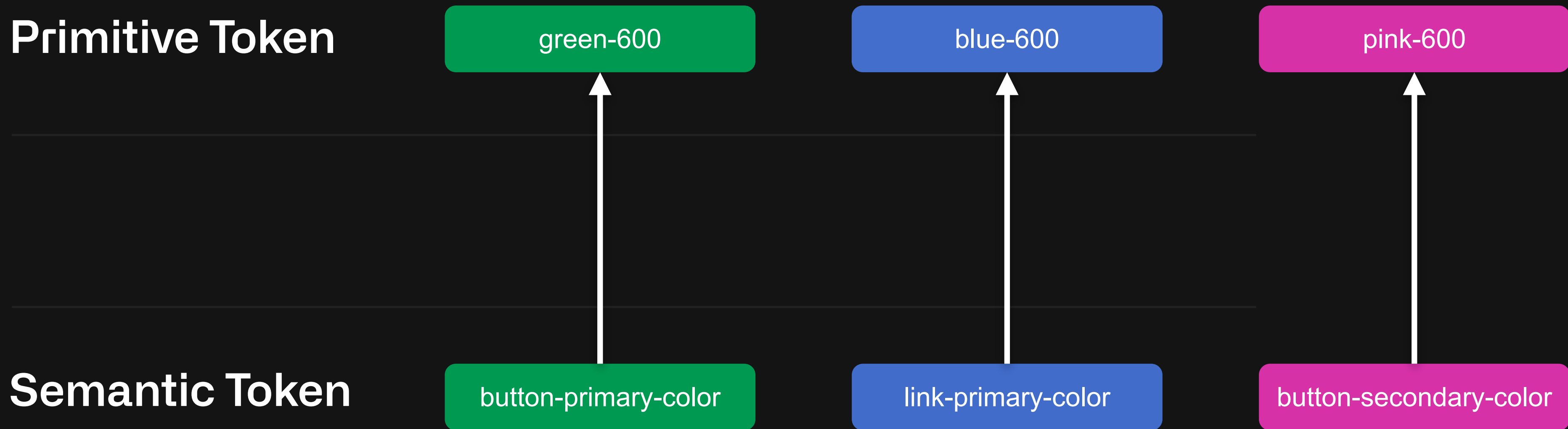
pink-600

Semantic Token

button-primary-color

link-primary-color

button-secondary-color



PREVENT ASSUMPTIONS

Token Layers

Primitive Token

green-600

blue-600

pink-600

System Token

color-primary

text-primary

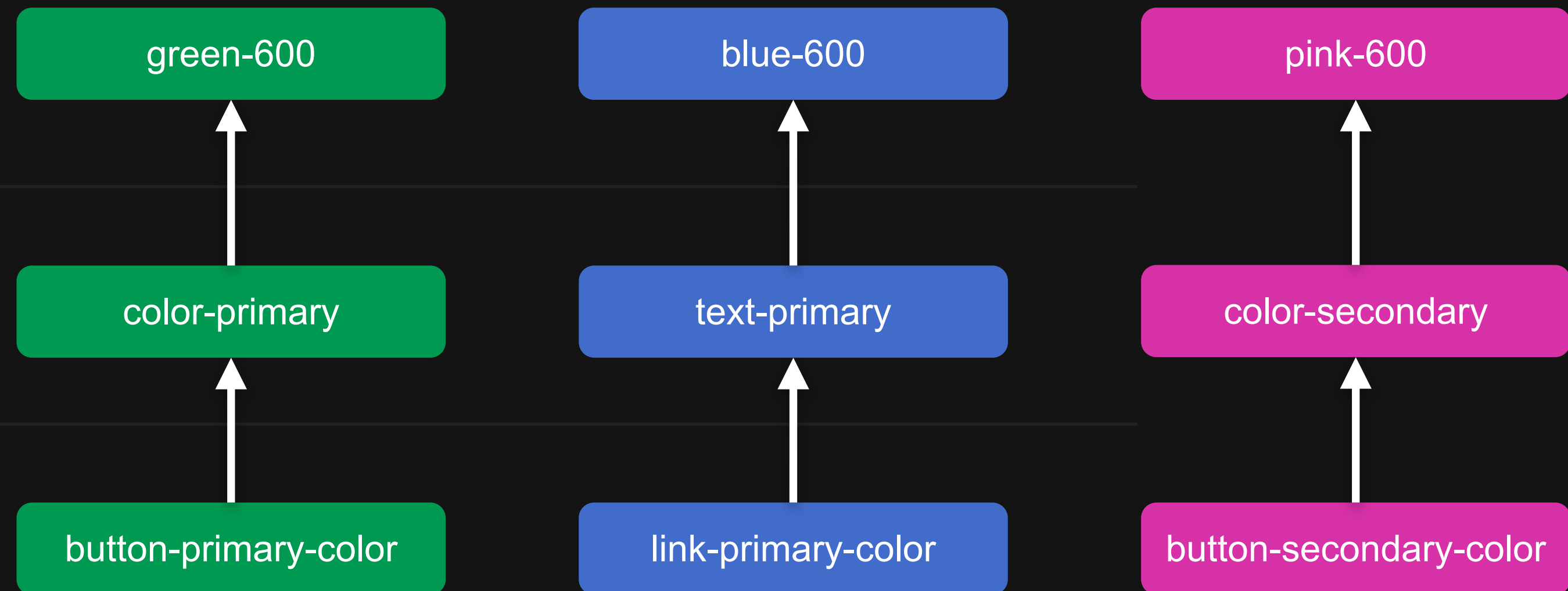
color-secondary

Semantic Token

button-primary-color

link-primary-color

button-secondary-color



PREVENT ASSUMPTIONS

Token Layers

Primitive Token

green-600

blue-600

pink-600

System Token

color-primary

text-primary

color-secondary

Semantic Token

button-primary-color

link-primary-color

button-secondary-color



PREVENT ASSUMPTIONS

Token Layers

Primitive Token

green-600

blue-600

pink-600

System Token

color-primary

text-primary

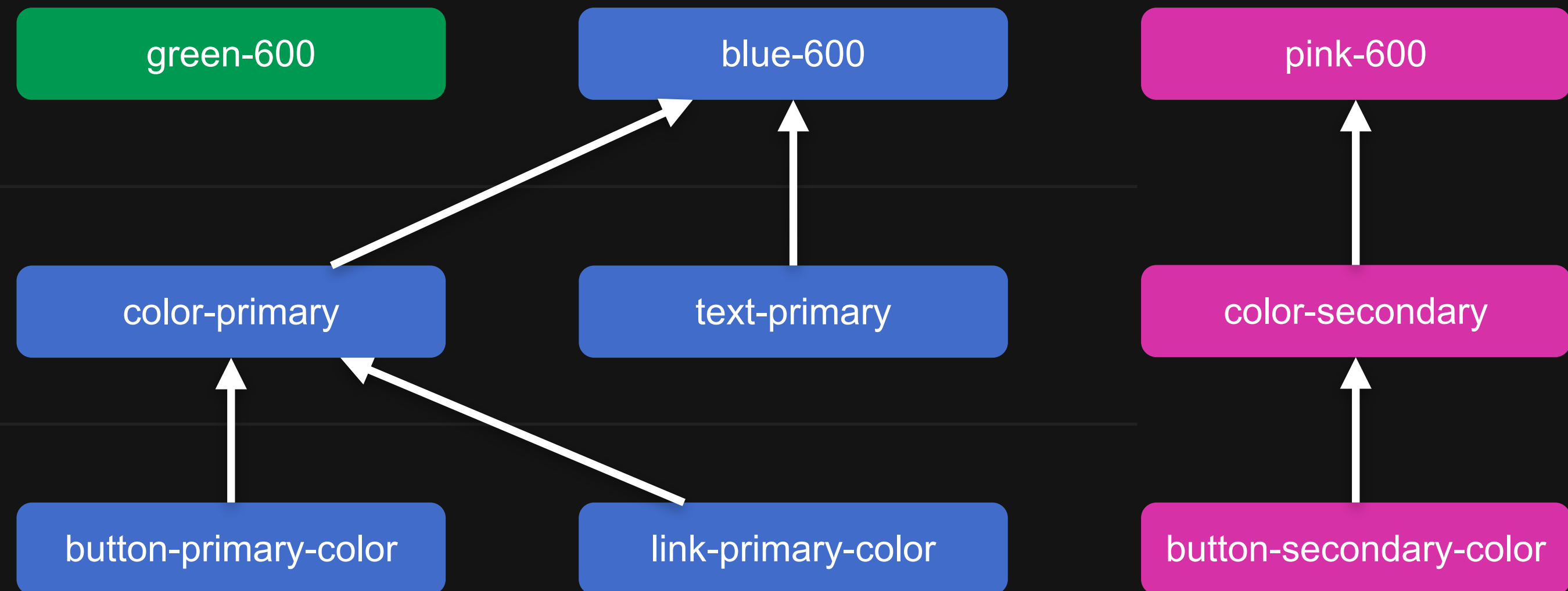
color-secondary

Semantic Token

button-primary-color

link-primary-color

button-secondary-color



Benefits of a design system

- A single source of truth
- Ensure consistency of the product
- Hierarchy of reusable components
- Designers can focus on interesting problems
- Developers can focus on interesting problems



How do we avoid assumptions?

