

A decorative border made of colorful, irregular pebble-like shapes in shades of blue, orange, and grey surrounds the central text.

STYLED COMPONENTS FOR REACT - STYLE YOUR WEBSITES AND WEB APPLICATIONS WITHOUT STRESS

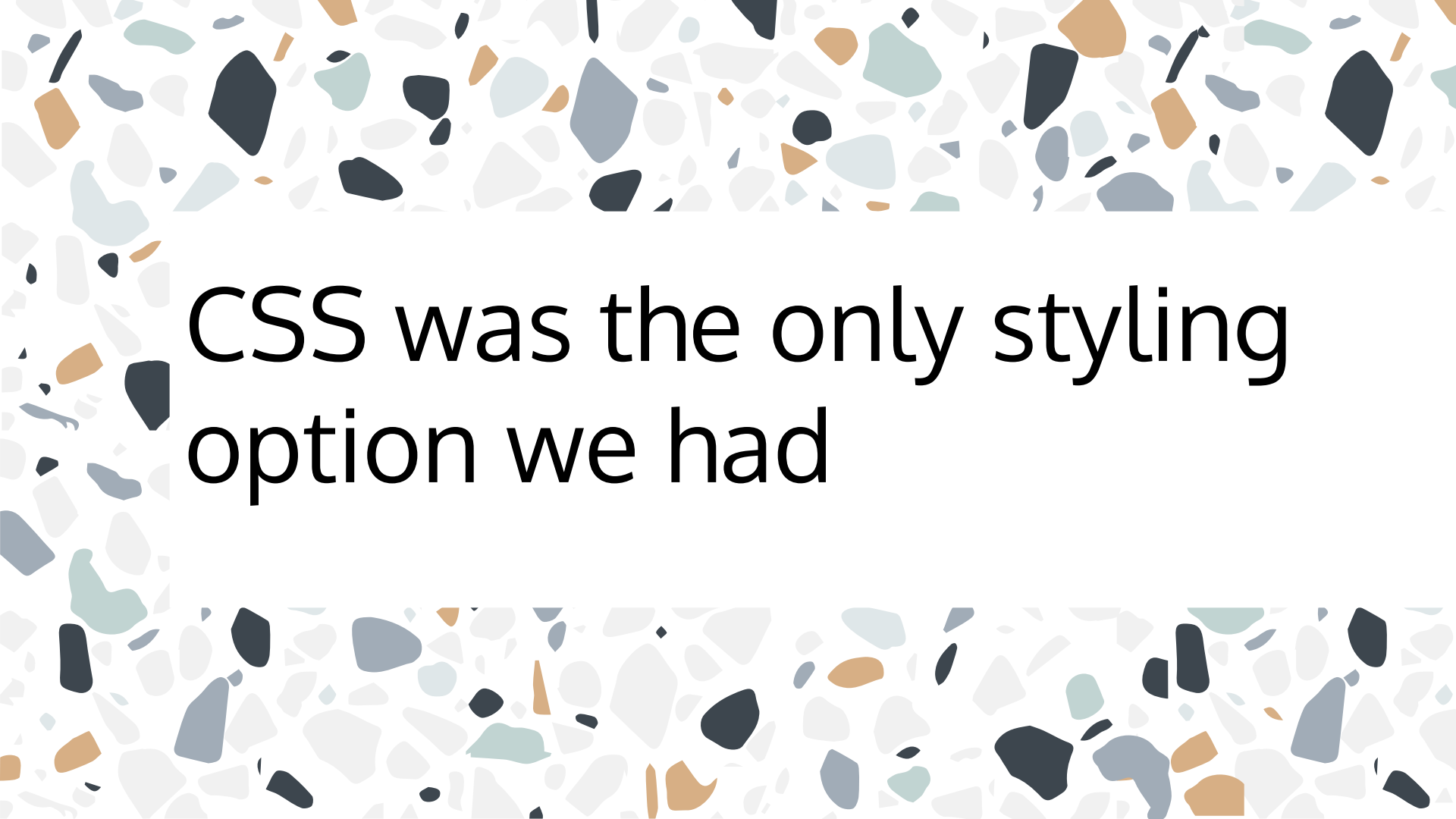
BY KAREN SIMISOLA EFEREYAN



I Code
I Write
I speak
karenefereyan@gmail.com
karene.netlify.app

A decorative border made of small, irregularly shaped pebbles in various colors (dark blue, light blue, orange, and grey) surrounds the central text.

