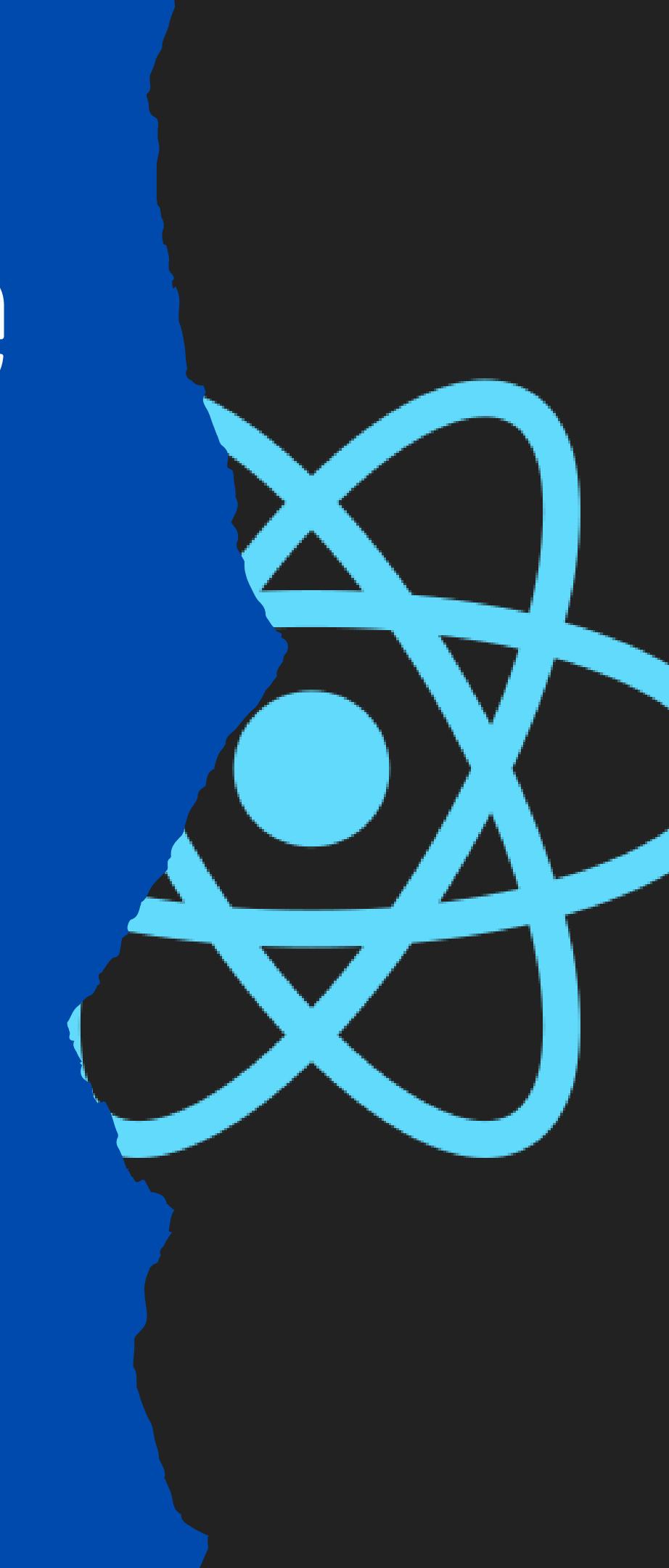


# Building Considerate React Component APIs with TypeScript



Presented by Daria Caraway

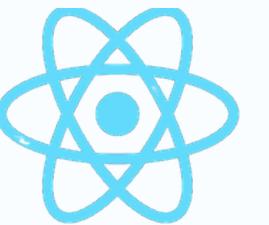
Software Engineer



Boulder, CO USA



Working With



# Daria Caraway



@dariacaraway





# Considerate React Component APIs



What... is .. that...



# Bark Park: A Pet Social Media Company

A place to find friends for your pets.

@dariacaraway

# The Product MVP Goal

# Current State - Edit Profile Page

BARK PARK

Explore Play Dates <sup>2</sup> Park Finder



## Edit Profile



**Mia is a very good girl. She loves long hikes and chasing bunnies. Mia's favorite food is beef jerky treats and hamburger. Her favorite season is Winter. During the first snow of the season, you will find her rolling around outside having a great time.**

Mia 🐶

Species:

Dog



[Change Photo](#)

[Edit Bio](#)

[Change Color](#)

[Save](#)

@dariacaraway

# Meet Lisa



## New Hire at **BARK PARK**

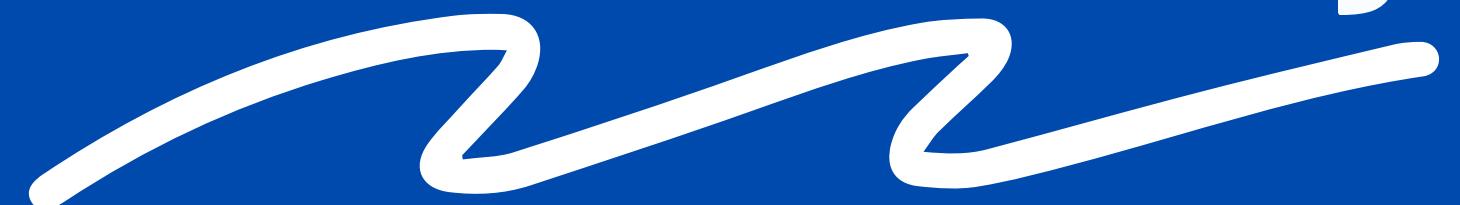
Starting a new role on the web app team

## JavaScript Developer

Never worked with TypeScript



# Lisa's First Day



Exploring the Product

Web Team

@dariacaraway

# Onboarding

Lisa's first ticket



**UI-100**

Card contents are switched

The contents of the profile card are rendering in the wrong place. The body is rendering in the header, and the header in the body.

**Mia is a very good girl. She loves long hikes and chasing bunnies. Mia's favorite food is beef jerky treats and hamburger. Her favorite season is Winter. During the first snow of the season, you will find her rolling around outside having a great time.**

Mia 🐾

# Card - The Consumer



EditProfile.jsx

```
<span css={cardContainer}>  
  <Card input1={bodyText} input2={headerText} />  
</span>
```

# Card - The Consumer



EditProfile.jsx

```
<span css={cardContainer}>  
  <Card input1={bodyText} input2={headerText} />  
</span>
```



INPUT1? INPUT2?



# Card - The Component



Card.jsx

```
function Card({input1, input2}) {  
  return (  
    <article css={cardContainer}>  
      <h1 css={cardHeading}>{input1}</h1>  
      <p css={CardBody}>{input2}</p>  
    </article>  
  )  
}
```

# Card - The Component



Card.jsx

```
function Card({input1, input2}) {
  return (
    <article css={cardContainer}>
      <h1 css={cardHeading}>{input1}</h1>
      <p css={CardBody}>{input2}</p>
    </article>
  )
}
```

Hidden Context: **input1 is the header text, input2 is the body text**

[Code](#) [Issues](#) [Pull requests 4](#) [Actions](#) [Projects](#) [Security](#) [Insights](#) [Settings](#)

[UI100-fix-body...](#) [10 branches](#) [0 tags](#) [Go to file](#) [Add file](#) [Code](#)

This branch is 2 commits ahead of master. [Pull request](#) [Compare](#)

darcar31 fixed	03cd547 7 days ago	13 commits
public	(feat): Explore page	7 days ago
src	fixed	7 days ago
.gitignore	steps	6 months ago
README.md	base	7 days ago
package.json	(feat): Explore page	7 days ago
tsconfig.json	Initialize project using Create React App	9 months ago
yarn.lock	(feat): Explore page	7 days ago

[README.md](#)

## Bark Park Main Application

This is the main repo for the Bark Park web application.

### Running the App

[About](#) [Readme](#)

[Releases](#) [Create a new release](#)

[Packages](#) [Publish your first package](#)

[Languages](#)

TypeScript 92.4% JavaScript 3.3%  
HTML 2.8% CSS 1.5%

# Card - Opening a PR

@dariacaraway

## (bug): Card - Fix swapped header and body #3

Open

Changes from 1 commit ▾

File filter... ▾

Clear filters Jump to... ▾



6 6 src/components/card/Card.jsx

```
24     font-size: 18px;  
25       
26  
27 - function Card({input1, input2}) {  
28     return (  
29         <article css={cardContainer}>  
30 -         <h1 css={cardHeading}>{input1}</h1>  
31 -         <p css={CardBody}>{input2}</p>  
32     </article>  
33 )  
34 }
```

....

Review changes ▾

# Card - The Pull Request

```
24     font-size: 18px;  
25       
26  
27 + function Card({headerText, bodyText}) {  
28     return (  
29         <article css={cardContainer}>  
30 +         <h1 css={cardHeading}>{headerText}</h1>  
31 +         <p css={CardBody}>{bodyText}</p>  
32     </article>  
33 )  
34 }
```

@dariacaraway

### (bug): Card - Fix swapped header and body #3

Open

Changes from 1 commit ▾

File filter... ▾

Clear filters Jump to... ▾



6 6 src/components/card/Card.jsx

24     font-size: 18px;

25     `

## Hidden context.

27 - function Card({input1, input2}) {

28     return (

29       <article css={cardContainer}>

30 -       <h1 css={cardHeading}>{input1}</h1>

31 -       <p css={CardBody}>{input2}</p>

32     </article>

33     )

34 }

....