- ☒ Prevent assumptions
- ☐ Manage broken assumptions
- ☐ Resist assumptions

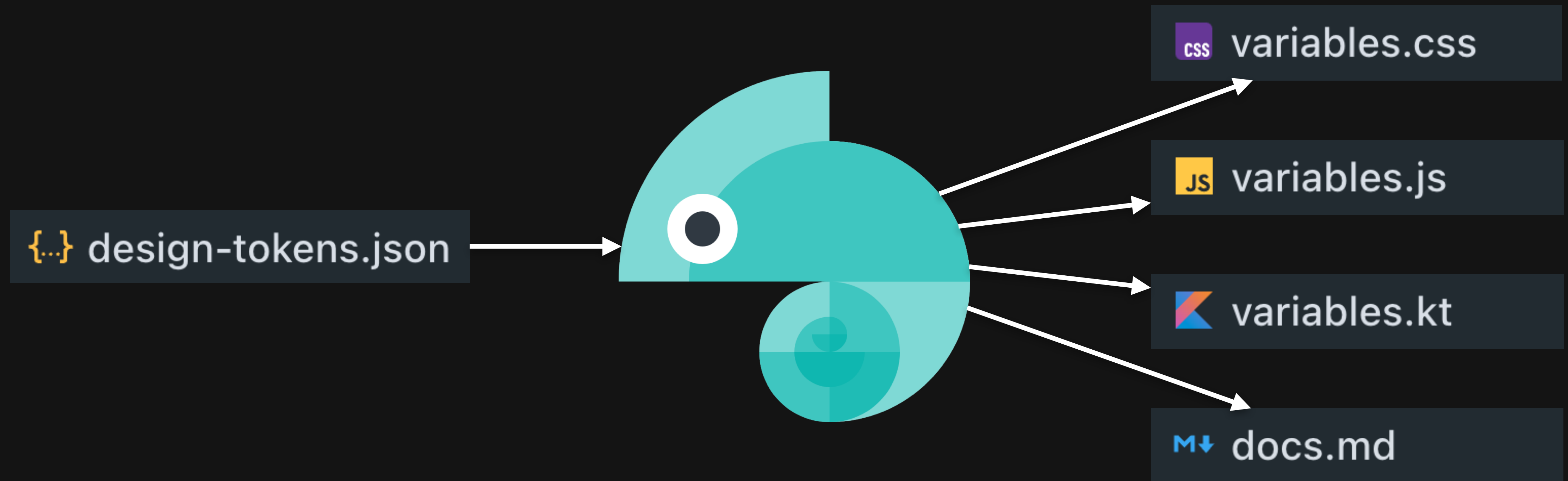
HANDLE CHANGES

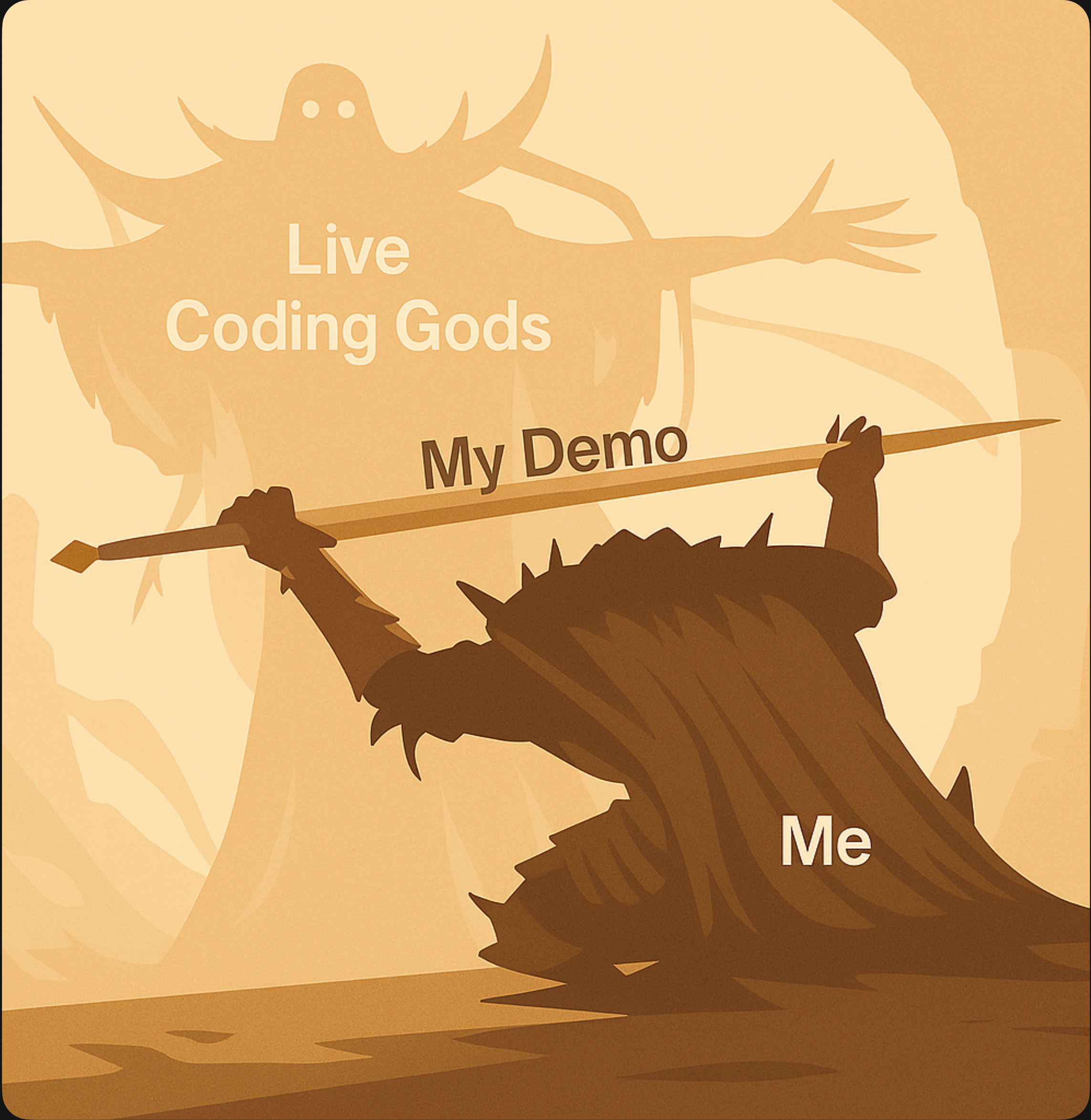
 design-tokens.json

```
1 {
2   "color": {
3     "background": {
4       "brand": {
5         "default": {
6           "description": "Bg/Brand/Default",
7           "type": "color",
8           "value": "{color primitives.brand.800}"
9         },
10        "secondary": {...},
11        "tertiary": {...},
12      }
13    },
14    "color primitives": {
15      "brand": {
16        "800": {
17          "type": "color",
18          "value": "#2c2c2cff",
19          "blendMode": "normal"
20        }
21      }
22    },
23    "size": {
24      "space": {
25        "100": {
26          "type": "dimension",
27          "value": 4,
28        }
29      }
30    }
31  }
```


HANDLE CHANGES

Style Dictionary





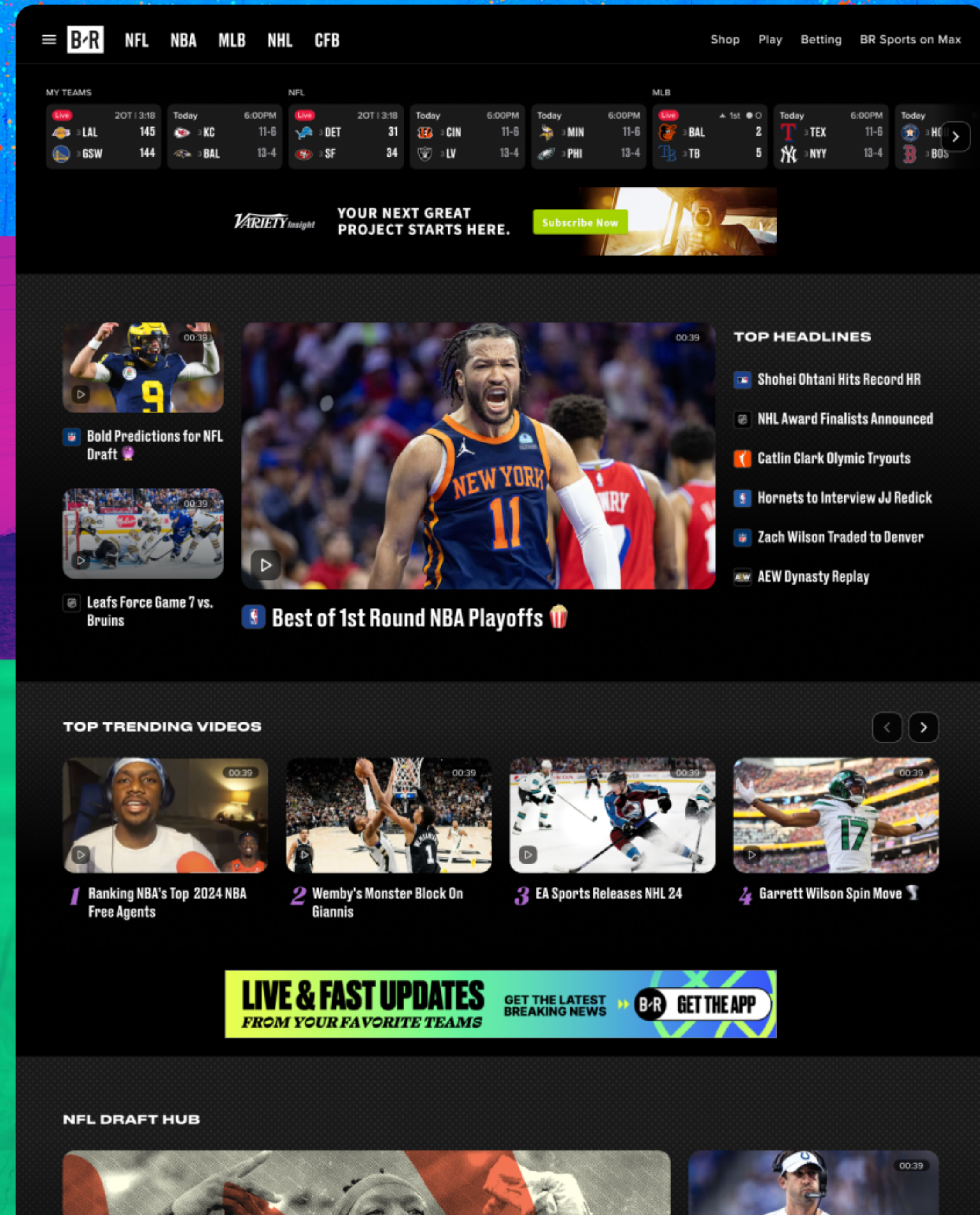
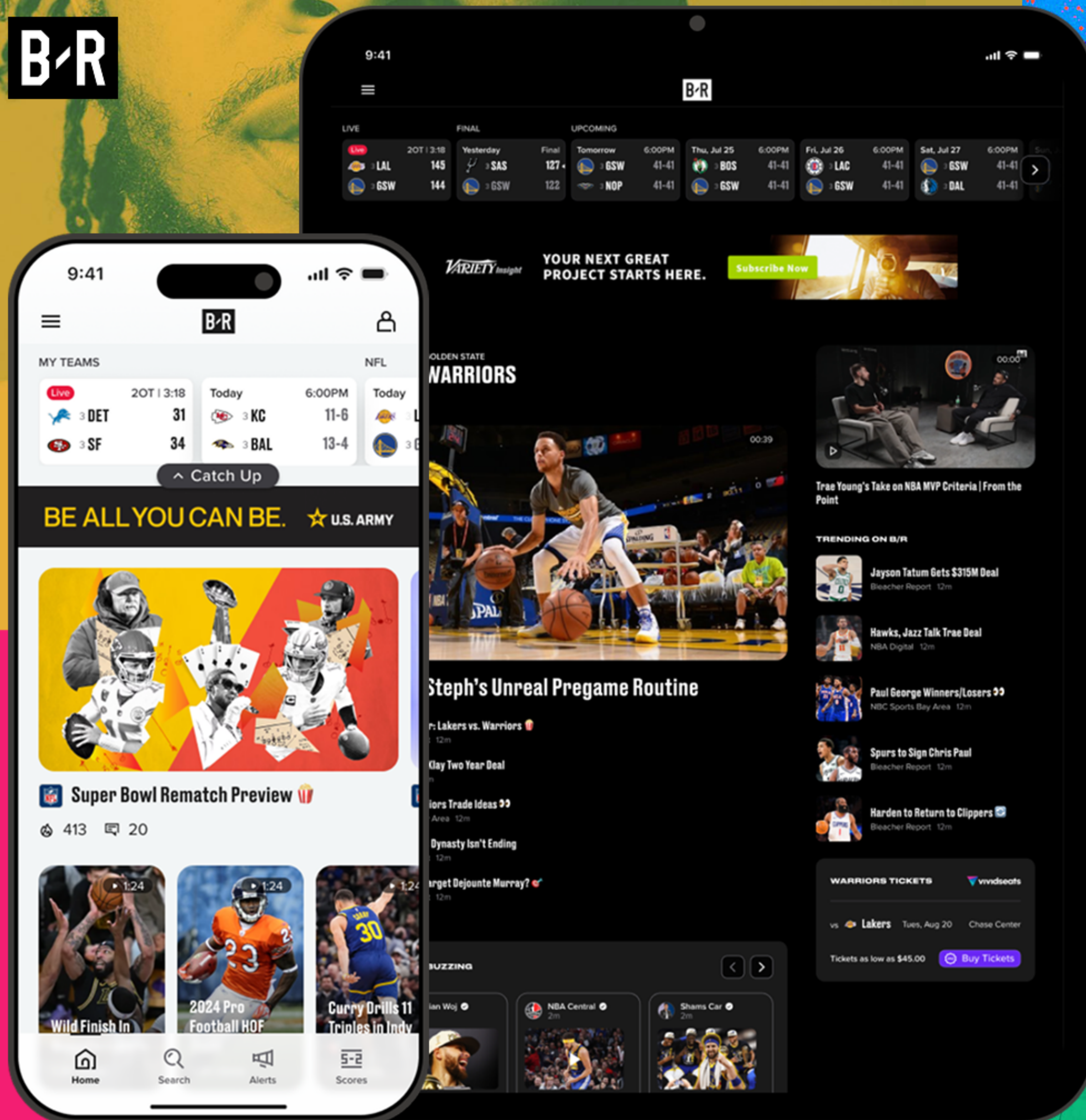
Live
Coding Gods