Styled Components

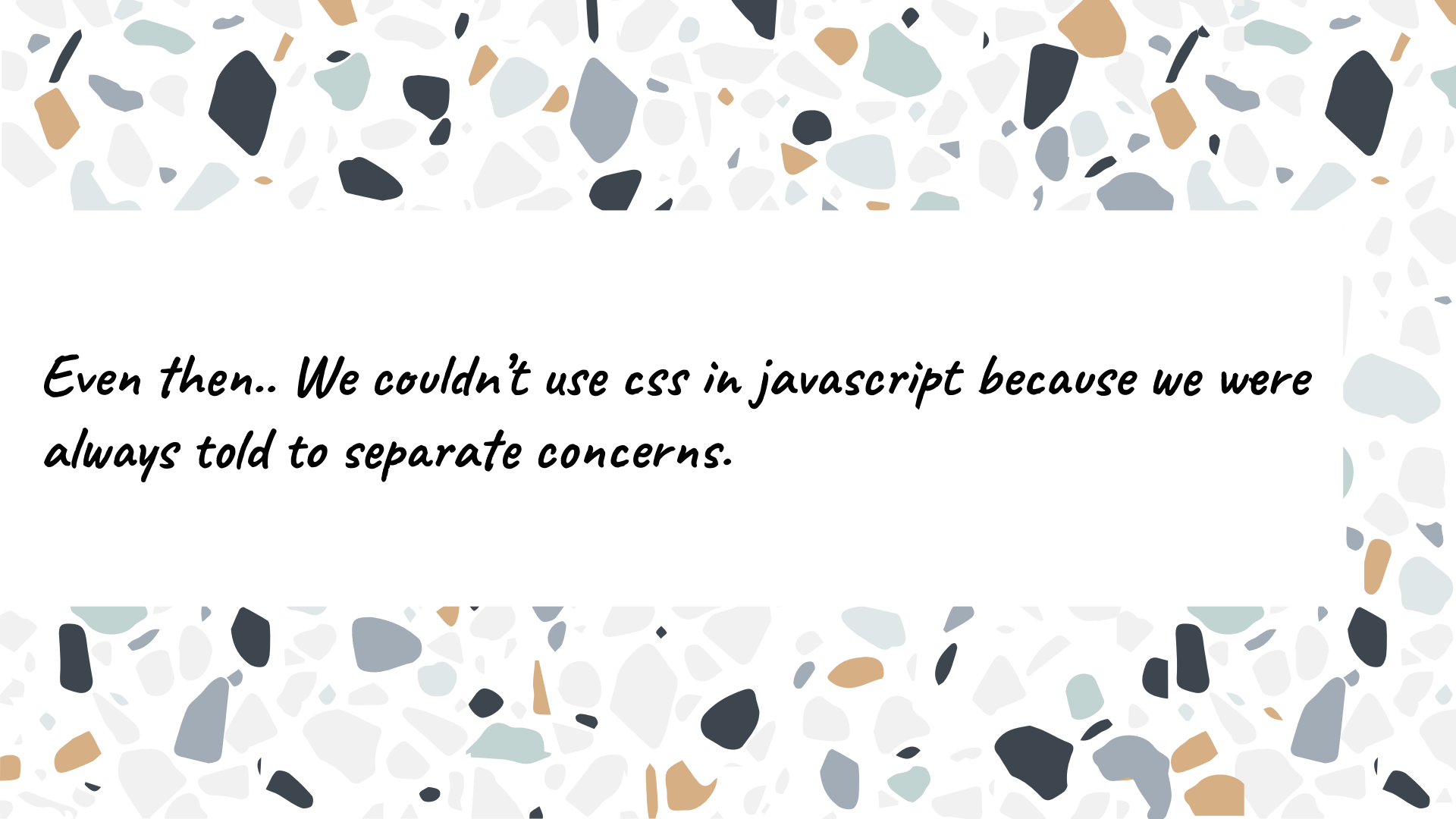


CSS was the only styling
option we had

CSS PREPROCESSORS TO THE RESCUE

- *SASS*
- *STYLUS*