Review changes ▾

24     font-size: 18px;

25     `

## Baked-in context.

27 + function Card({headerText, bodyText}) {

28     return (

29       <article css={cardContainer}>

30 +       <h1 css={cardHeading}>{headerText}</h1>

31 +       <p css={CardBody}>{bodyText}</p>

32     </article>

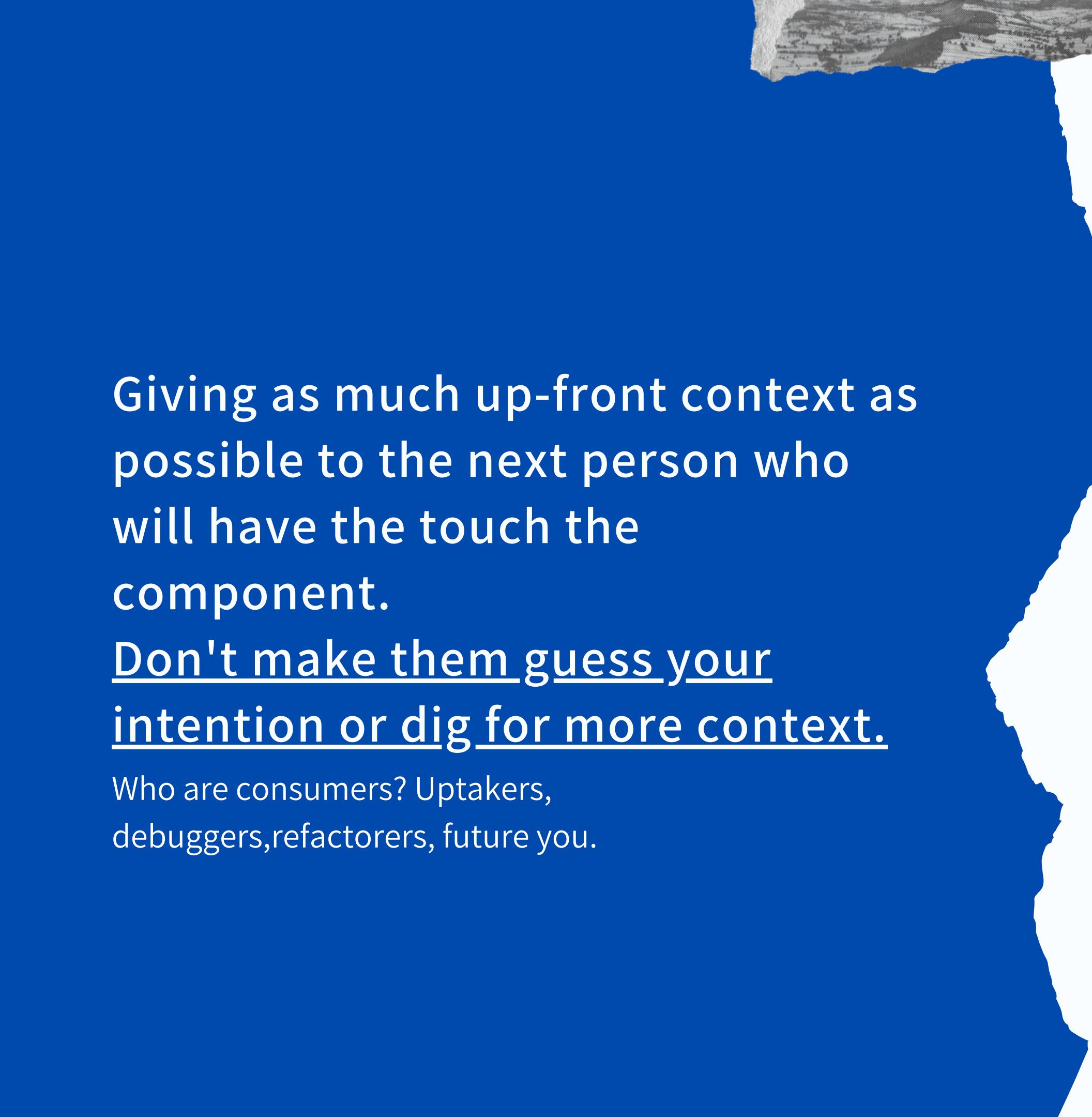
33     )

34 }

# Card - The Pull Request

@dariacaraway

# Considerate React Component APIs



Giving as much up-front context as possible to the next person who will have to touch the component.

Don't make them guess your intention or dig for more context.

Who are consumers? Uptakers, debuggers, refactors, future you.

# Card - Bug Fix



**Mia is a very good girl. She loves long hikes and chasing bunnies. Mia's favorite food is beef jerky treats and hamburger. Her favorite season is Winter. During the first snow of the season, you will find her rolling around outside having a great time.**

Mia 🐾

# Card - Bug Fix



**Mia is a very good girl. She loves long hikes and chasing bunnies. Mia's favorite food is beef jerky treats and hamburger. Her favorite season is Winter. During the first snow of the season, you will find her rolling around outside having a great time.**

Mia 🐶



Mia 🐶

Mia is a very good girl. She loves long hikes and chasing bunnies. Mia's favorite food is beef jerky treats and hamburger. Her favorite season is Winter. During the first snow of the season, you will find her rolling around outside having a great time.

↑ **UI-234**

Convert Select to TypeScript

# Getting familiar

TypeScript Conversion

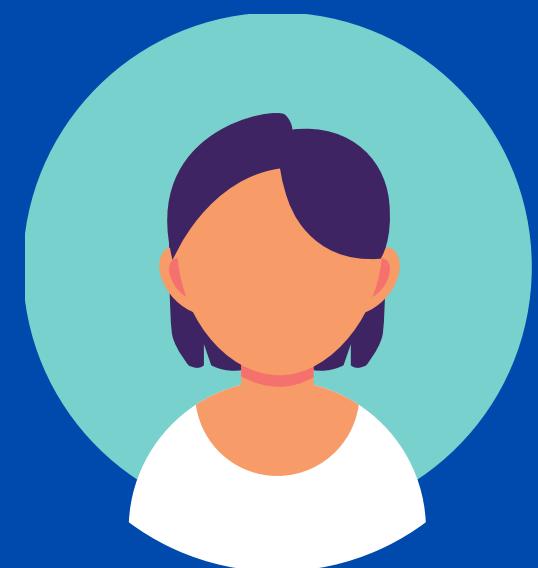
The team has decided to move to TypeScript.  
Update the select component to TS with proper  
types.

Species:

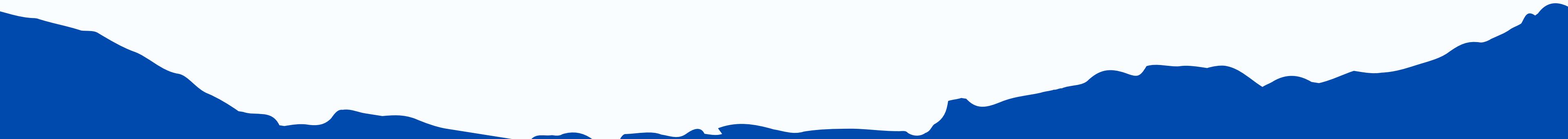
Dog



# Digging into TypeScript



# What is TypeScript?



Typed superset  
of JavaScript

Compile-time type checking.

Compiles to JavaScript.

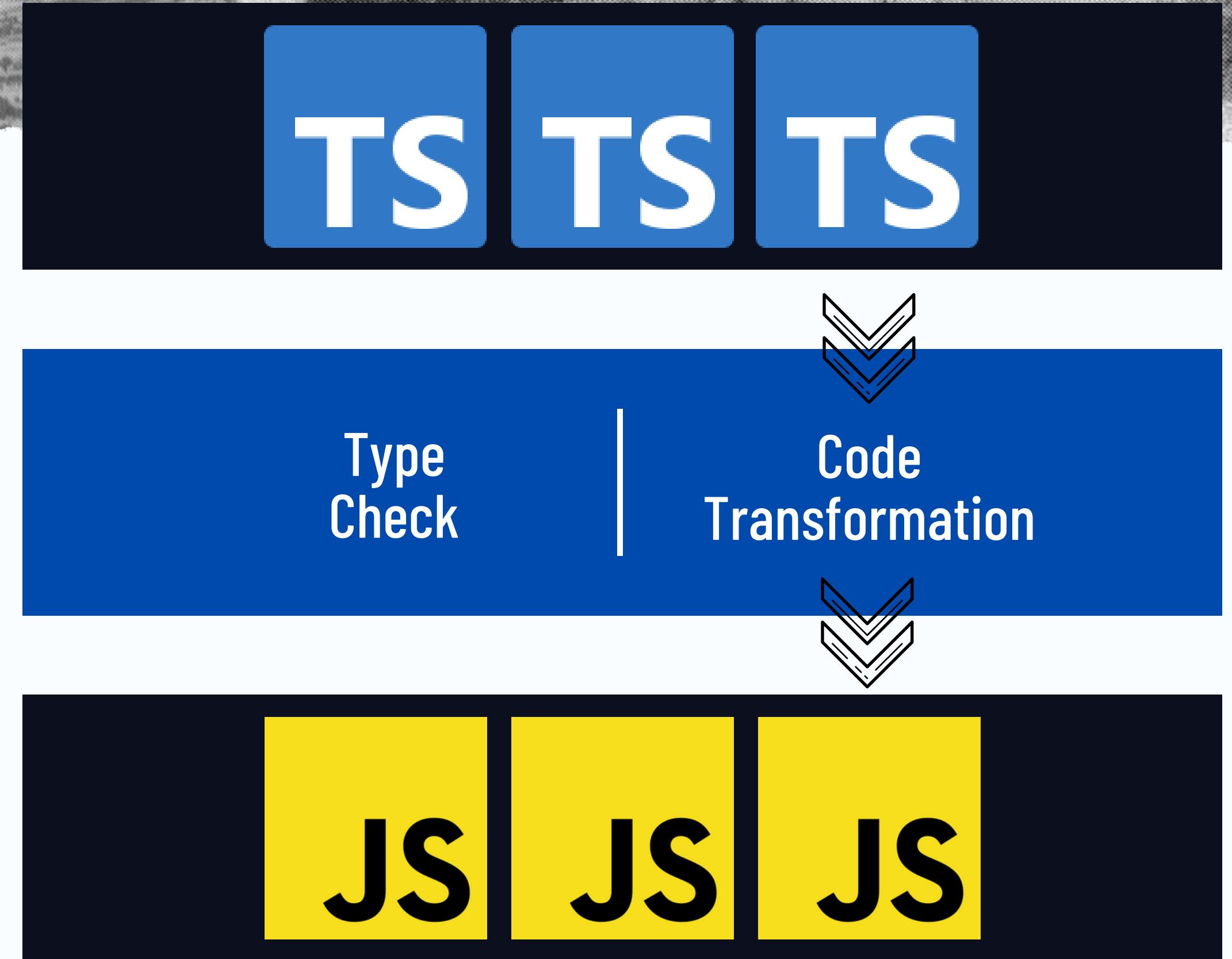
# Why use TypeScript?