My Demo

Me

...But Sarah, wasn't that in itself





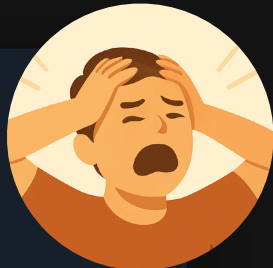
How do we avoid assumptions?

- ☒ Prevent assumptions
- ☒ Manage broken assumptions
- ☐ Resist assumptions

Variables

css variables.css

```
1 :root {
2   --light: #ffffff;
3   --dark: #1A1E23;
4   --teal: #12F7D6;
5   --teal-light: #98FAEC;
6   --background: #292F36;
7 }
```



<C/> SinanTokmak

Home Blogs

Instagram Discord Github

☰

👤

<>

📄

✍️

✉️

Sinan

Full-stack developer

✉️ abdurrahman_sinan@hotmail.com

📍 Turkey

🏠 Full-time / Freelancer

🌐 www.sinantokmak.com

HTML CSS JS REACT

Download CV ⬇️

<h1>

Hey I'm Sinan, Full-Stack Developer

</h1>

<p>

I help business grow by crafting amazing web experiences. If you're looking for a developer that likes to get stuff done,

</p>

Let's Talk ✉️

4 Programming Language

6 Development Tools

8 Years of Experience

About Me

Hello!

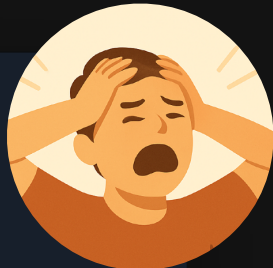
My name is Sinan and I specialize in web development that utilizes HTML, CSS, JS, and REACT etc. I am a highly motivated individual and eternal optimist dedicated to writing clear, concise, robust code that works. Striving to never stop learning and improving. When I'm not coding, I am writing bolgs, reading, or picking up some new hands-on art project like photography. I like to have my perspective and belief systems challenged so that I see the world through new eyes.



Variables

CSS variables.css

```
1 :root {
2   --background-1: #292F36;
3   --background-2: #1A1E23;
4   --brand-1: #12F7D6;
5   --brand-2: #98FAEC;
6   --button-fill: #43454D;
7   --text: #ffffff;
8   --text-on-dark: #292F36;
9 }
```



<C/> SinanTokmak

Home Blogs

Instagram Discord Github

Sinan
Full-stack developer

abdurrahman_sinan@hotmail.com

Turkey

Full-time / Freelancer

www.sinantokmak.com

HTML CSS JS REACT

Download CV

Hey
I'm Sinan,
Full-Stack Developer

I help business grow by crafting amazing web experiences. If you're looking for a developer that likes to get stuff done,

Let's Talk

4 Programming Language

6 Development Tools

8 Years of Experience

About Me

Hello!

My name is Sinan and I specialize in web development that utilizes HTML, CSS, JS, and REACT etc.

I am a highly motivated individual and eternal optimist dedicated to writing clear, concise, robust code that works. Striving to never stop learning and improving. When I'm not coding, I am writing bolgs, reading, or picking up some new hands-on art project like photography.

I like to have my perspective and belief systems challenged so that I see the world through new eyes.

Variables

CSS variables.css

```
1 :root {
2   --background-1: #292F36;
3   --background-2: #1A1E23;
4   --brand-1: #12F7D6;
5   --brand-2: #98FAEC;
6   --button-fill: #43454D;
7   --text: #ffffff;
8   --text-on-dark: #292F36;
9 }
```



Never rely on primitive tokens in your stylesheets. To rely on primitives is to rely on assumptions.

<C/> SinanTokmak

HomeBlogs

InstagramDiscordGithub

Sinan

Full-stack developer

abdurrahman_sinan@hotmail.com

Turkey

Full-time / Freelancer

www.sinantokmak.com

HTMLCSSJSREACT

Download CV

Developer

Hey

I'm Sinan,

Full-Stack Developer

I help business grow by crafting amazing web experiences. If you're looking for a developer that likes to get stuff done,

Let's Talk

4

Programming Language

6

Development Tools

8

Years of Experience

0

Hello!

My name is Sinan and I specialize in web development that utilizes HTML, CSS, JS, and REACT etc.

I am a highly motivated individual and eternal optimist dedicated to writing clear, concise, robust code that works. Striving to never stop learning and improving.

When I'm not coding, I am writing bolgs, reading, or picking up some new hands-on art project like photography.

I like to have my perspective and belief systems challenged so that I see the world through new eyes.

RESIST ASSUMPTIONS

Variables

CSS variables.css

```
1 :root {  
2   --background-1: #292F36;  
3   --background-2: #1A1E23;  
4   --brand-1: #12F7D6;  
5   --brand-2: #98FAEC;  
6   --button-fill: #43454D;  
7   --text: #ffffff;  
8   --text-on-dark: #292F36;  
9 }
```



Never rely on primitive tokens in your stylesheets. To rely on primitives is to rely on assumptions.



<C/> SinanTokmak

Home Blogs

Instagram

Discord

Github

Developer

Hey
I'm Sinan,
Full-Stack Developer

crafting amazing web experiences. If you're
that likes to get stuff done,

41k

4 Programming
Language

6 Development
Tools

8 Years of
Experience

<p>

Hello!

My name is Sinan and I specialize in web development that utilizes HTML, CSS, JS, and REACT etc.
I am a highly motivated individual and eternal optimist dedicated to writing clear, concise, robust code that works. Striving to never stop learning and improving.
When I'm not coding, I am writing blogs, reading, or picking up some new hands-on art project like photography.
I like to have my perspective and belief systems challenged so that I see the world through new eyes.