- *LESS*





Even then.. We couldn't use css in javascript because we were always told to separate concerns.



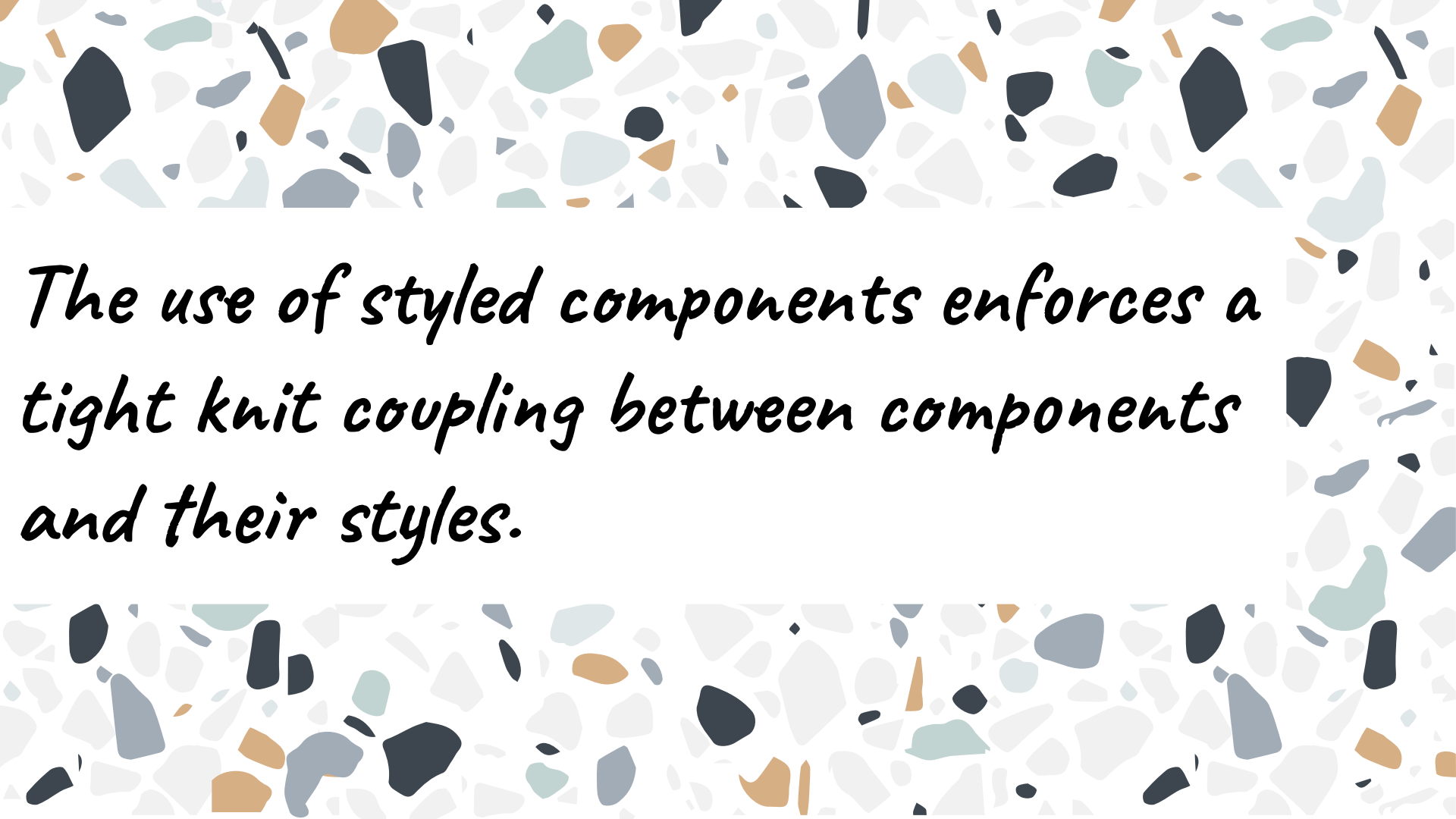
Why?

