

# Styling components in React Native

Kyrylo Yakovenko

*With React Native, you don't use a special language or syntax for defining styles.*

**NoOcsss**

**NO**

selectors

**NO**

selectors  
cascades

# NO

selectors

cascades

pseudo classes/elements

# NO

selectors

cascades

pseudo classes/elements

media queries

# NO

selectors

cascades

pseudo classes/elements

media queries

units

# NO

selectors

cascades

pseudo classes/elements

media queries

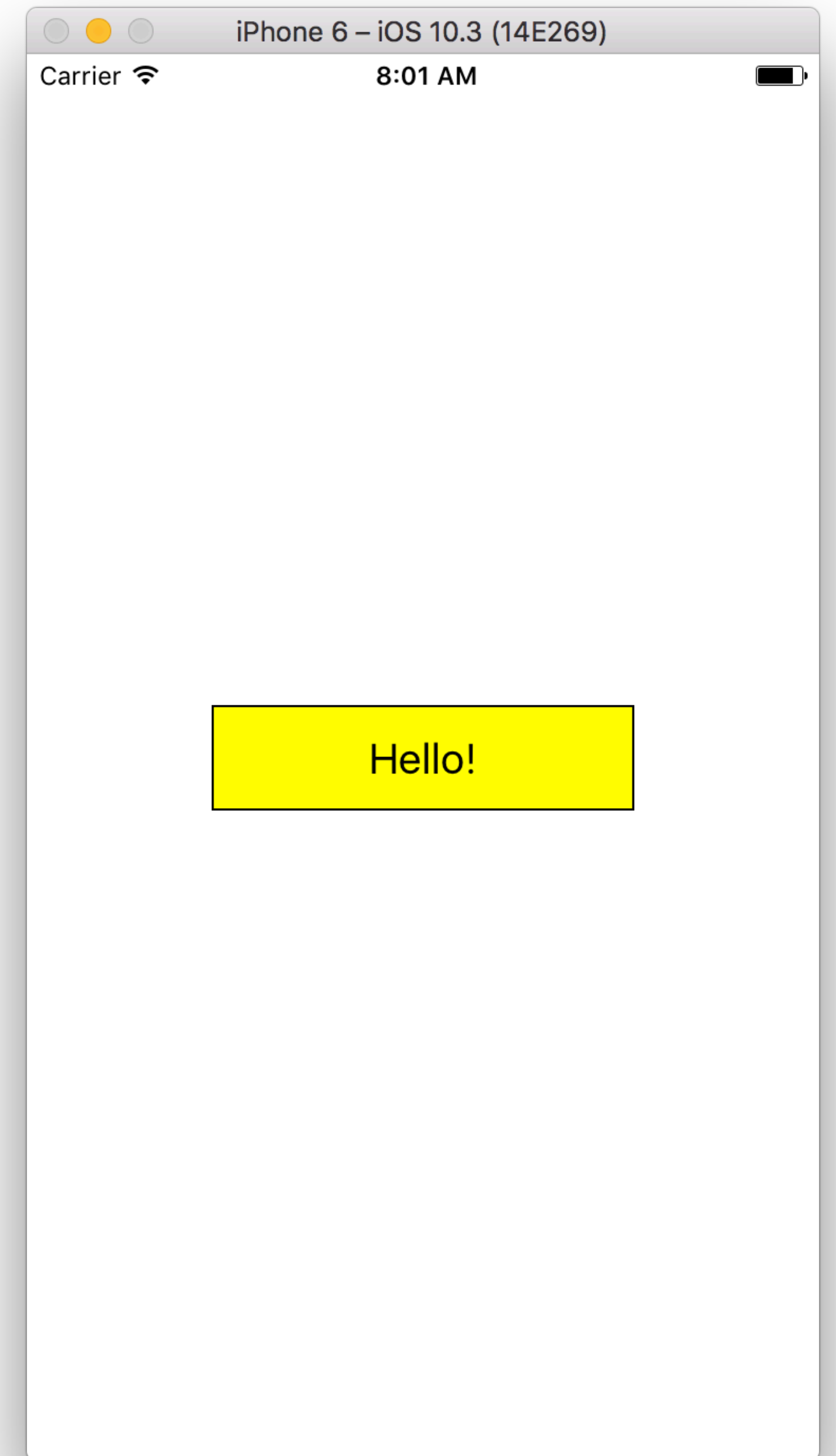
units

shorthands

```
import React from 'react';
import { StyleSheet, Text, View } from 'react-native';

export default class App extends React.Component {
  render() {
    return (
      <View style={styles.container}>
        <View style={styles.box}>
          <Text style={styles.title}>Hello!</Text>
        </View>
      </View>
    );
  }
}
```