</p>

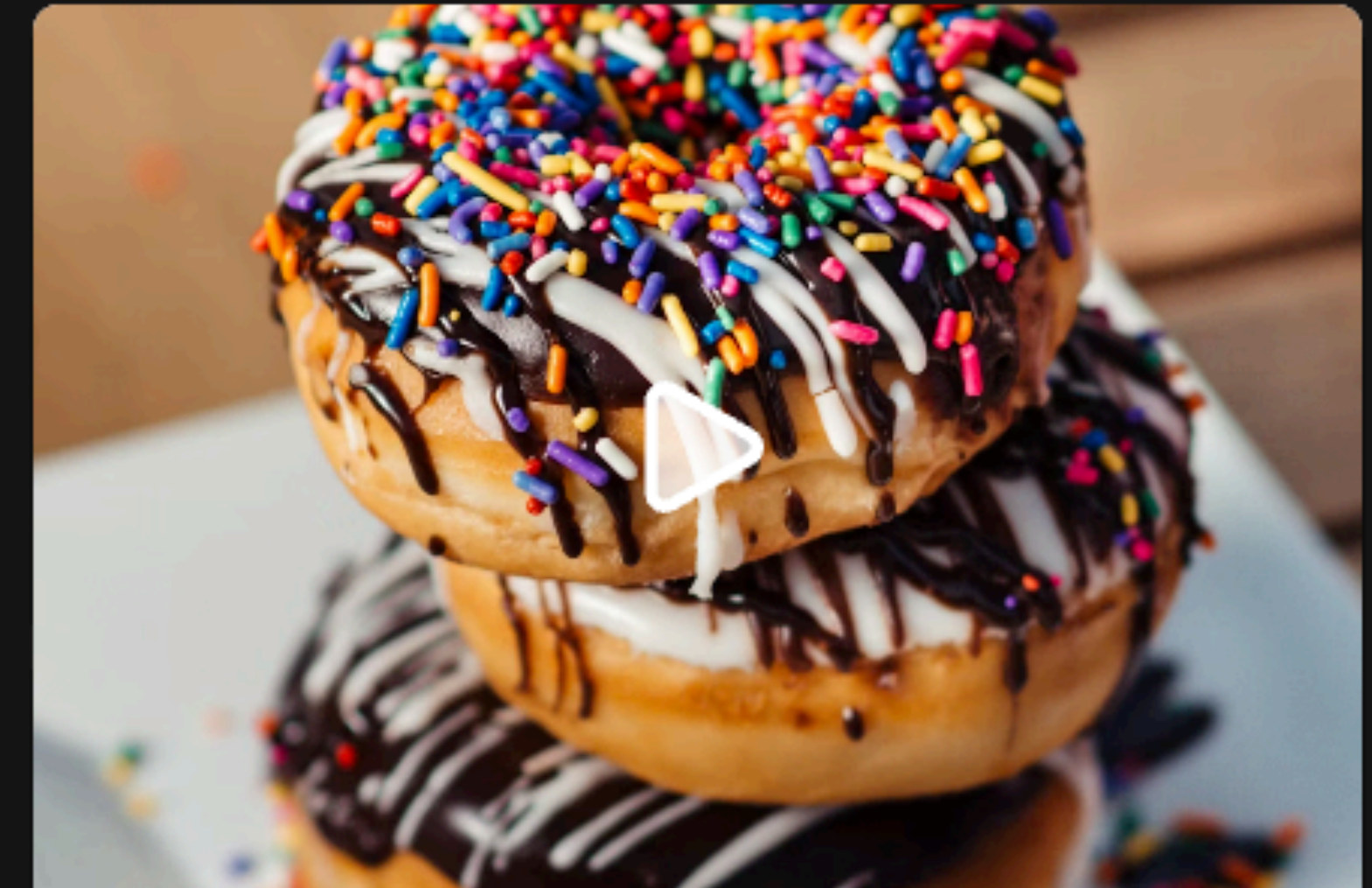
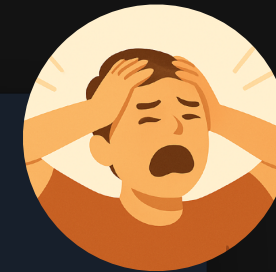


RESIST ASSUMPTIONS

Monoliths

🔗 Card.tsx

```
1 <Card
2   title="Donuts"
3   description="A donut is more than..."
4   thumbnailUrl="https://unsplash.com/image.jpg"
5   link="https://buy.now"
6   videoUrl="https://unsplash.com/video.mp4"
7   secondaryLink="https://nutritional.info"
8   stars="4.5"
9 />
```



Donuts ★ ★ ★ ★ ★

A donut is more than just a pastry, it is a wish fulfilled

Buy Now

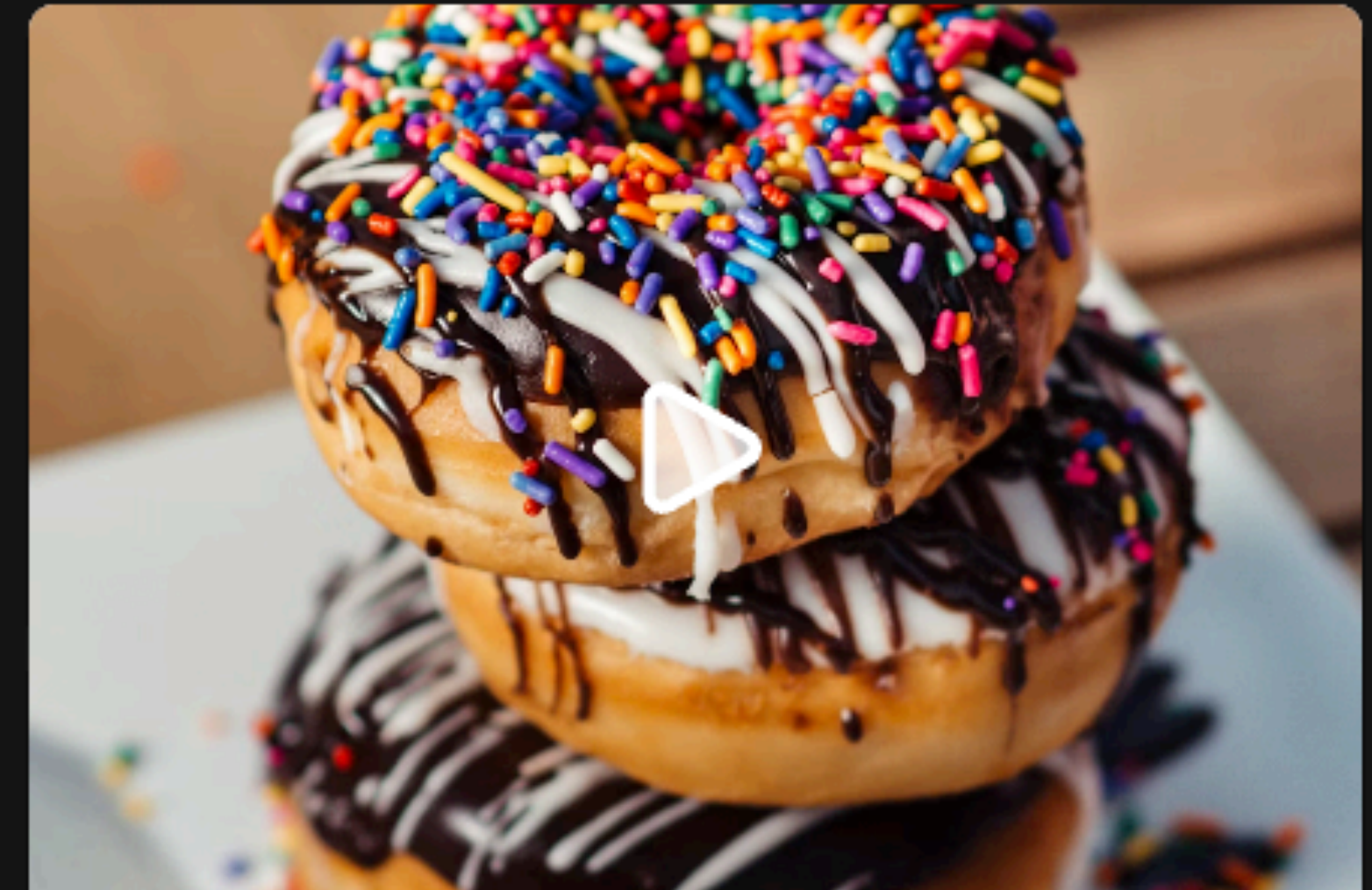
Nutritional Info

RESIST ASSUMPTIONS

Monoliths

Card.tsx

```
1 <Card
2   media={<VideoMedia url="https://unsplash.com/video.mp4" />}
3   title={
4     <Typography>Burgers</Typography>
5     <Stars rating="4.5" />
6   }
7   description="A donut is more than..."
8   buttons={
9     <Button>Buy Now</Button>
10    <Button>Nutritional Info</Button>
11  }
12 />
```