*Because React Components
Were Written in Javascript*

*And React Components
Should Be Self
Sufficient And Reusable*

*Styled Components are visual
primitives for the component age*





The use of styled components enforces a tight knit coupling between components and their styles.

UNIQUE CLASSNAMES

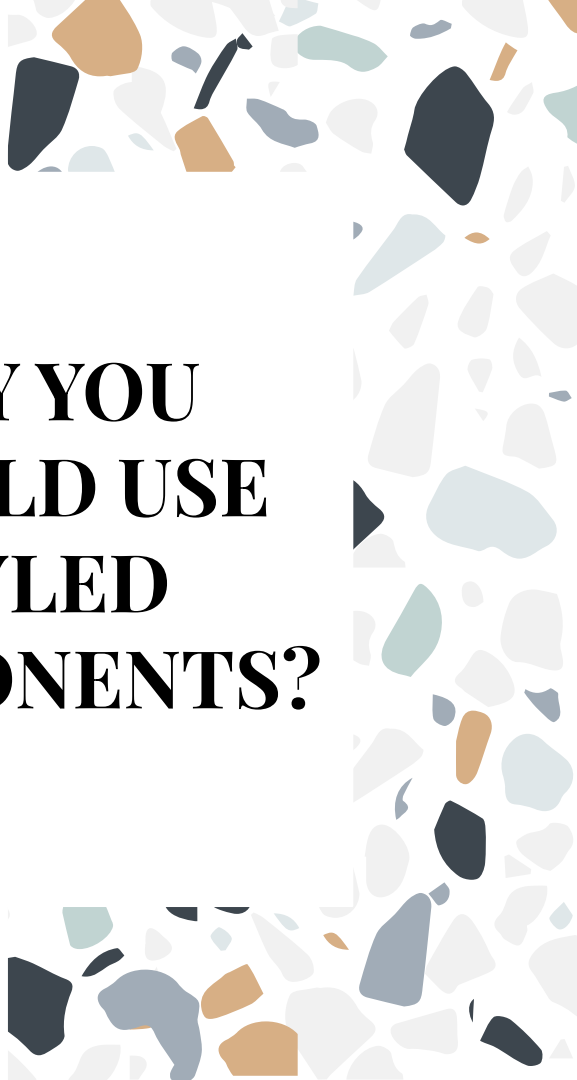
Get rid of unnecessary divs that cause unnecessary confusion. Use clear and perfectly understandable classnames.

STYLE SCOPING

Make individual components self sufficient by creating styles scoped to that particular component.. Even add props to expand your options.

FEATURE RICH

Use all features of Css like media queries, pseudo selectors etc.



**WHY YOU
SHOULD USE
STYLED
COMPONENTS?**

INSTALLATION

- *Npm install --save styled-components(as a development dependency)*
- *Yarn add styled-components*
- *Npm install styled-components*

Styled Component Syntax

Import styled from 'styled-components'

```
Const MyStyledButton = styled.button`
```

```
Background:transparent;
```

```
Color:hotpink;
```

```
Border: 2px solid hotpink;
```

```
,
```

```
Function MyComponent(){
```

```
Return <MyStyledButton>I'm a Reusable Button Component</MyStyledButton>
```

```
}
```

A decorative border made of irregular, colorful stone chips in shades of grey, teal, orange, and dark blue, framing the central text.

confusing?