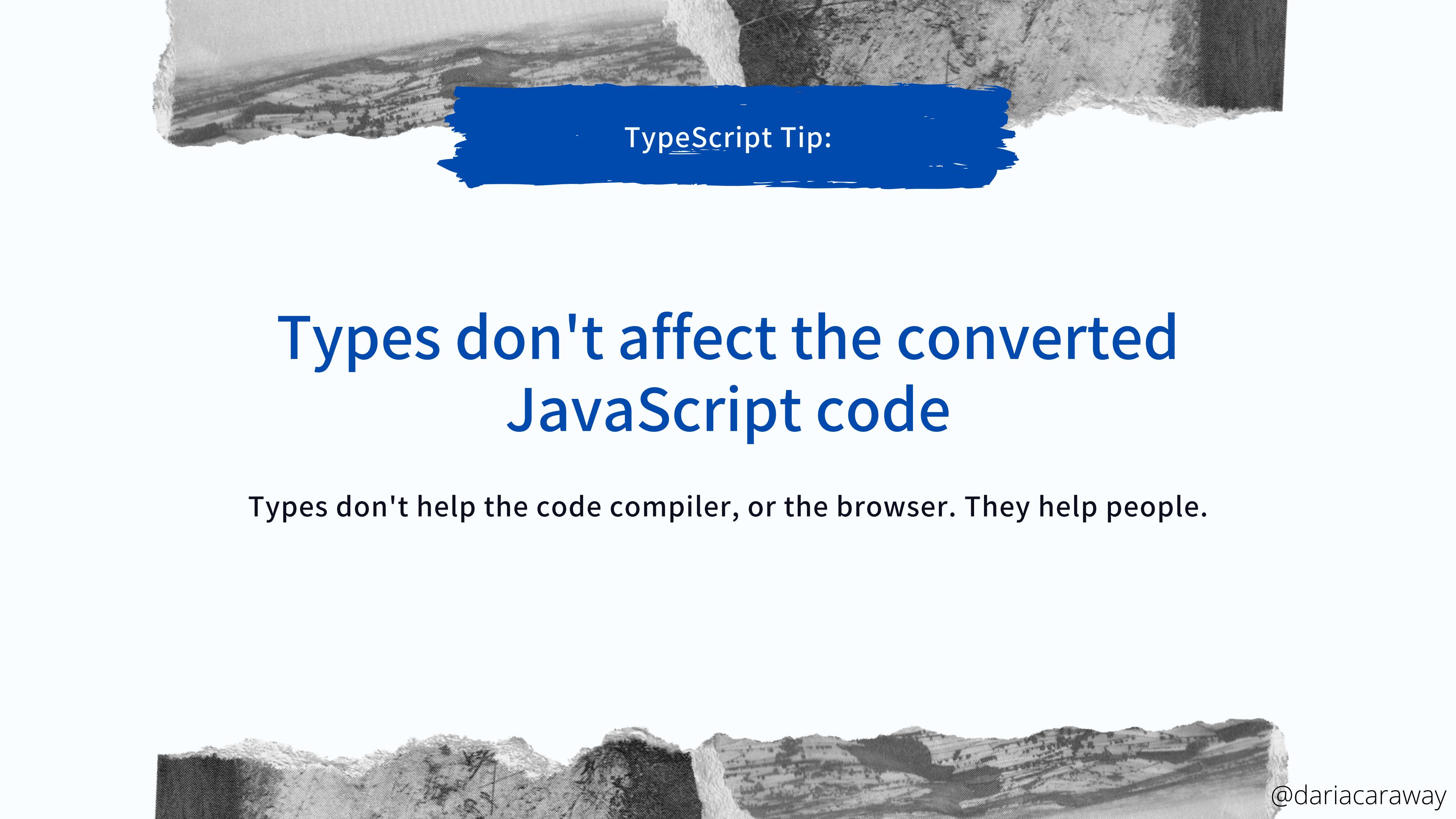
THE USUAL PITCH:

**Static code  
type checking**

Types can help catch bugs during compile-time instead of run-time.

# TypeScript Compiler





TypeScript Tip:

# Types don't affect the converted JavaScript code

Types don't help the code compiler, or the browser. They help people.

MY PITCH:

# TypeScript is a communication tool



Allows  
developers to  
add more  
upfront context  
into their code

# Using TypeScript to be more considerate...



# TypeScript Conversion

Select component conversion

↑ **UI-234**

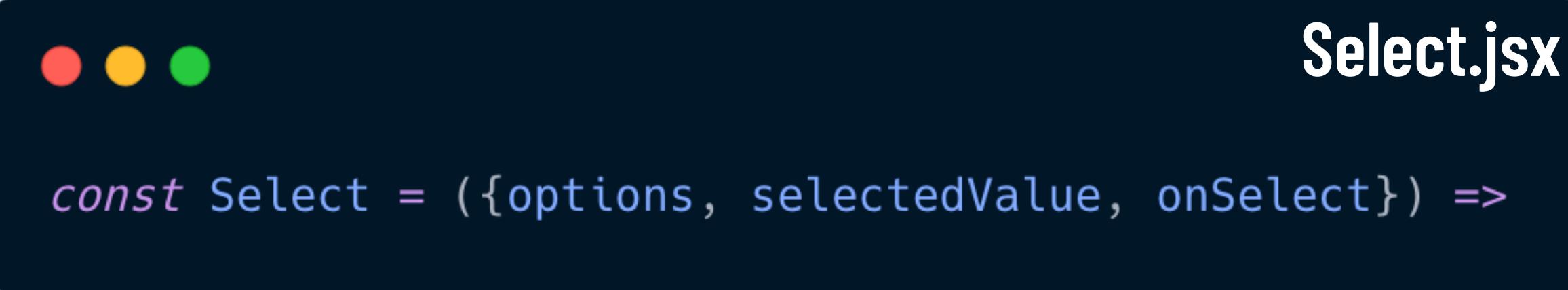
Convert Select to TypeScript

The team has decided to move to TypeScript.  
Update the select component to TS with proper  
types.

Species:

Dog

# Select - The Component



**options**  
**selectedValue**  
**onSelect**

[Code](#)[Issues](#)[Pull requests 5](#)[Actions](#)[Projects](#)[Security](#)[Insights](#)[Settings](#)

UI234-ts-select ▾

10 branches

0 tags

[Go to file](#)[Add file ▾](#)[Code ▾](#)[About](#)

Bark Park main web app repo.

[Readme](#)

## Releases

No releases published

[Create a new release](#)

## Packages

No packages published

[Publish your first package](#)

## Languages



darcar31 second try

ff81eb0 7 days ago 14 commits



public

(feat): Explore page

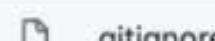
7 days ago



src

second try

7 days ago



.gitignore

steps

6 months ago



README.md

second try

7 days ago



package.json

(feat): Explore page

7 days ago



tsconfig.json

Initialize project using Create React App

9 months ago



yarn.lock

(feat): Explore page

7 days ago

[README.md](#)

## Bark Park Main Application

This is the main repo for the Bark Park web application.

### Running the App

Select - Opening The PR

@dariacaraway

(feat): Select - Convert to typescript #4

Open

Changes from 1 commit ▾

File filter... ▾

Clear filters

Jump to... ▾



10 src/components/select>Select.jsx → src/components/select>Select.tsx

@@ -19,9 +19,15 @@ const selectStyles = css`

19 }

20 ,

21

22 - const Select = ({options, selectedValue, onSelect}) => {

Review changes ▾

23

19 }

20 ,

21

22 + interface SelectProps {

23 + options: any[]

24 + selectedValue: any

25 + onSelect: Function

26 + }

27 +

28 + const Select: React.FC<SelectProps> = ({options, selectedValue, onSelect}:  
SelectProps) => {

29

Select - The Pull Request - First Try

@dariacaraway

▼ 10 src/components/select&gt;Select.jsx → src/components/select&gt;Select.tsx

↑ @@ -19,9 +19,15 @@ const selectStyles = css`

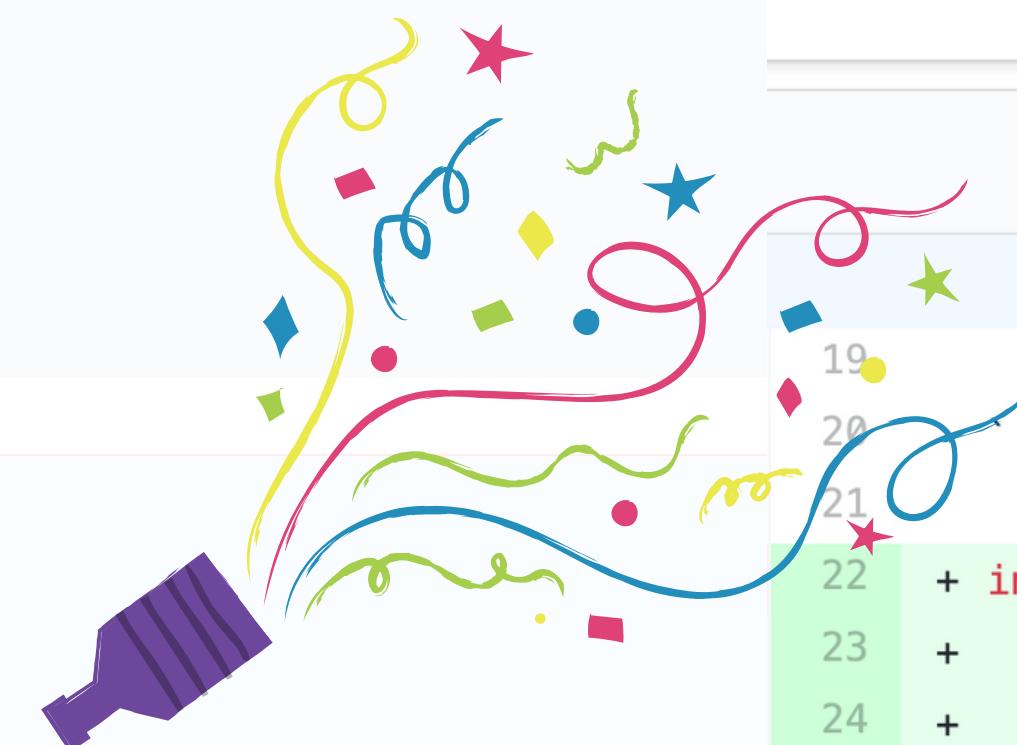
19 }

20 ,

21

22 - const Select = ({options, selectedValue, onSelect}) =&gt; {

23



```
+ interface SelectProps {  
+   options: any[]  
+   selectedValue: any  
+   onSelect: Function  
+ }  
+  
+ const Select: React.FC<SelectProps> = ({options, selectedValue, onSelect}:  
SelectProps) => {  
  ...
```

29

Review changes ▾



Select - The Pull Request - First Try

@dariacaraway

# Testing the Component API Like a Consumer



Checking If Your Types  
Are Considerate

Putting aside any implicit  
knowledge of the subject, what  
do the types communicate?

```
19      }
20
21
22 + interface SelectProps {
23 +   options: any[]
24 +   selectedValue: any
25 +   onSelect: Function
26 +
27 +
28 + const Select: React.FC<SelectProps> = ({options, selectedValue, onSelect}:
29   SelectProps) => {
30
31 +   const onSelectHandler = (e: any) => {
32     const selectedOption = parseInt(e.currentTarget.value)
33     onSelect(options[selectedOption].value)
34   }
35 }
```



ANY? FUNCTION? ???