Donuts ★ ★ ★ ★ ★

A donut is more than just a pastry, it is a wish fulfilled

Buy Now

Nutritional Info

RESIST ASSUMPTIONS

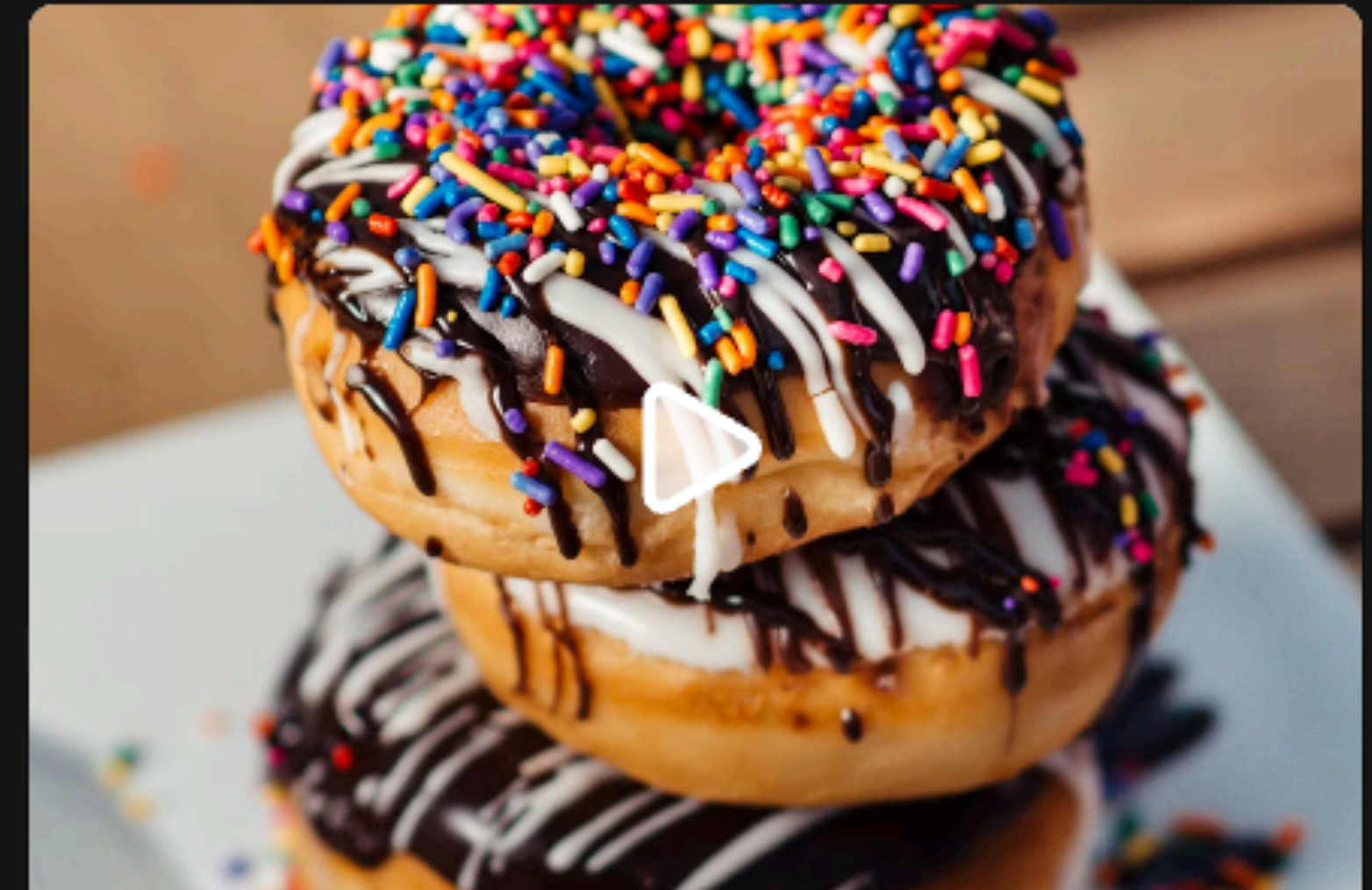
Monoliths

Card.tsx

```
1 <Card
2   media={<VideoMedia url="https://unsplash.com/video.mp4" />}
3   title={
4     <Typography>Burgers</Typography>
5     <Stars rating="4.5" />
6   }
7   description="A donut is more than..."
8   buttons={
9     <Button>Buy Now</Button>
10    <Button>Nutritional Info</Button>
11  }
12 />
```



Focus on creating small pieces that do one thing and do it well.
These can be composed together to create rich experiences.



Donuts ★ ★ ★ ★ ★

A donut is more than just a pastry, it is a wish fulfilled

Buy Now

Nutritional Info

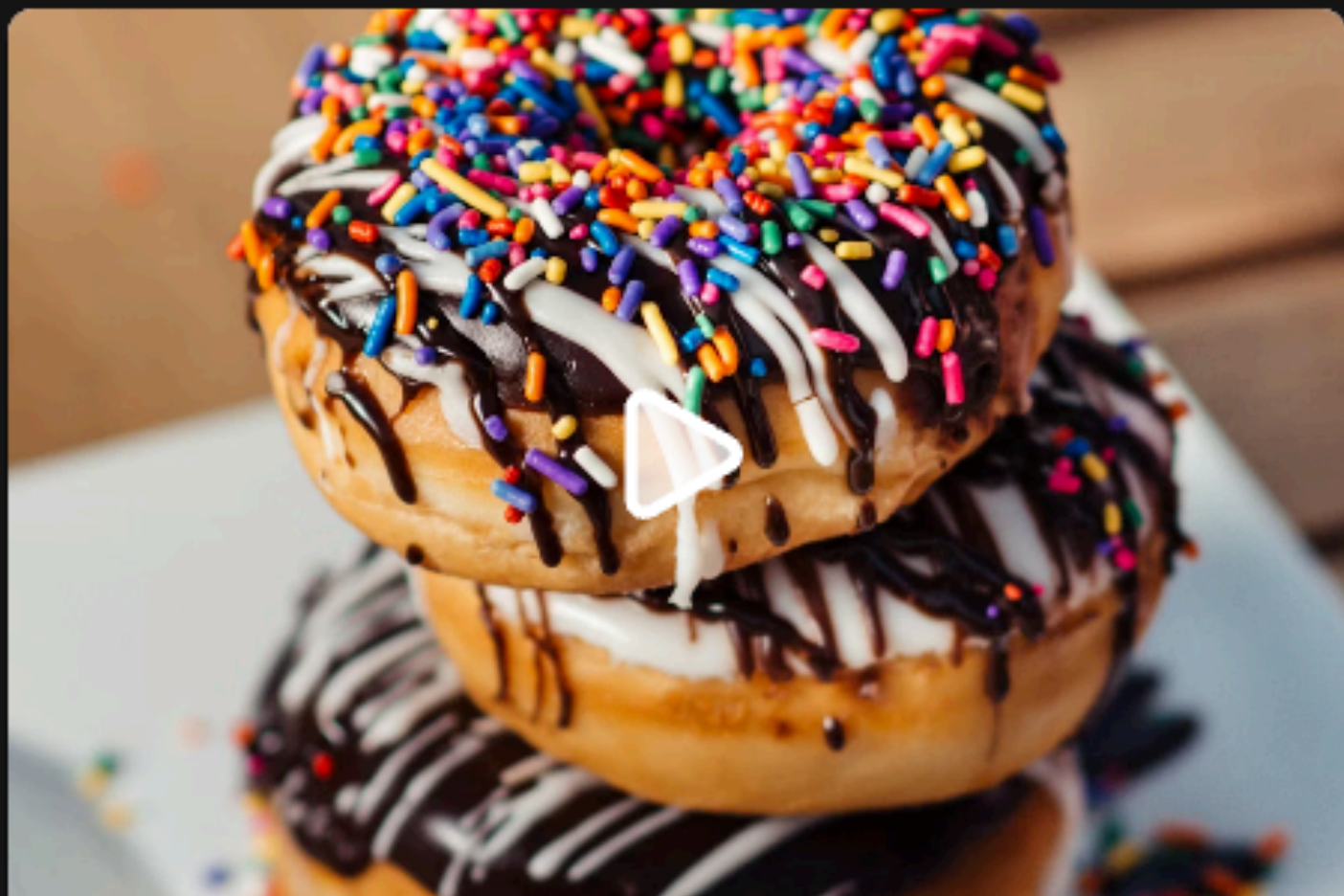


Premium Donuts

★★★★★ (128)