Let's break it down

Regular HTML Equivalent

```
Button{  
  Background:transparent;  
  Color:hotpink;  
  border:2px solid hotpink;  
  
}
```

```
<button>I am a not easily reusable button!</button>
```





Styled Components are React Components right?

Let's prop it up

A decorative border at the top and bottom of the slide, featuring a pattern of colorful, irregular shapes resembling pebbles or stones in shades of grey, blue, orange, and teal.

*Using props, we can dynamically
and easily extend component
styling*

Import styled from 'styled-components'

```
Const MyStyledButton = styled.button`
```

```
Background:transparent;
```

```
Color:hotpink;
```

```
Border: 2px solid hotpink;
```

```
Padding: ${props => props.size === "big" ? "10px 20px" : "5px 10px"}
```

```
`
```

```
Function MyComponent(){
```

```
Return (
```

```
<React.Fragment>
```

```
<MyStyledButton size = "big" type = "button">I am a big sized
```

```
button</MyStyledButton>
```

```
<MyStyledButton size = "medium" type = "button">I am a medium sized
```

```
button</MyStyledButton>
```

```
</React.Fragment>
```

```
)
```

```
}
```



*We can even inherit and restyle
styled components*



Remember our good old friend,
spread operator(...) from
Javascript?



Import styled from 'styled-components'

```
Const BigPaddedContained = styled.section`
```

```
Width: 100%;
```

```
Max-width:1024px;
```

```
Margin:0 auto;
```

```
Padding: 10px 50px;
```

```
`
```

```
Const SmallPaddedContainer = styled.section `
```

```
Width: 100%;
```

```
Max-width: 1024px;
```

```
Margin: 0 auto;
```

```
Padding: 10px 20px;
```

```
`
```



 *Should We Do That?*

DRY

```
import styled from 'styled-components';
```

```
Const BigPaddedContainer = styled.section`
```

```
Width: 100%;
```

```
Max-width:1024px;
```

```
Margin:0 auto;
```

```
Padding: 10px 50px;
```

```
`
```

```
Const SmallPaddedContainer = styled(BigPaddedContainer)`
```

```
Padding: 10px 20px;
```

```
`
```

```
Function Home(){
```

```
Return (
```

```
<BigPaddedContainer>
```

```
    <h1>The padding around me is great</h1>
```

```
</BigPaddedContainer>
```