## Select - Testing The Component API

...

**Hidden Context: Options are objects that contain a label and a value, the callback will give you back the value for the selected option**

```
19      }
20
21
22 + interface SelectProps {
23 +   options: any[]
24 +   selectedValue: any
25 +   onSelect: Function
26 +
27 +
28 + const Select: React.FC<SelectProps> = ({options, selectedValue, onSelect}:
29   SelectProps) => {
30
30 +   const onSelectHandler = (e: any) => {
31     const selectedOption = parseInt(e.currentTarget.value)
32     onSelect(options[selectedOption].value)
33 }
```

## 13 src/components/select&gt;Select.tsx

@@ -19,13 +19,18 @@ const selectStyles = css`

19 }  
20 '  
21

22 interface SelectProps {

23 - options: any[]  
24 - selectedValue: any  
25 - onSelect: Function

26 }

27  
28 - const Select: React.FC<SelectProps> = ({options, selectedValue, onSelect}:  
SelectProps) => {29  
30     const onSelectHandler = (e: any) => {  
31         const selectedOption = parseInt(e.currentTarget.value)

... ↴

Review changes ▾

... ↴

19 }

20 '  
21

22 + interface SelectOption {

23 +     value: string

24 +     label: string

25 + }

26 +

27     interface SelectProps {

28 +     options: SelectOption[]

29 +     selectedValue: string

30 +     onSelect: (value: string) => void

31 }

32

33 +     const Select: React.FC<SelectProps> = ({options, selectedValue, onSelect}:  
SelectProps) => {

34

35         const onSelectHandler = (e: any) => {  
36             const selectedOption = parseInt(e.currentTarget.value)

## 13 src/components/select&gt;Select.tsx

@@ -19,13 +19,18 @@ const selectStyles = css`

19 }  
20 '  
21

```
22 interface SelectProps {  
23 - options: any[]  
24 - selectedValue: any  
25 - onSelect: Function  
26 }  
27  
28 - const Select: React.FC<SelectProps> = ({options, selectedValue, onSelect}:  
  SelectProps) => {  
29  
30   const onSelectHandler = (e: any) => {  
31     const selectedOption = parseInt(e.currentTarget.value)
```

Hidden Context

```
19 }  
20 '  
21  
22 + interface SelectOption {  
23 +   value: string  
24 +   label: string  
25 + }  
26 +  
27 interface SelectProps {  
28 +   options: SelectOption[]  
29 +   selectedValue: string  
30 +   onSelect: (value: string) => void  
31 }  
32  
33 + const Select: React.FC<SelectProps> = ({options, selectedValue, onSelect}:  
  SelectProps) => {  
34  
35   const onSelectHandler = (e: any) => {  
36     const selectedOption = parseInt(e.currentTarget.value)
```

Baked-in Context

# Compiling types are not always helpful



TypeScript types can be as loose or as strict as you want

It is up to the dev to think like a consumer and figure out how to bring the most value.

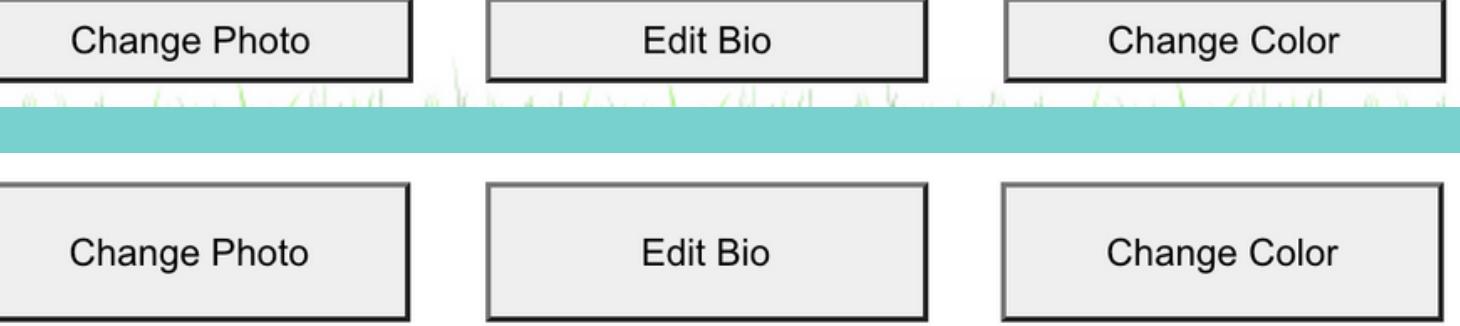


# Getting the Hang of It

Button sizes

↑ **UI-237** Increase the size of the buttons

The design team wants all of the buttons on the page to be a uniform larger size. Update all of the buttons to size medium.



# Button - The Component



Button.tsx

```
interface ButtonProps {  
  size: string  
  variant: string  
  onClick: (e: React.MouseEvent<HTMLButtonElement>) => void  
  children: React.ReactNode  
}  
  
const Button: React.FC<ButtonProps> = ({size, variant, onClick, children}: ButtonProps) =>
```

# Button - The Component



Button.tsx

```
interface ButtonProps {  
  size: string  
  variant: string  
  onClick: (e: React.MouseEvent<HTMLButtonElement>) => void  
  children: React.ReactNode  
}  
  
const Button: React.FC<ButtonProps> = ({size, variant, onClick, children}: ButtonProps) =>
```



SIZE: PIXELS? S/M/L? ???

# Button - The Component



## Button.tsx

```
const Button: React.FC<ButtonProps> = ({size, variant, onClick, children}: ButtonProps) => {
  let sizeStyle
  let variantStyle

  switch(size) {
    case 'small':
      sizeStyle = smallButtonCss
      break
    case 'medium':
      sizeStyle = mediumButtonCss
      break
    case 'large':
      sizeStyle = largeButtonCss
      break
  }

  switch(variant) {
    case 'primary':
      variantStyle = primaryButtonCss
      break
    case 'secondary':
      variantStyle = secondaryButtonCss
      break
  }
}
```

Hidden Context: sizes and variants are  
be predefined strings that map to  
corresponding css

# Button - Adding Context to The Component



Button.tsx

```
type ButtonSize = 'small' | 'medium' | 'large'  
type ButtonVariant = 'primary' | 'secondary'  
  
export interface ButtonProps {  
  size: ButtonSize  
  variant: ButtonVariant
```

## String Literal Unions

Enum- like behavior. Button component will only accept the strings defined in ButtonSize.

[Code](#)[Issues](#)[Pull requests 6](#)[Actions](#)[Projects](#)[Security](#)[Insights](#)[Settings](#)

UI237-button-s...

10 branches 0 tags

[Go to file](#)[Add file](#)[Code](#)[About](#)

This branch is 2 commits ahead of master.

[Pull request](#) [Compare](#)

darcar31 string union

b6e403a 7 days ago 13 commits

public	(feat): Explore page	7 days ago
src	string union	7 days ago
.gitignore	steps	6 months ago
README.md	string union	7 days ago
package.json	(feat): Explore page	7 days ago
tsconfig.json	Initialize project using Create React App	9 months ago
yarn.lock	(feat): Explore page	7 days ago

README.md

## Bark Park Main Application

This is the main repo for the Bark Park web application.

[Running the App](#)

# Button - Opening The PR

@dariacaraway

(feat): Edit Profile - Make buttons larger #5

Open

Changes from 1 commit ▾ File filter... ▾ × Clear filters

Jump to... ▾ ⚙ ▾

▼ 7  src/components/button/Button.tsx 

36

```
37 interface ButtonProps {  
38   - size: string  
39   - variant: string  
40   onClick: (e: React.MouseEvent<HTMLButtonElement>) => void  
41   children: React.ReactNode  
42 }
```

....

Review changes ▾

...

```
36  
37 + type ButtonSize = 'small' | 'medium' | 'large'  
38 + type ButtonVariant = 'primary' | 'secondary'  
39 +  
40 interface ButtonProps {  
41   + size: ButtonSize  
42   + variant: ButtonVariant  
43   onClick: (e: React.MouseEvent<HTMLButtonElement>) => void  
44   children: React.ReactNode  
45 }
```

## Button - The Pull Request

@dariacaraway

▼ 7  src/components/button/Button.tsx 

36

```
37 interface ButtonProps {  
38   - size: string Hidden Context  
39   - variant: string  
40   onClick: (e: React.MouseEvent<HTMLButtonElement>) => void  
41   children: React.ReactNode  
42 }
```

....

Review changes ▾

## Button - The Pull Request

```
36  
37 + type ButtonSize = 'small'|'medium'|'large'  
38 + type ButtonVariant = 'primary' | 'secondary'  
39 +  
40 interface ButtonProps { Baked-in Context  
41 + size: ButtonSize  
42 + variant: ButtonVariant  
43   onClick: (e: React.MouseEvent<HTMLButtonElement>) => void  
44   children: React.ReactNode  
45 }
```

@dariacaraway

# Testing the Component API Like a Consumer



What does it look like to use the component?

String literal unions seem great, but do you have still to click into the component to find the options?