\$79 ~~\$129~~

 Add to Cart



Vanilla

Chocolate

Boston Cream

Donuts ★★★★★

A donut is more than just a pastry, it is a wish



DM

Donut Man

I Make Donuts

1.2K
Followers

432
Following

89
Posts

Follow

Message

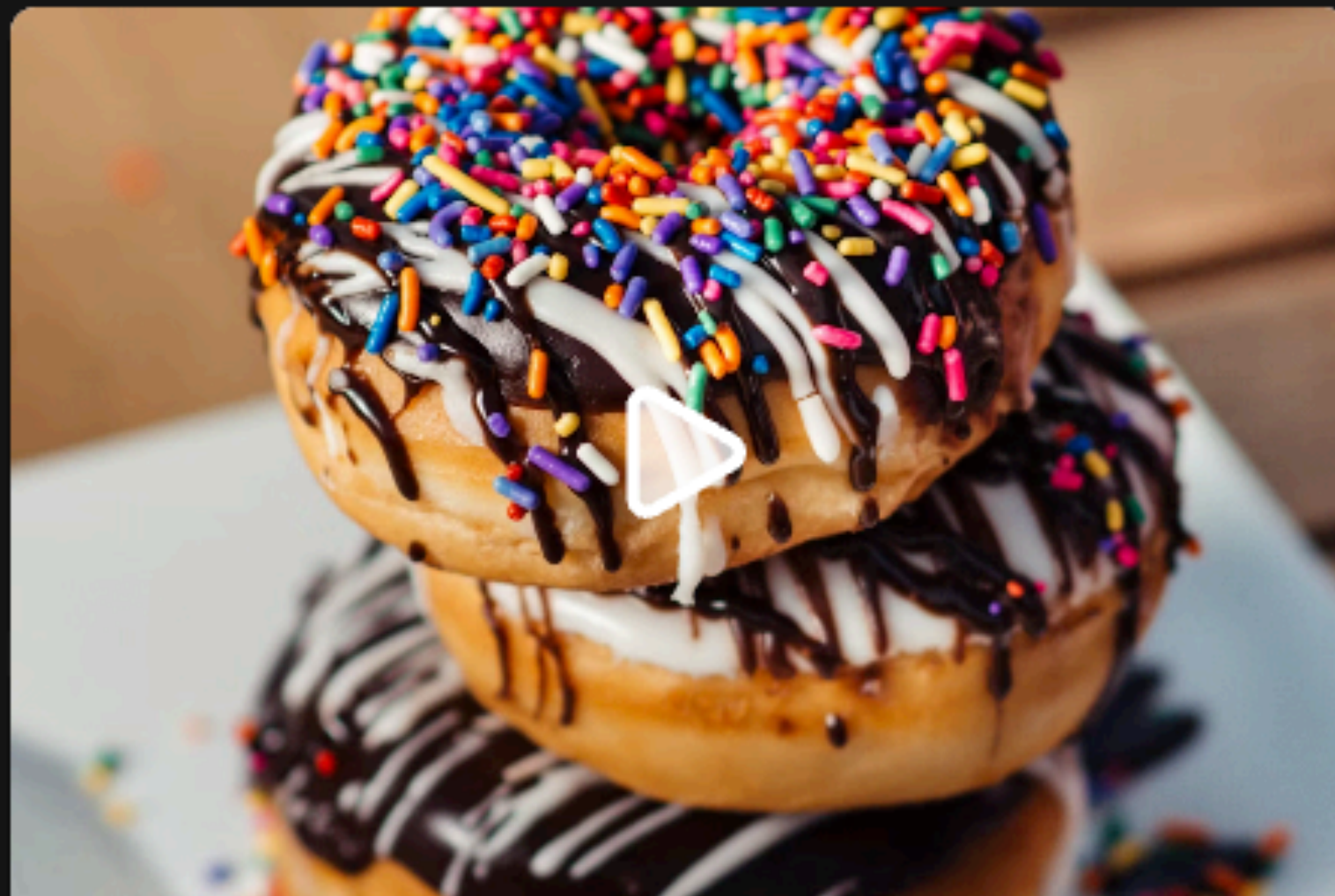
Donuts

A donut is more than just a pastry, it is a wish fulfilled

Learn More

A donut is more than just a pastry, it is a wish fulfilled

 24  8 



Donuts ★★★★★

A donut is more than just a pastry, it is a wish fulfilled

Buy Now

Nutritional Info



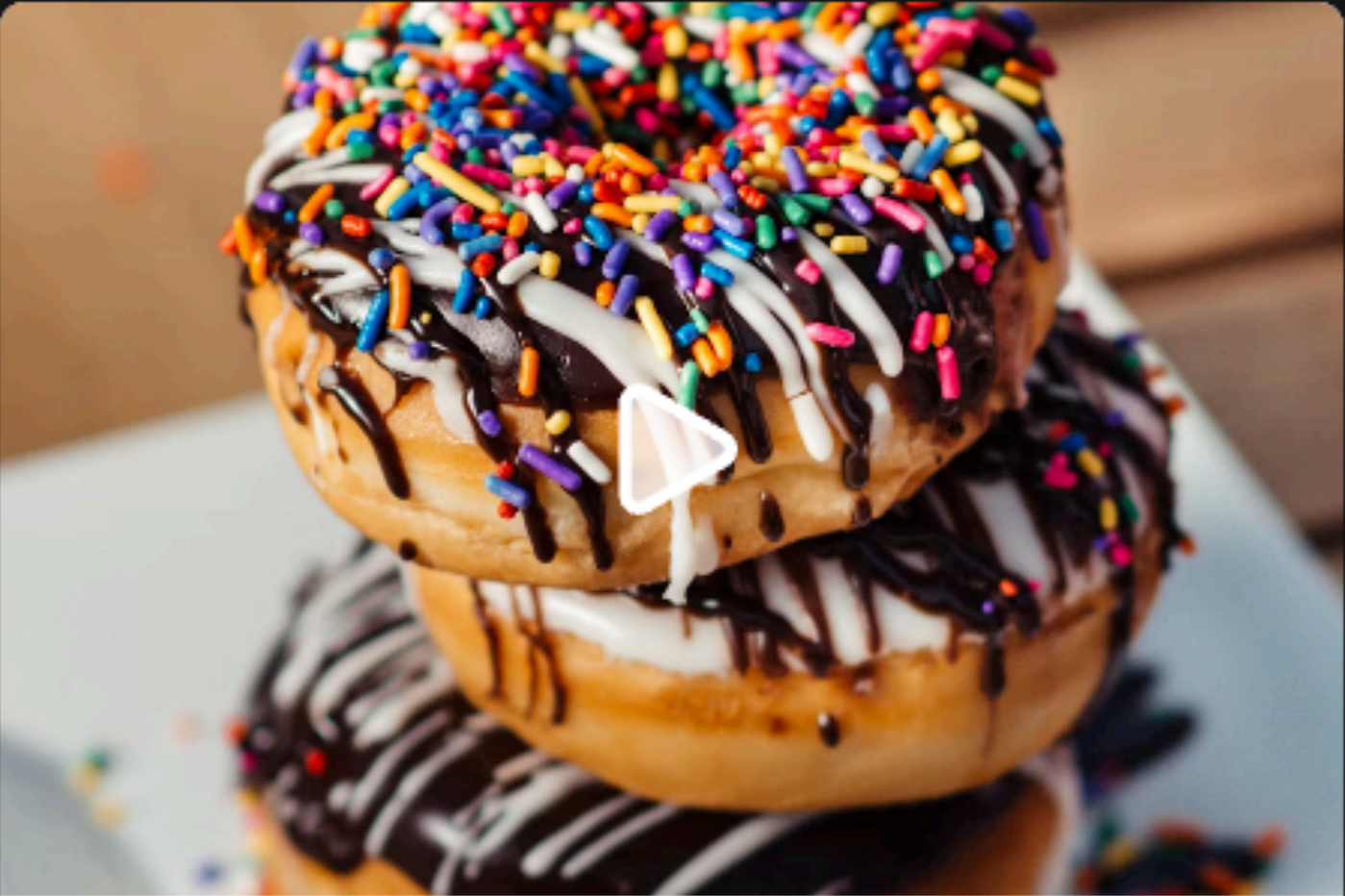


Premium Donuts

★★★★★ (128)

\$79 ~~\$129~~

 Add to Cart



- Vanilla
- Chocolate
- Boston Cream

Donuts ★★★★★

A donut is more than just a pastry, it is a wish



DM



Donuts

A donut is more than just a pastry, it is a wish fulfilled

Learn More

A donut is more than just a pastry, it is a wish fulfilled

 24  8 



Donuts ★★★★★

A donut is more than just a pastry, it is a wish fulfilled

[Buy Now](#) [Nutritional Info](#)



HTML Semantics



🔗 Button.tsx

```
1 export const Button = ({ label, onClick, href }) => {  
2   if (href) {  
3     return <a className="button-style" href={href}>{label}</a>  
4   }  
5  
6   return (  
7     <button className="button-styles" onClick={onClick}>  
8       {label}  
9     </button>  
10  )  
11 }
```

HTML Semantics

🔗 Button.tsx

```
1 export const Button = ({ component, label, ...componentProps }) => {  
2   const Component = component || "button"  
3  
4   return (  
5     <Component  
6       className="button-styles"  
7       {...componentProps}  
8     >  
9       {label}  
10    </Component>  
11  )  
12 }
```



🔗 View.tsx

```
1 <Button component="a" href="https://some.link">Go to Cart</Button>  
2 <Button onClick={addToCart}>Add to Cart</Button>
```


HTML Semantics

🔗 Button.tsx

```
1 export const Button = ({ component, label, ...componentProps }) => {  
2   const Component = component || "button"  
3  
4   return (  
5     <Component  
6       className="button-styles"  
7       {...componentProps}  
8     >  
9       {label}  
10    </Component>  
11  )  
12 }
```



Composition is an easy way to apply the display of one component to the behaviour of another.

🔗 View.tsx

```
1 <Button component="a" href="https://some.link">Go to Cart</Button>  
2 <Button onClick={addToCart}>Add to Cart</Button>
```

HTML Semantics

🔗 Button.tsx

```
1 export const Button = ({ component, ...componentProps }) => {  
2   const Component = component  
3  
4   return (  
5     <Component  
6       className="button"  
7       {...componentProps}  
8     >  
9       {label}  
10    </Component>  
11  )  
12 }
```



an easy way to apply the display of one
the behaviour of another.

🔗 View.tsx

```
1 <Button component="a" href="https://some.link">Go to Cart</Button>  
2 <Button onClick={addToCart}>Add to Cart</Button>
```


Patterns

🔗 View.tsx

```
1 <Typography color="#484848" />
```



🔗 View.tsx

```
1 <Label color="red" />
```



🔗 View.tsx

```
1 <div className="self-end">
2   <Label color="red" />
3 </div>
```



🔗 View.tsx

```
1 <Icon onClick={(event: string) => {}} />
```



Patterns

🔗 View.tsx

```
1 <Typography color="primary" />
```



🔗 View.tsx

```
1 <Label color="secondary" />
```



🔗 View.tsx

```
1 <Label color="primary" className="self-end" />  
2  
3
```



🔗 View.tsx

```
1 <Icon onClick={(event: MouseEvent<HTMLElement>) => {}} />
```



Patterns

 View.tsx

```
1 <Typography color="primary" />
```

 View.tsx

```
1 <Label color="secondary" />
```

 View.tsx

```
1 <Label color="primary" className="self-end" />  
2  
3
```



Code is read way more often than it's written. If we get it right once, we benefit from it hundreds of times.

 View.tsx

```
1 <Icon onClick={(event: MouseEvent<HTMLElement>) => {}} />
```

Patterns

🔗 View.tsx

```
1 <Typography color="primary" />
```

🔗 View.tsx

```
1 <Label color="secondary">
```

🔗 View.tsx

```
1 <Label color="primary">
2
3
```

🔗 View.tsx

```
1 <Icon onClick={(event: MouseEvent<HTMLElement>) => {}} />
```



Code is read way more often than it's written. If we get it right once, we benefit from it hundreds of times.

RESIST ASSUMPTIONS

Generalizations

Engine.ts

```
1 class ValidationEngine {
2   constructor() {
3     this.rules = new Map();
4     this.transformers = new Map();
5     this.validators = new Map();
6   }
7
8   registerRule(name, config) {
9     this.rules.set(name, {
10      type: config.type,
11      params: config.params,
12      errorMessage: config.errorMessage,
13      priority: config.priority || 0,
14    });
15  }
16
17   registerTransformer(name, fn) {
18     this.transformers.set(name, fn);
19   }
20
21   registerValidator(type, fn) {
22     this.validators.set(type, fn);
23   }
24
25   async validate(data, ruleNames) {
26     const results = [];
27     const sortedRules = ruleNames
28       .map(name => ({ name, ...this.rules.get(name) }))
29       .sort((a, b) => b.priority - a.priority);
30
31     for (const rule of sortedRules) {
32       const validator = this.validators.get(rule.type);
33       const result = await validator(data);
34       results.push({ rule: rule.name, passed: result, message: rule.errorMessage });
35     }
36     return results;
37   }
38 }
```