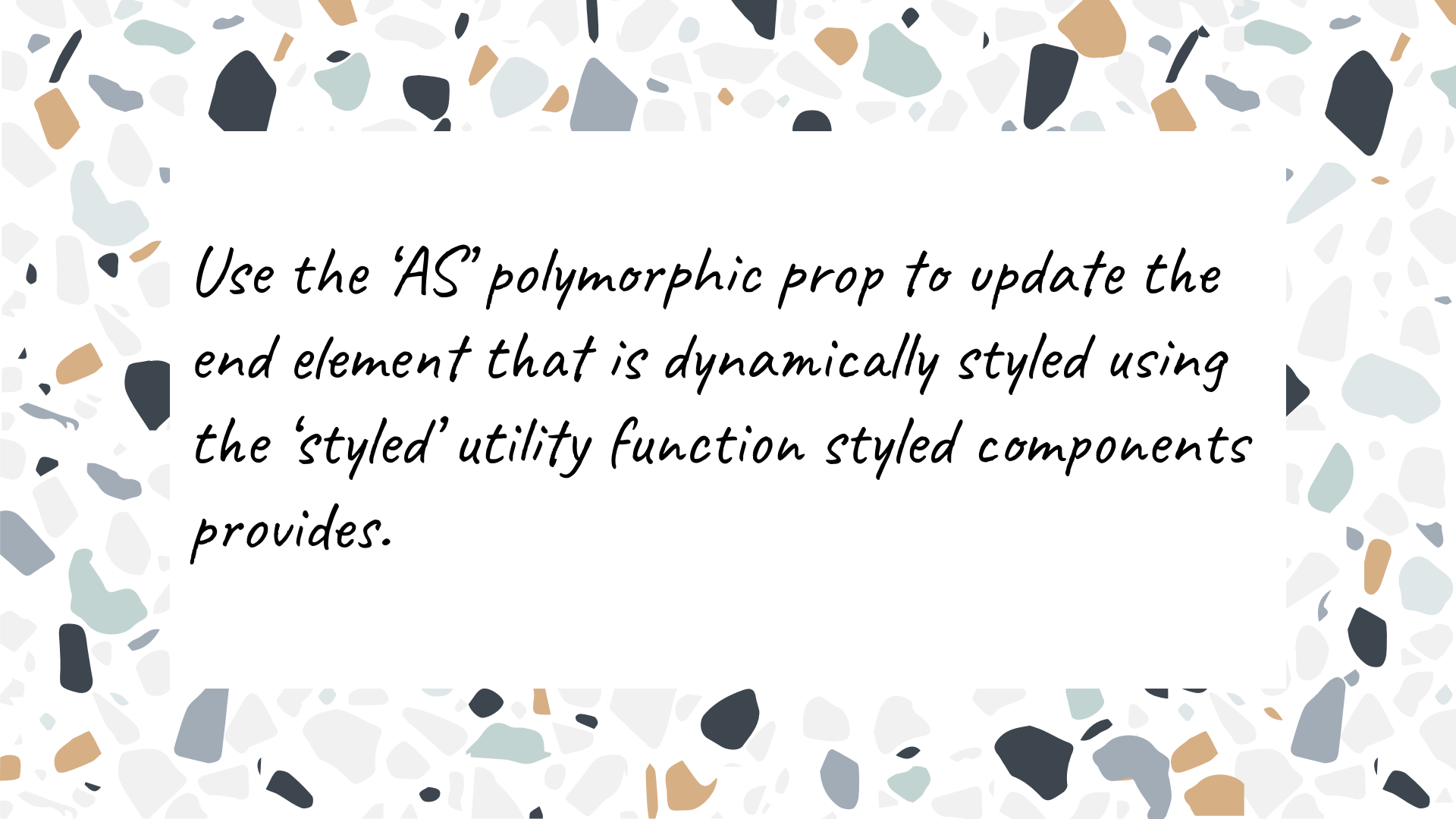
```
)
```

```
Function Contact(){
```

```
Return (
```

```
<SmallPaddedContainer><h1>The paddng around me is smaller</h1></SmallPaddedContainer>
```

```
}
```

A decorative border composed of various colored pebbles or stones in shades of blue, orange, and grey, arranged in a pattern around the central text.

Use the 'AS' polymorphic prop to update the end element that is dynamically styled using the 'styled' utility function styled components provides.

```
Function Contact(){  
  Return (  
    <SmallPaddedContainer as = "div">  
      <h1>I am in a div element with a smaller padding as opposed to the  
component from which I was cloned which is a section with a larger  
padding</h1>  
    </SmallPaddedContainer>  
  )  
}
```



```
Function Contact(){  
  Return (  
    <SmallPaddedContainer as = {BigPaddedContainer}>  
      <h1>This small padded container is also of the end type of  
      BigPaddedContainer which is a section element</h1>  
    </SmallPaddedContainer>  
  )  
}
```

Styled Components also support

1. SCSS like syntax(nesting and more..)
2. Animations(keyframes)
3. Global Styling (CSS Resets)
4. CSS Helper Functions



With power comes responsibility





*Remember to maintain your sanity in your use of
styled components*





Further Reading

- ❖ [Styled Components Documentation](#)
 - ❖ [How to Use Styled Components in React](#)
 - ❖ [Styled Components: Enforcing Best Practices in Component Based Systems](#)
- 



THANK YOU FOR LISTENING

karenefe.netlify.app