# Testing The Component API - IntelliSense

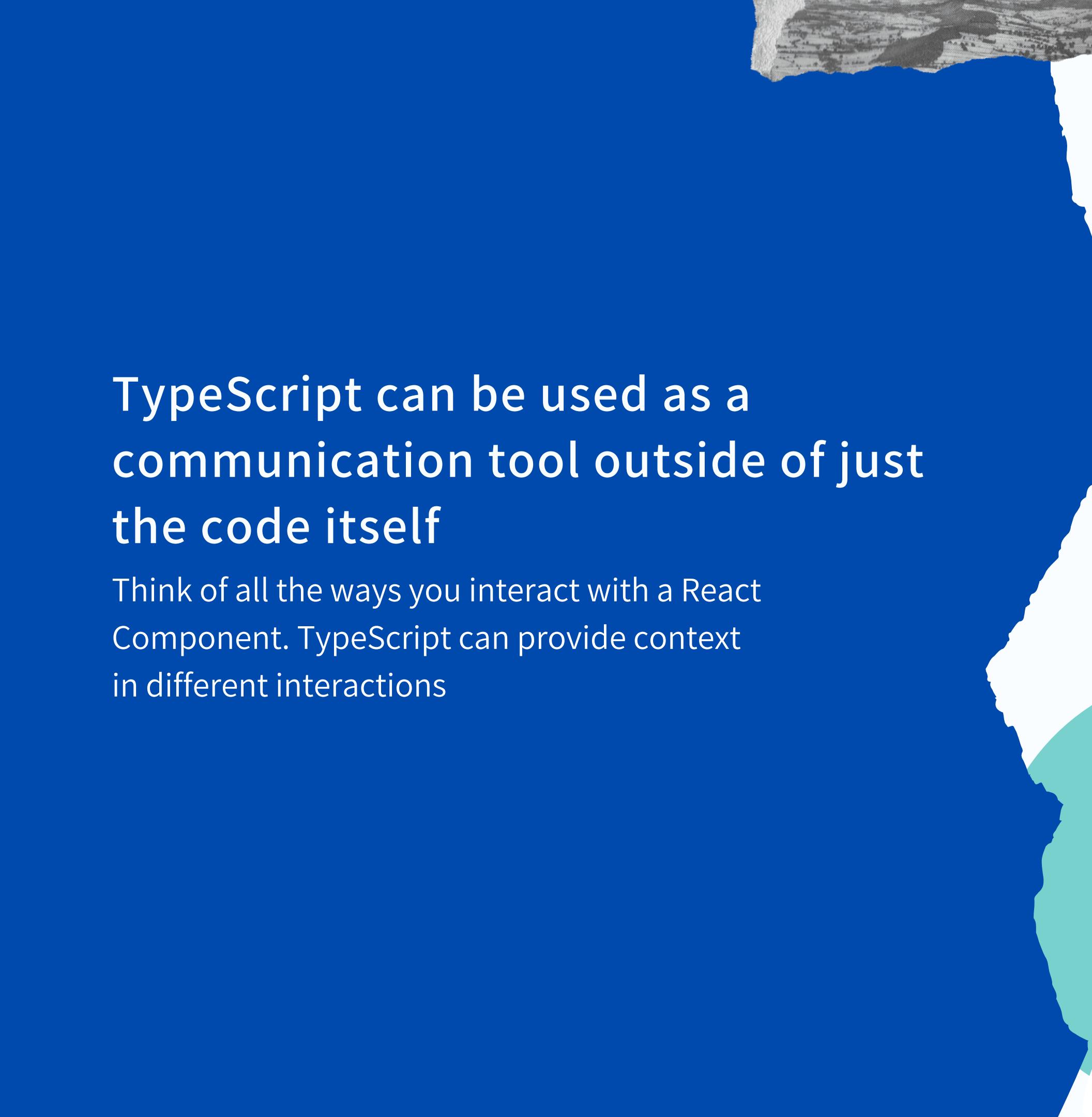


Baked-in Context

EditProfile.tsx

Auto Complete & IntelliSense

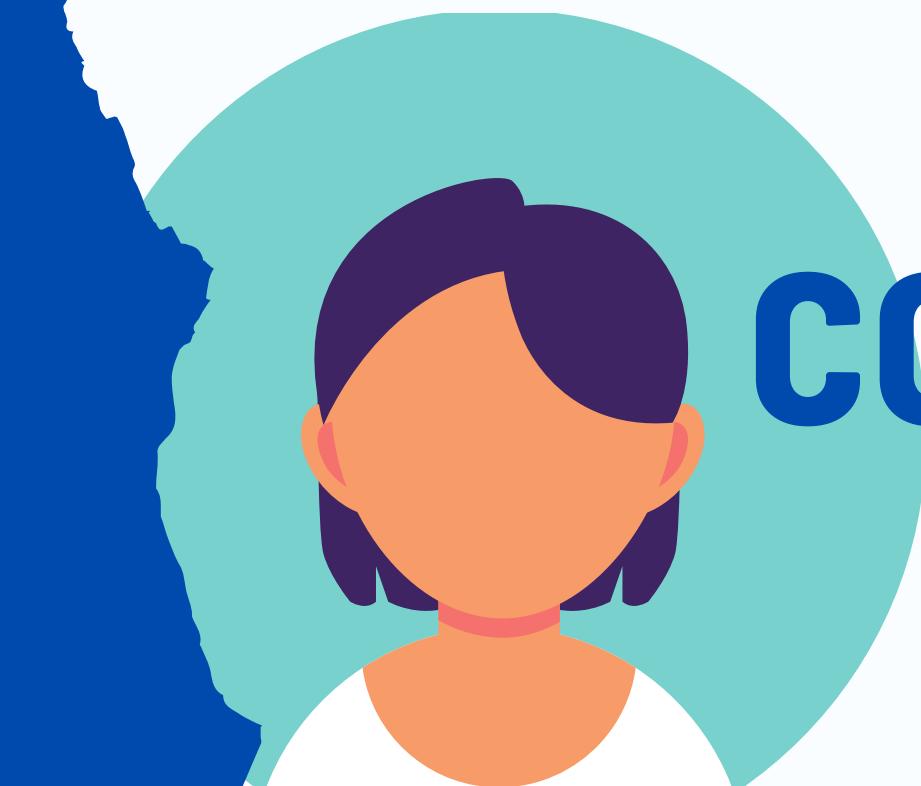
The intellisense consumers see when interacting with your components is another communication tool to consider.



TypeScript can be used as a communication tool outside of just the code itself

Think of all the ways you interact with a React Component. TypeScript can provide context in different interactions

There are many aspects to being a consumer



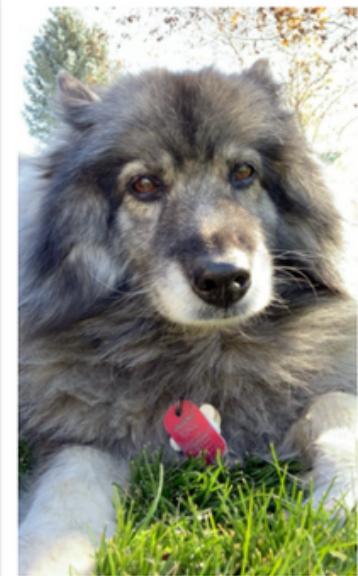
# Medium Buttons

BARK PARK

Explore Play Dates <sup>2</sup> Park Finder



## Edit Profile



Mia 🐾

Mia is a very good girl. She loves long hikes and chasing bunnies. Mia's favorite food is beef jerky treats and hamburger. Her favorite season is Winter. During the first snow of the season, you will find her rolling around outside having a great time.

Species:

Dog



Change Photo

Edit Bio

Change Color

Save

@dariacaraway

# Medium Buttons

BARK PARK

Explore Play Dates <sup>2</sup> Park Finder



BARK PARK

Edit Profile



Edit Profile

Mia 🐶

Mia is a very good girl. She loves long hikes and chasing bunnies. Mia's favorite food is beef jerky treats and hamburger. Her favorite season is Winter. During the first snow of the season, you will find her rolling around outside having a great time.

Species:

Dog ▾

Change Photo

Edit Bio

Change Color



Save



@dariacaraway

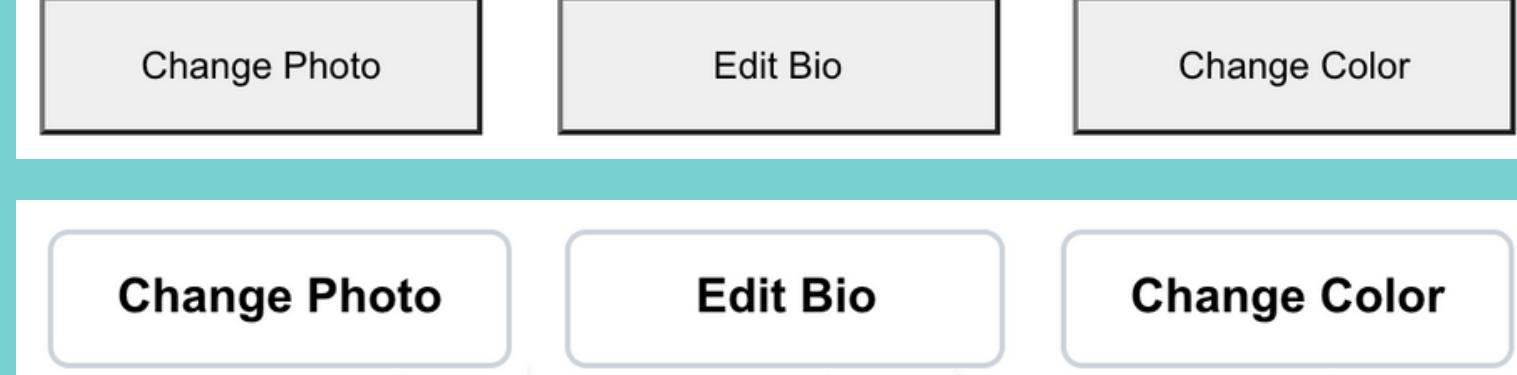
# Button Expert Lisa

Awesome Buttons

↑ UI-502

Convert buttons to AwesomeButton

The core component team has created a new button that all other teams need to uptake. Convert all the buttons on the Edit Profile page to the new AwesomeButton.