View.tsx

```
1 const [email, setEmail] = useState('');
2 const [validationResults, setValidationResults] = useState([]);
3
4 // CURRENT USE CASE: Just validate an email
5 const validateEmail = async () => {
6   const results = await engine.validate(email, ['emailMinLength', 'emailPattern']);
7   setValidationResults(results);
8 };
9
10 // NEW USE CASE Y: Validate password strength with contextual rules
11 // This DOESN'T FIT our validation engine at all!
12 const [password, setPassword] = useState('');
13 const [passwordIssues, setPasswordIssues] = useState([]);
14
15 const validatePassword = () => {
16   const issues = [];
17
18   // Password needs to be checked against the email
19   if (email && password.toLowerCase().includes(email.split('@')[0].toLowerCase())) {
20     issues.push('Password contains part of your email');
21   }
22
23   // Password strength depends on multiple fields together
24   if (password.length < 12 && ![A-Z]/.test(password) && ![0-9]/.test(password)) {
25     issues.push('Password is too weak - needs length OR complexity');
26   }
27
28   // Needs to check against a blocklist (async operation with external data)
29   const commonPasswords = ['password123', '123456', 'qwerty'];
30   if (commonPasswords.includes(password.toLowerCase())) {
31     issues.push('Password is too common');
32   }
33
34   setPasswordIssues(issues);
35 };
```

```
29 });
30
31 return e;
32 });
```

```
tate('current');
```

```
validation engine
=> {
engine();
```

```
ngth', (value, params) => {
arams.min;
```

```
rn', (value, params) => {
.regex).test(value);
```

```
ngth', {
```

```
short',
```

```
rn', {
```

```
@[^@]+\.[^@]+$' },
mail format',
```



Generalizations

🔗 View.tsx

```
1 const validateEmail = (email) => {  
2   return email.includes('@') && email.length ≥ 5  
3 }  
4  
5 const validatePassword = (password, email) => {  
6   return password.includes(email.split('@')[0])  
7 }
```



Generalizations

 View.tsx

```
1 const validateEmail = (email) => {  
2   return email.includes('@') && email.length ≥ 5  
3 }  
4  
5 const validatePassword = (password, email) => {  
6   return password.includes(email.split('@')[0])  
7 }
```



Never generalize because you're worried about the next use case.
Only generalize on the use cases you know.

Generalizations

🔗 View.tsx

```
1 const validateEmail
2   return email.incl
3 }
4
5 const validateP
6   return pass
7 }
```



Generalizations make your code more complicated. And complicated code takes longer to write, and much longer to debug.

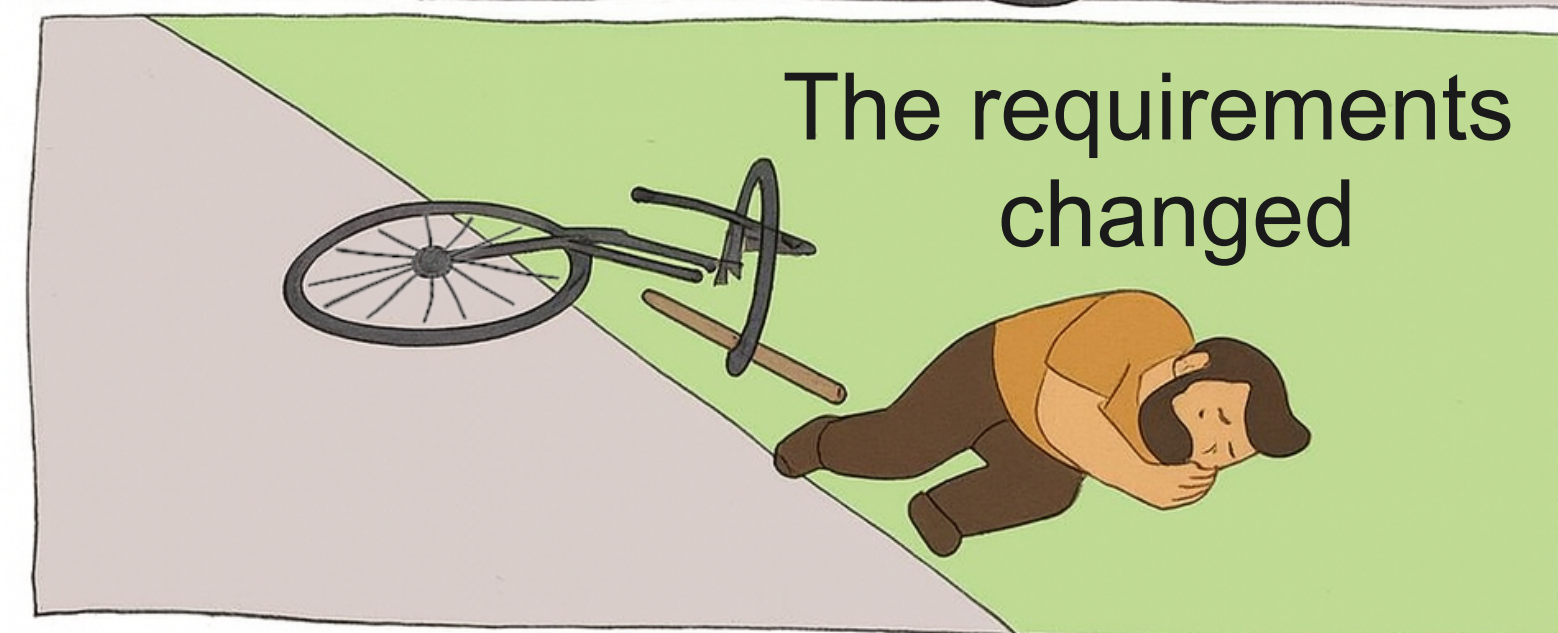
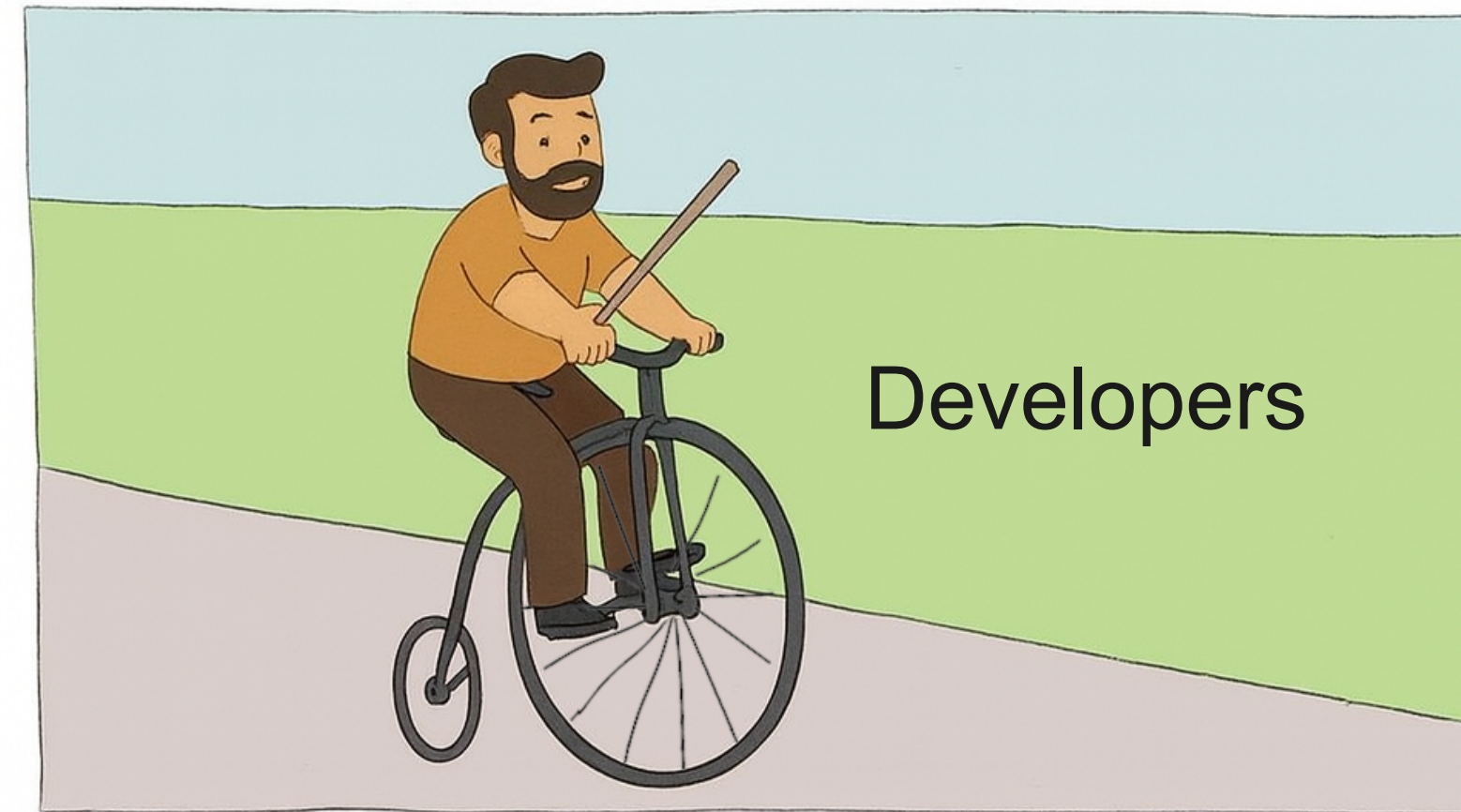
How do we avoid assumptions?