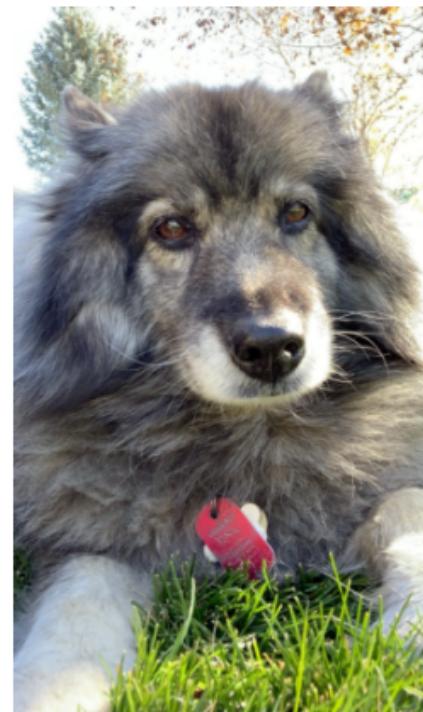
# Save Button Converted to AwesomeButton

BARK PARK

Explore Play Dates <sup>2</sup> Park Finder



## Edit Profile



Mia 🐶

Mia is a very good girl. She loves long hikes and chasing bunnies. Mia's favorite food is beef jerky treats and hamburger. Her favorite season is Winter. During the first snow of the season, you will find her rolling around outside having a great time.

Species:

Dog ▾

Change Photo

Edit Bio

Change Color



Save

@dariacaraway

# AwesomeButton - The Consumer



EditProfile.tsx

```
<div css={saveContainer}>
  <AwesomeButton size='large' variant='primary' onClick={saveButtonClick}> Save </AwesomeButton>
</div>
```

Same API as the old Button

[Code](#)[Issues](#)[Pull requests 7](#)[Actions](#)[Projects](#)[Security](#)[Insights](#)[Settings](#)

UI502-uptake-a...

10 branches

0 tags

[Go to file](#)[Add file](#)[Code](#)

This branch is 3 commits ahead of master.

[Pull request](#)[Compare](#)

darcar31 omit

5a9ae01 7 days ago 14 commits



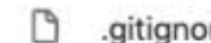
public (feat): Explore page

7 days ago



src omit

7 days ago



.gitignore steps

6 months ago



README.md omit

7 days ago



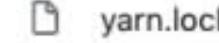
package.json (feat): Explore page

7 days ago



tsconfig.json Initialize project using Create React App

9 months ago



yarn.lock (feat): Explore page

7 days ago

README.md

## Bark Park Main Application

This is the main repo for the Bark Park web application.

[Running the App](#)

# AwesomeButton - Opening The PR

@dariacaraway

### About

Bark Park main web app repo.

[Readme](#)

### Releases

No releases published

[Create a new release](#)

### Packages

No packages published

[Publish your first package](#)

### Languages



Open

(feat): EditProfile - Convert Buttons to awesome buttons #13

Changes from 1 commit ▾

File filter... ▾

Clear filters Jump to... ▾



6 src/prez-steps/EditProfile.tsx

```
@@ -169,9 +169,9 @@ export const StepTwo = () => {
    </div>
    </div>
    <div css={buttonContainer}>
-     <Button size='medium' variant='secondary' onClick=
+ {editPhotoClick}> Change Photo </Button>
+ - <Button size='medium' variant='secondary' onClick=
+ {buttonThreeClick}> Edit Bio</Button>
-     <Button size='medium' variant='secondary' onClick=
+ {colorButtonClick}> Change Color </Button>
    </div>
    <div css={saveContainer}>
        <AwesomeButton size='large' variant='primary' onClick=
{saveButtonClick}> Save </AwesomeButton>

```

...

Review changes ▾

```
169         </div>
170         </div>
171         <div css={buttonContainer}>
172 +          <AwesomeButton size='medium' variant='secondary' onClick=
+ {editPhotoClick}> Change Photo </AwesomeButton>
173 +          <AwesomeButton size='medium' variant='secondary' onClick=
+ {buttonThreeClick}> Edit Bio</AwesomeButton>
174 +          <AwesomeButton size='medium' variant='secondary' onClick=
+ {colorButtonClick}> Change Color </AwesomeButton>
175         </div>
176         <div css={saveContainer}>
177             <AwesomeButton size='large' variant='primary' onClick=
{saveButtonClick}> Save </AwesomeButton>
```

# AwesomeButton - The Pull Request

@dariacaraway

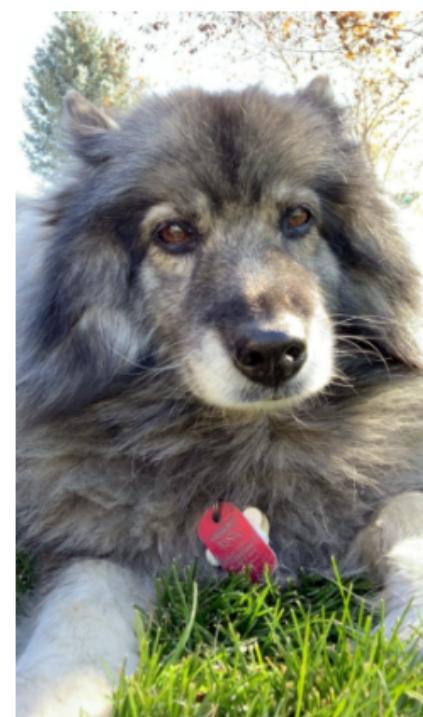
# Awesome Button - Testing The Change

BARK PARK

Explore Play Dates <sup>2</sup> Park Finder



## Edit Profile



Mia 🐶

Mia is a very good girl. She loves long hikes and chasing bunnies. Mia's favorite food is beef jerky treats and hamburger. Her favorite season is Winter. During the first snow of the season, you will find her rolling around outside having a great time.

Species:

Dog ▾

Change Photo

Edit Bio

Change Color

Save

@dariacaraway

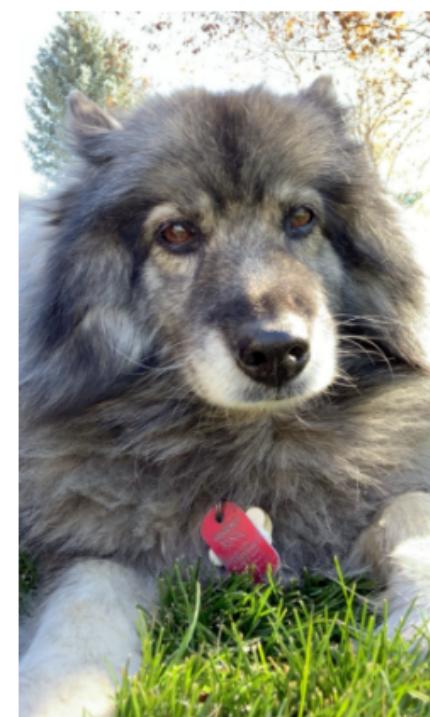
# Awesome Button - Testing The Change

BARK PARK

Explore Play Dates <sup>2</sup> Park Finder



## Edit Profile



Mia 🐶

Mia is a very good girl. She loves long hikes and chasing bunnies. Mia's favorite food is beef jerky treats and hamburger. Her favorite season is Winter. During the first snow of the season, you will find her rolling around outside having a great time.

Species:

Dog ▾

Change Photo

Edit Bio

Change Color



Save

@dariacaraway

# AwesomeButton - The Component



AwesomeButton.tsx

```
const AwesomeButton: React.FC<AwesomeButtonProps> = ({variant, onClick, children}: AwesomeButtonProps) => {
  const sizeStyle = large
  let variantStyle

  switch(variant) {
    case 'primary':
      variantStyle = primary
      break
    case 'secondary':
      variantStyle = secondary
      break
  }

  return (
    <button onClick={onClick} css={[buttonStyles, sizeStyle, variantStyle]}>
      {children}
    </button>
  )
}
```

SIZE PROP IS NOT  
USED

# AwesomeButton - The Component



AwesomeButton.tsx

```
// Button.tsx
export interface ButtonProps {
  size: ButtonSize
  variant: ButtonVariant
  onClick: (e: React.MouseEvent<HTMLButtonElement>) => void
  children: React.ReactNode
}

// AwesomeButton.tsx
interface AwesomeButtonProps extends Partial<ButtonProps> {}

const AwesomeButton: React.FC<AwesomeButtonProps> = ({variant, onClick, children}: AwesomeButtonProps) =>
```



THE API TAKES A SIZE? ???

# AwesomeButton - The Component



## AwesomeButton.tsx

```
// Button.tsx
export interface ButtonProps {
  size: ButtonSize
  variant: ButtonVariant
  onClick: (e: React.MouseEvent<HTMLButtonElement>) => void
  children: React.ReactNode
}

// AwesomeButton.tsx
interface AwesomeButtonProps extends Partial<ButtonProps> {}

const AwesomeButton: React.FC<AwesomeButtonProps> = ({variant, onClick, children}: AwesomeButtonProps) =>
```

PARTIAL

## Partial Type

The partial type sets every property of the provided type to **optional**.

# AwesomeButton - The Component



AwesomeButton.tsx

```
// Button.tsx
export interface ButtonProps {
  size: ButtonSize
  variant: ButtonVariant
  onClick: (e: React.MouseEvent<HTMLButtonElement>) => void
  children: React.ReactNode
}

// AwesomeButton.tsx
interface AwesomeButtonProps extends Partial<ButtonProps> {}

const AwesomeButton: React.FC<AwesomeButtonProps> = ({variant, onClick, children}: AwesomeButtonProps) =>
```

PARTIAL

Hidden Context: Accept all of the props in  
button props, except size

# Awesome Button - Adding Context



```
export interface ButtonProps {  
    size: ButtonSize  
    variant: ButtonVariant  
    onClick: (e: React.MouseEvent<HTMLButtonElement>) => void  
    children: React.ReactNode  
}  
  
interface AwesomeButtonProps extends Omit<ButtonProps, 'size'> {}
```

OMIT

Omit Type

The omit type constructs a new type with all the keys of the provided type, minus any of the keys specified.