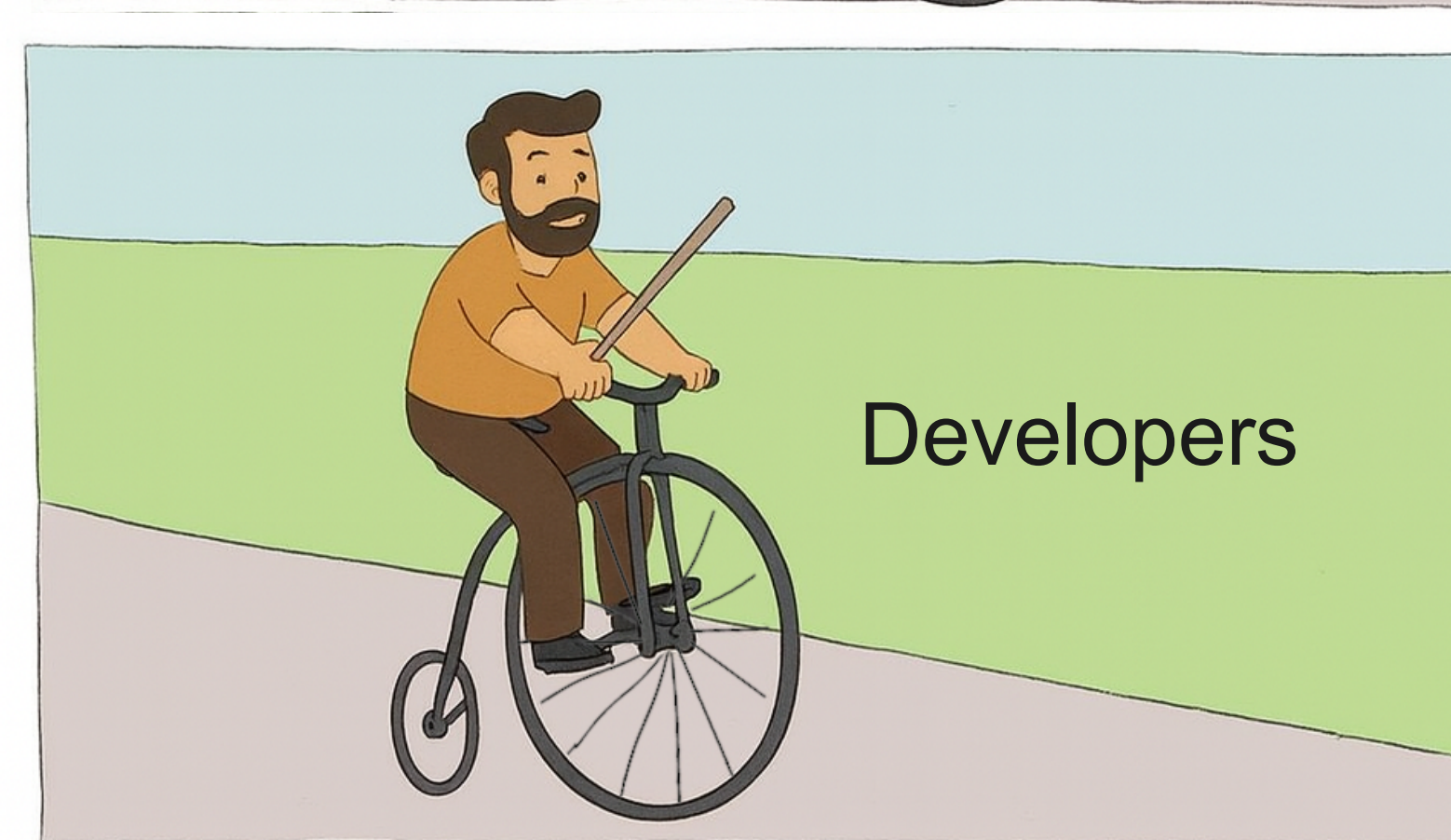
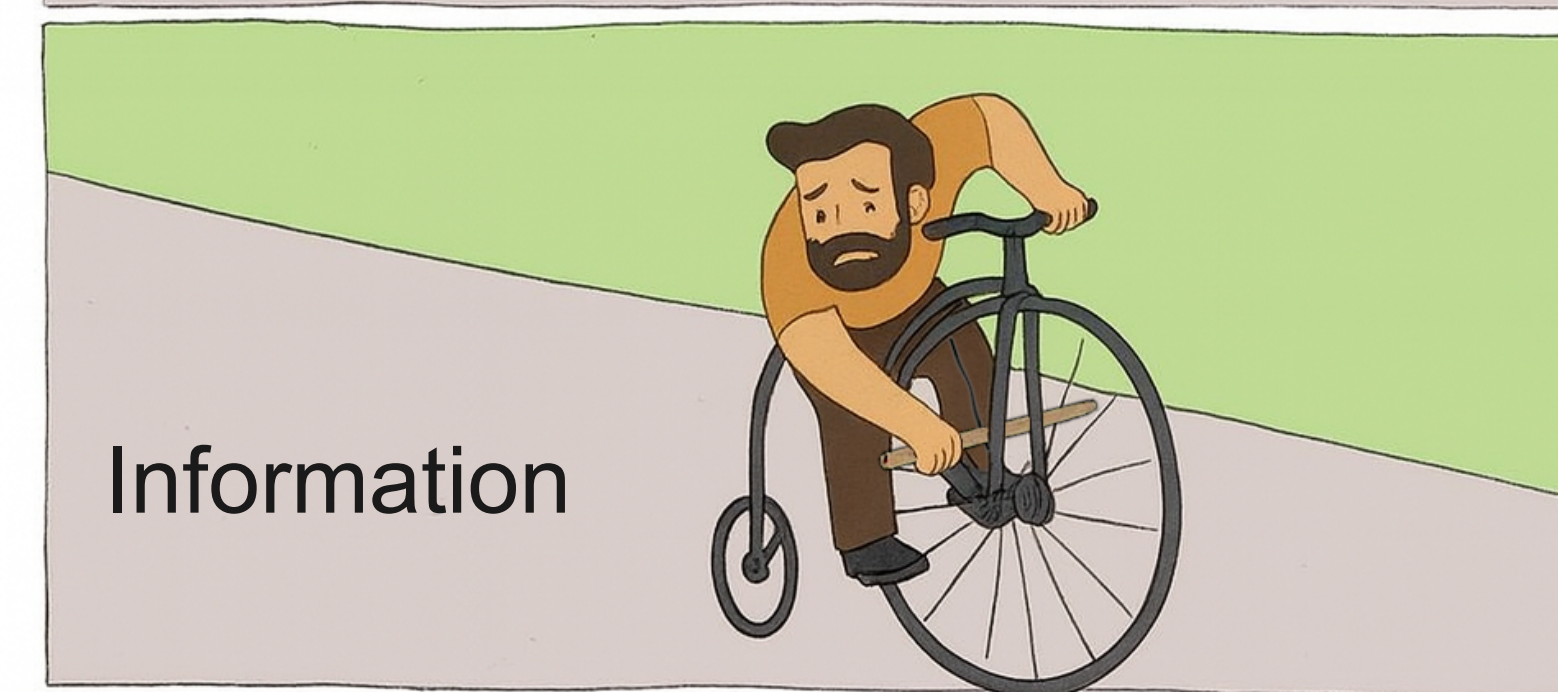
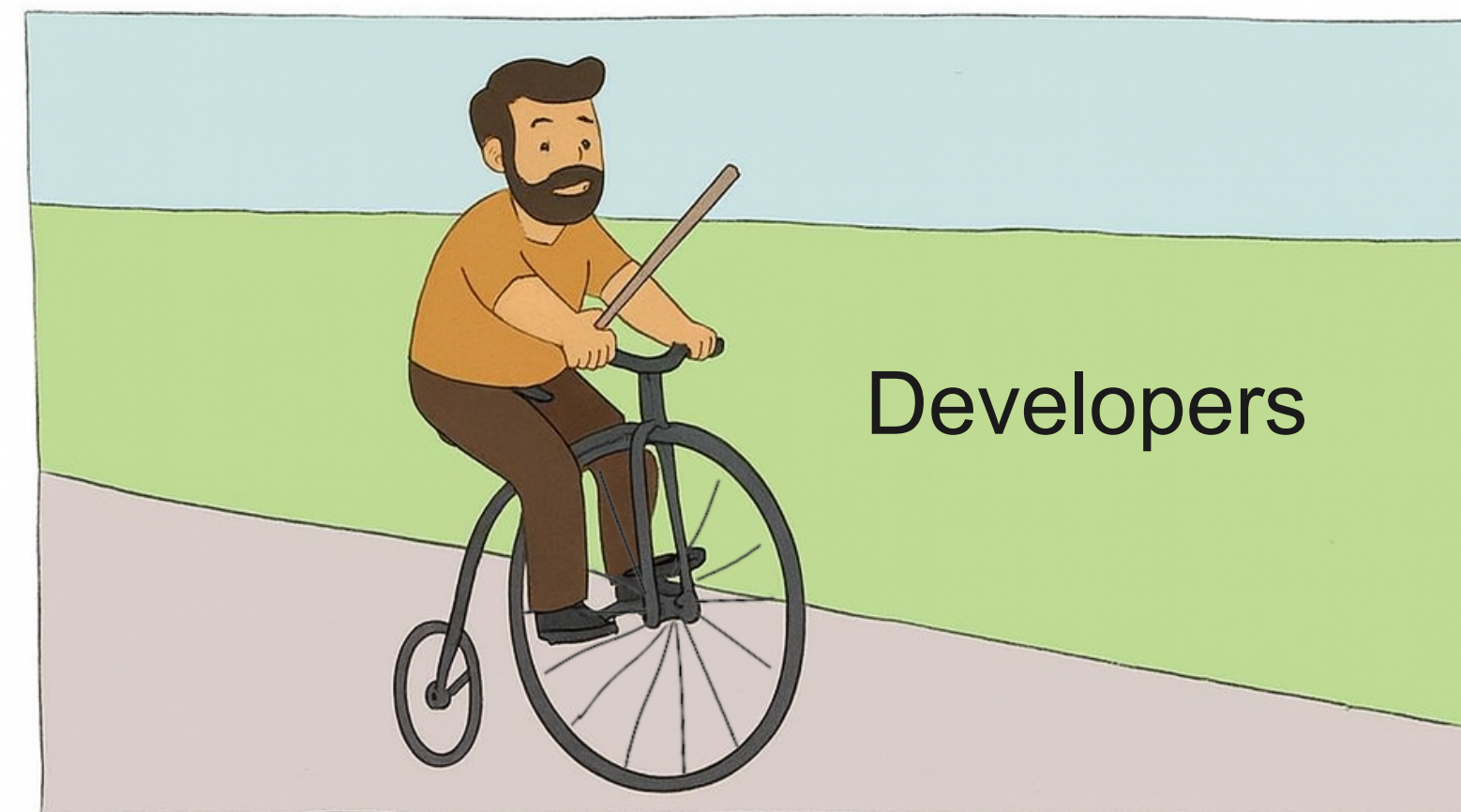
- ✓ Prevent assumptions
- ✓ Manage broken assumptions
- ✓ Resist assumptions

How do we avoid assumptions?

- ✓ Prevent assumptions
- ✓ Manage broken assumptions
- ✓ Resist assumptions









Thank You

Sarah Padovani, Web Team Lead

sarah.padovani@redspace.com

<https://github.com/sarahpads>