Open

(feat): EditProfile - Convert Buttons to awesome buttons #13

Changes from 1 commit ▾

File filter... ▾

Clear filters

Jump to... ▾



4 src/components/button/AwesomeButton.tsx

```
49 - interface AwesomeButtonProps extends Partial<ButtonProps> {}  
50  
51   const AwesomeButton: React.FC<AwesomeButtonProps> = ({variant, onClick,  
children}: AwesomeButtonProps) => {  
52     const sizeStyle = largeButtonCss  
....
```

Review changes ▾

...

```
49 + interface AwesomeButtonProps extends Omit<ButtonProps, 'size'> {}  
50  
51   const AwesomeButton: React.FC<AwesomeButtonProps> = ({variant, onClick,  
children}: AwesomeButtonProps) => {  
52     const sizeStyle = largeButtonCss
```

## AwesomeButton - Adding Context

@dariacaraway

 Open

(feat): EditProfile - Convert Buttons to awesome buttons #13

Changes from 1 commit ▾

File filter... ▾

Clear filters

Jump to... ▾



4  src/components/button/AwesomeButton.tsx

## Hidden Context.

```
49 - interface AwesomeButtonProps extends Partial<ButtonProps> {}  
50  
51   const AwesomeButton: React.FC<AwesomeButtonProps> = ({variant, onClick,  
children}: AwesomeButtonProps) => {  
52     const sizeStyle = largeButtonCss  
....
```

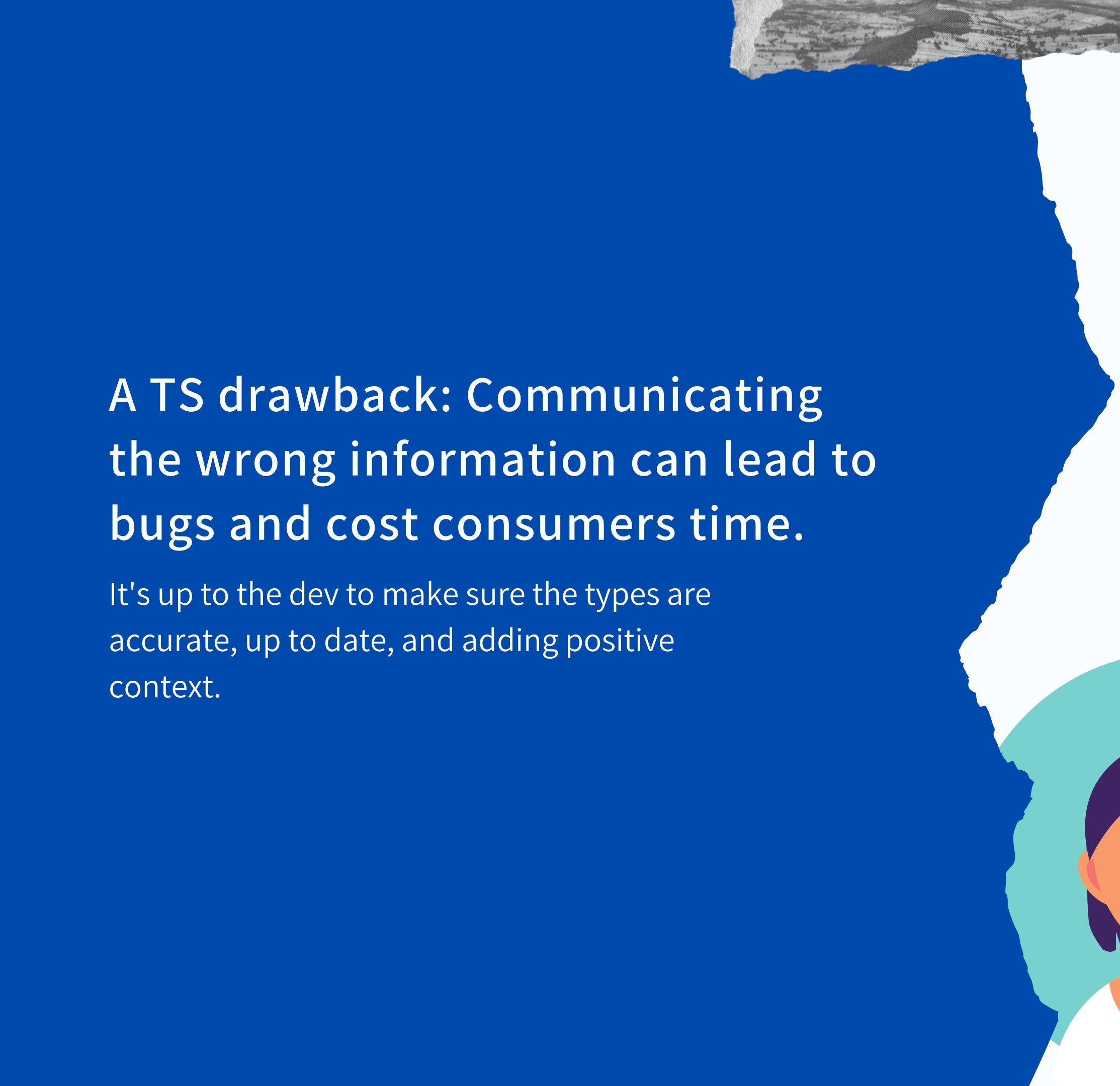
 Review changes ▾

## Baked-in Context.

```
49 + interface AwesomeButtonProps extends Omit<ButtonProps, 'size'> {}  
50  
51   const AwesomeButton: React.FC<AwesomeButtonProps> = ({variant, onClick,  
children}: AwesomeButtonProps) => {  
52     const sizeStyle = largeButtonCss
```

# AwesomeButton - Adding Context

@dariacaraway



A TS drawback: Communicating the wrong information can lead to bugs and cost consumers time.

It's up to the dev to make sure the types are accurate, up to date, and adding positive context.

Incorrect or misleading types can be worse than no types



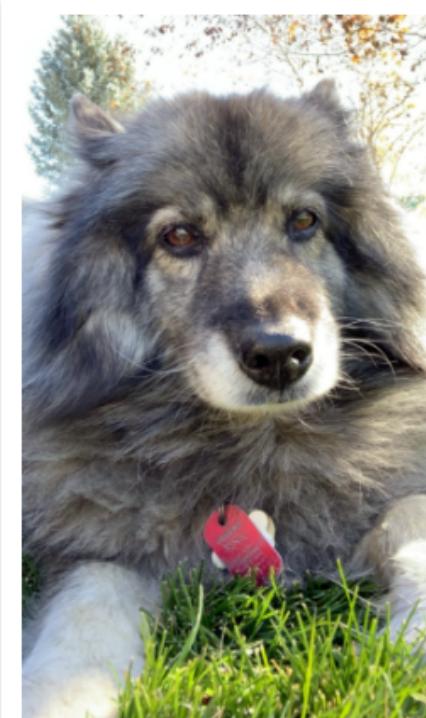
# Awesome Button Uptake

BARK PARK

Explore Play Dates <sup>2</sup> Park Finder



## Edit Profile



Mia 🐶

Mia is a very good girl. She loves long hikes and chasing bunnies. Mia's favorite food is beef jerky treats and hamburger. Her favorite season is Winter. During the first snow of the season, you will find her rolling around outside having a great time.

Species:

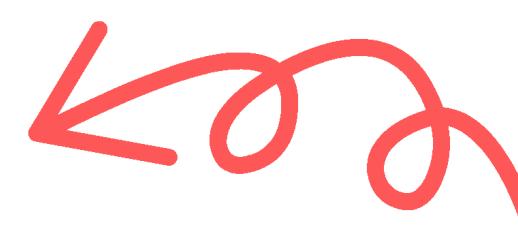
Dog ▾



Change Photo

Edit Bio

Change Color



Save

@dariacaraway

# TypeScript Connoisseur Lisa

Selecting numbers

↑ **UI-651**

Update select to allow numbers

Users want the ability to display the age of their pet. To do this, we need a select component that can take numbers, not just strings.

Species:

Age:

# Select - The Component



Select.tsx

```
export interface SelectOption {  
  value: string  
  label: string  
}  
  
interface SelectProps {  
  options: SelectOption[]  
  selectedValue: string  
  onSelect: (value: string) => void  
}  
  
const Select: React.FC<SelectProps> = ({options, selectedValue, onSelect}: SelectProps) =>
```

# Select - The Component



Select.tsx

```
export interface SelectOption {  
  value: string  
  label: string  
}  
  
interface SelectProps {  
  options: SelectOption[]  
  selectedValue: string  
  onSelect: (value: string) => void  
}  
  
const Select: React.FC<SelectProps> = ({options, selectedValue, onSelect}: SelectProps) =>
```

# Select - The Component - Supporting Numbers



Select.tsx

```
export interface SelectOption {  
  value: string | number  
  label: string  
}  
  
interface SelectProps {  
  options: SelectOption[]  
  selectedValue: (string | number)  
  onSelect: (value: string | number) => void  
}  
  
const Select: React.FC<SelectProps> = ({options, selectedValue, onSelect}: SelectProps) => {
```

Union Type

Composes a type from the union of two or more types. The property must be one of the supplied types.

[Code](#) [Issues](#) [Pull requests 8](#) [Actions](#) [Projects](#) [Security](#) [Insights](#) [Settings](#)

[UI651-number-s...](#) [10 branches](#) [0 tags](#) [Go to file](#) [Add file](#) [Code](#)

This branch is 3 commits ahead of master. [Pull request](#) [Compare](#)

darcar31 generic	158eef9 7 days ago	14 commits
public	(feat): Explore page	7 days ago
src	generic	7 days ago
.gitignore	steps	6 months ago
README.md	(feat): update readme and run scripts	7 days ago
package.json	(feat): Explore page	7 days ago
tsconfig.json	Initialize project using Create React App	9 months ago
yarn.lock	(feat): Explore page	7 days ago

[README.md](#)

## Bark Park Main Application

This is the main repo for the Bark Park web application.

carbon (50).png carbon (49).png

About Bark Park main web app repo. [Readme](#)

Releases No releases published [Create a new release](#)

Packages No packages published [Publish your first package](#)

Languages

TypeScript 92.4%	JavaScript 3.3%
HTML 2.8%	CSS 1.5%

# Select - Opening A PR

@dariacaraway

6 src/components/select&gt;Select.tsx

```
22 interface SelectOption {  
23   - value: string  
24   label: string  
25 }  
26  
27 interface SelectProps {  
28   options: SelectOption[]  
29   - selectedValue: string  
30   - onSelect: (value: string) => void  
31 }  
32  
33 function Select({options, selectedValue, onSelect}: SelectProps) {  
  ....  
}
```

Review changes ▾

```
22 interface SelectOption {  
23   + value: string | number  
24   label: string  
25 }  
26  
27 interface SelectProps {  
28   options: SelectOption[]  
29   + selectedValue: string | number  
30   + onSelect: (value: string | number) => void  
31 }  
32  
33 function Select({options, selectedValue, onSelect}: SelectProps) {
```

...

## Select - The Pull Request - First Try

@dariacaraway

6     src/components/select>Select.tsx

```

22  interface SelectOption {
23  -   value: string
24  -   label: string
25  }
26
27  interface SelectProps {
28    options: SelectOption[]
29  -   selectedValue: string
30  -   onSelect: (value: string) => void
31  }
32
33  function Select({options, selectedValue, onSelect}: SelectProps) {
...

```

Review changes ▾



WHEN IS THE ONSELECT VALUE  
A STRING? WHEN IS IT A  
NUMBER?

## Select - The Pull Request - First Try

```

22  interface SelectOption {
23  +   value: string | number
24  -   label: string
25  }
26
27  interface SelectProps {
28    options: SelectOption[]
29  +   selectedValue: string | number
30  +   onSelect: (value: string | number) => void
31  }
32
33  function Select({options, selectedValue, onSelect}: SelectProps) {

```

@dariacaraway

6 src/components/select&gt;Select.tsx

```

22  interface SelectOption {
23  -   value: string
24  -   label: string
25  }
26
27  interface SelectProps {
28    options: SelectOption[]
29  -   selectedValue: string
30  -   onSelect: (value: string) => void
31  }
32
33  function Select({options, selectedValue, onSelect}: SelectProps) {
...

```

Review changes ▾

WHEN IS THE OUTPUT A STRING  
- WHEN IS IT A NUMBER?



## Select - The Pull Request - First Try

```

22  interface SelectOption {
23  +   value: string | number
24  -   label: string
25  }
26
27  interface SelectProps {
28    options: SelectOption[]
29  +   selectedValue: string | number
30  +   onSelect: (value: string | number) => void
31  }
32
33  function Select({options, selectedValue, onSelect}: SelectProps) {

```

Hidden Context: When the option value is a number,  
the onSelect value will also be a number

@dariacaraway

# Select - Adding Context



Select.tsx

```
export interface SelectOption<T> {  
  value: T  
  label: string  
}  
interface SelectProps<T> {  
  options: SelectOption<T>[]  
  selectedValue: T  
  onSelect: (value: T) => void  
}  
  
function Select<T> ({options, selectedValue, onSelect}: SelectProps<T>) {
```

Generic Type

Type T represents a type that the consumer will provide. All Ts are required to be the same type.

14 src/components/select>Select.tsx

```

22 - interface SelectOption {
23 -   value: string | number
24   label: string
25 }
26
27 - interface SelectProps {
28 -   options: SelectOption[]
29 -   selectedValue: string | number
30 -   onSelect: (value: string | number) => void
31 }
32

```

## Hidden Context.

```

33 - function Select({options, selectedValue, onSelect}: SelectProps) {
34   const onSelectHandler = (e: FormEvent<HTMLSelectElement>) => {
35     const selectedOption = parseInt(e.currentTarget.value)
36     onSelect(options[selectedOption].value)
.... ↓

```

Review changes ▾

```

22 + interface SelectOption<T> {
23 +   value: T
24   label: string
25 }
26

```

## Baked-in Context.

```

27 + interface SelectProps<T> {
28 +   options: SelectOption<T>[]
29 +   selectedValue: T
30 +   onSelect: (value: T) => void
31 }
32

```

```

33 + function Select<T> ({options, selectedValue, onSelect}: SelectProps<T>) {
34   const onSelectHandler = (e: FormEvent<HTMLSelectElement>) => {
35     const selectedOption = parseInt(e.currentTarget.value)
36     onSelect(options[selectedOption].value)

```

# TypeScript has a multitude of types



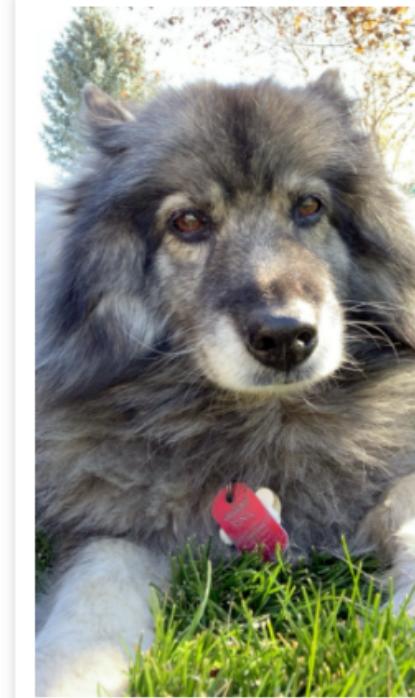
TypeScript has many additional features and is always releasing more to help you bake in the context that exists in your head.

It takes practice, but there is almost always a way to write considerate types.





## Edit Profile



Mia 🐾 - 12yrs

Mia is a very good girl. She loves long hikes and chasing bunnies. Mia's favorite food is beef jerky treats and hamburger. Her favorite season is Winter. During the first snow of the season, you will find her rolling around outside having a great time.

Species:

Age:

[Change Photo](#)[Edit Bio](#)[Change Color](#)[Save](#)

# Successful Product Implementation

@dariacaraway



**Lisa Coder**

Loves Dogs, React and TypeScript

# Lisa's team loves her



Lisa takes advantage of TypeScript as a communication tool to write considerate component APIs that are easier to reason about, refactor, and debug.

# Why. Use TypeScript?

# Why. Use TypeScript?

THE USUAL PITCH:

**Static Type  
Checking**

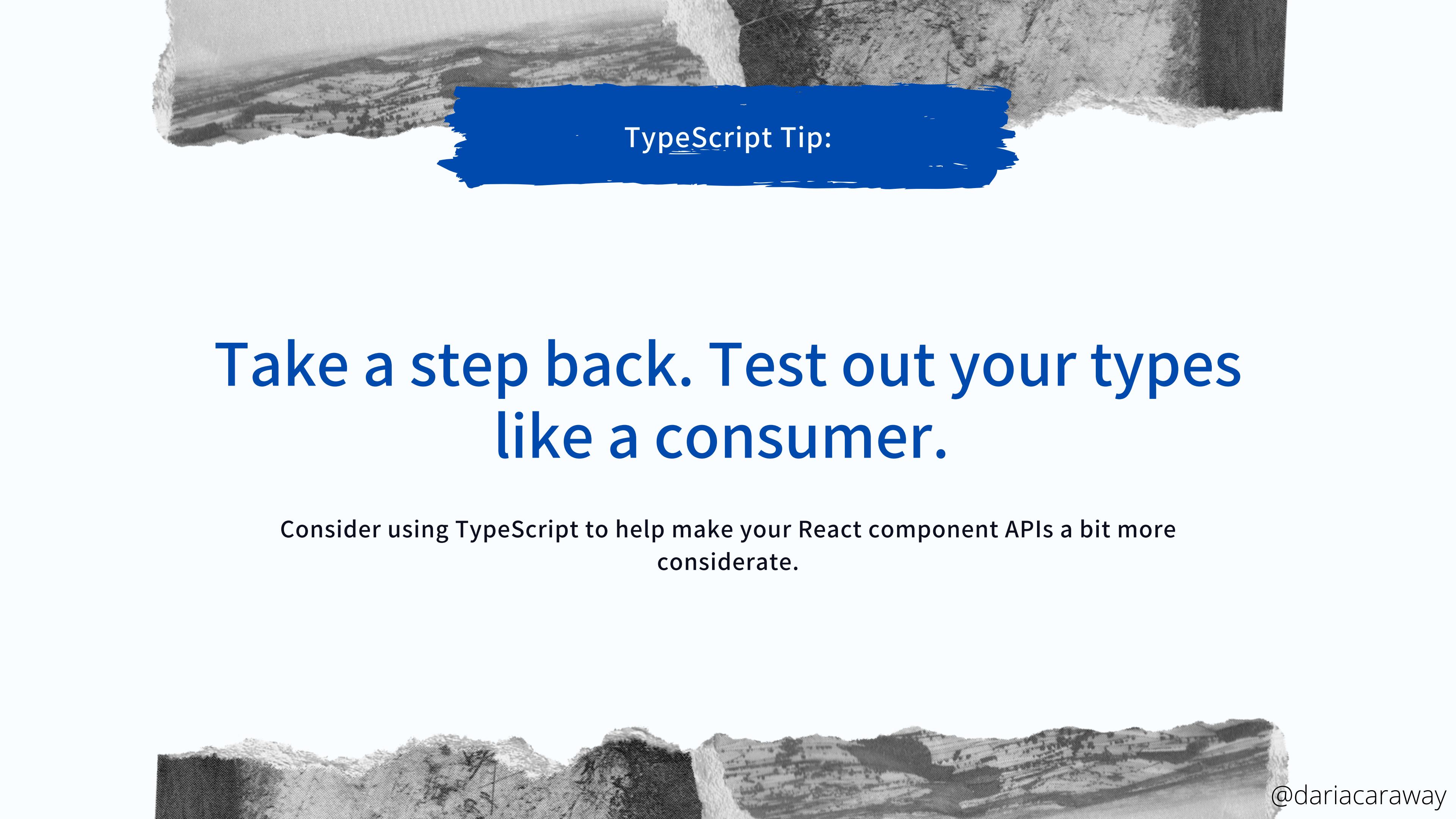
Types can help catch bugs during compile-time instead of run-time.

# Why. Use TypeScript?

MY PITCH:

Communication  
Tool

Types are a way to provide additional context and communicate intent directly in our code and the tools we already use.



TypeScript Tip:

# Take a step back. Test out your types like a consumer.

Consider using TypeScript to help make your React component APIs a bit more considerate.

# Thank You



Presented by Daria Caraway

@dariacaraway