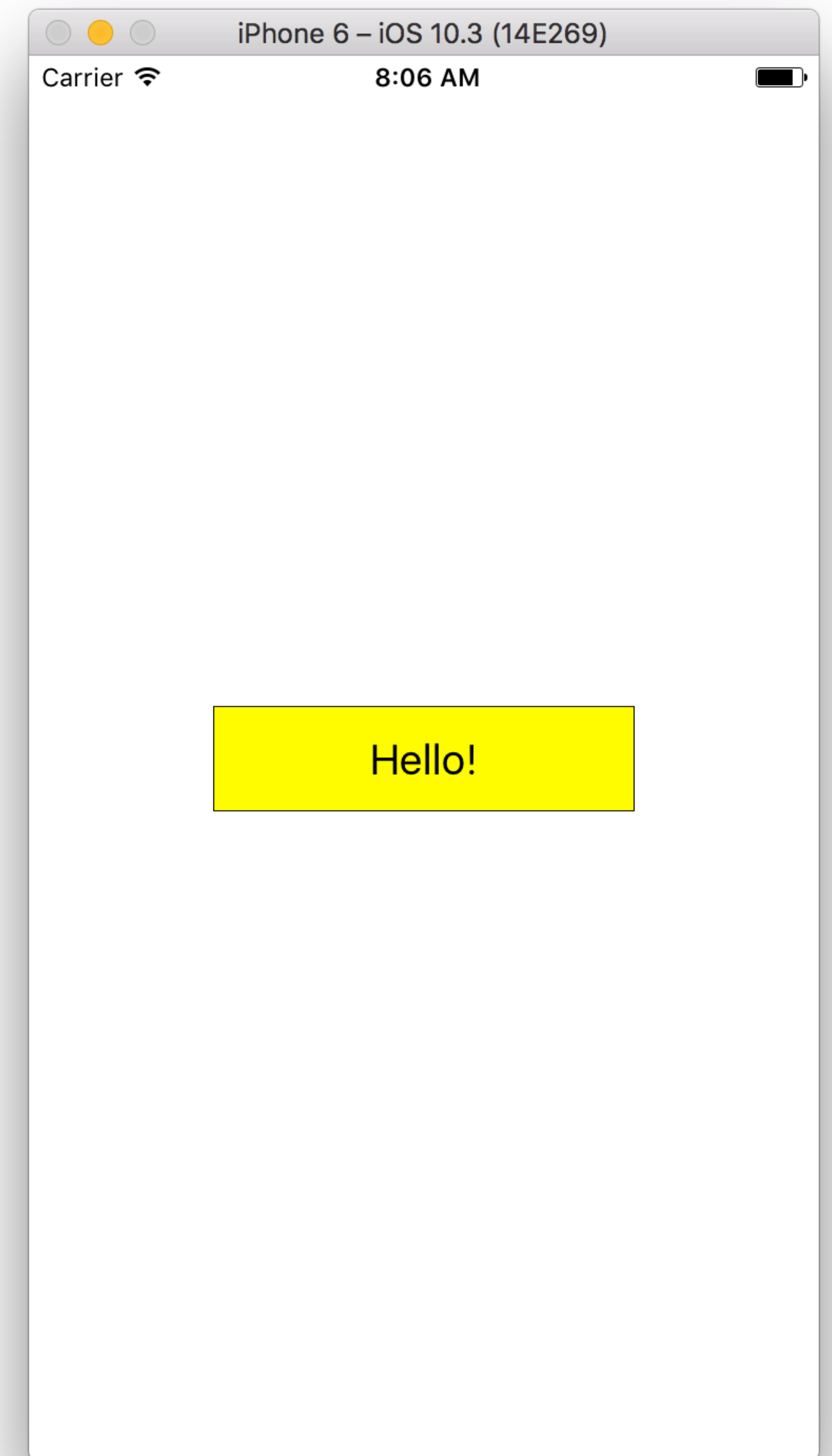
```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
  box: {
    backgroundColor: 'yellow',
    width: 200,
    height: 50,
    borderWidth: 1,
    borderColor: 'black',
    alignItems: 'center',
    justifyContent: 'center',
  },
  title: {
    fontSize: 20
  }
});
```



# Units

All dimensions in React Native are unitless, and represent density-independent pixels.

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
  box: {
    backgroundColor: 'yellow',
    width: 200,
    height: 50,
    borderWidth: StyleSheet.hairlineWidth,
    borderColor: 'black',
    alignItems: 'center',
    justifyContent: 'center',
  },
  title: {
    fontSize: 20
  }
});
```

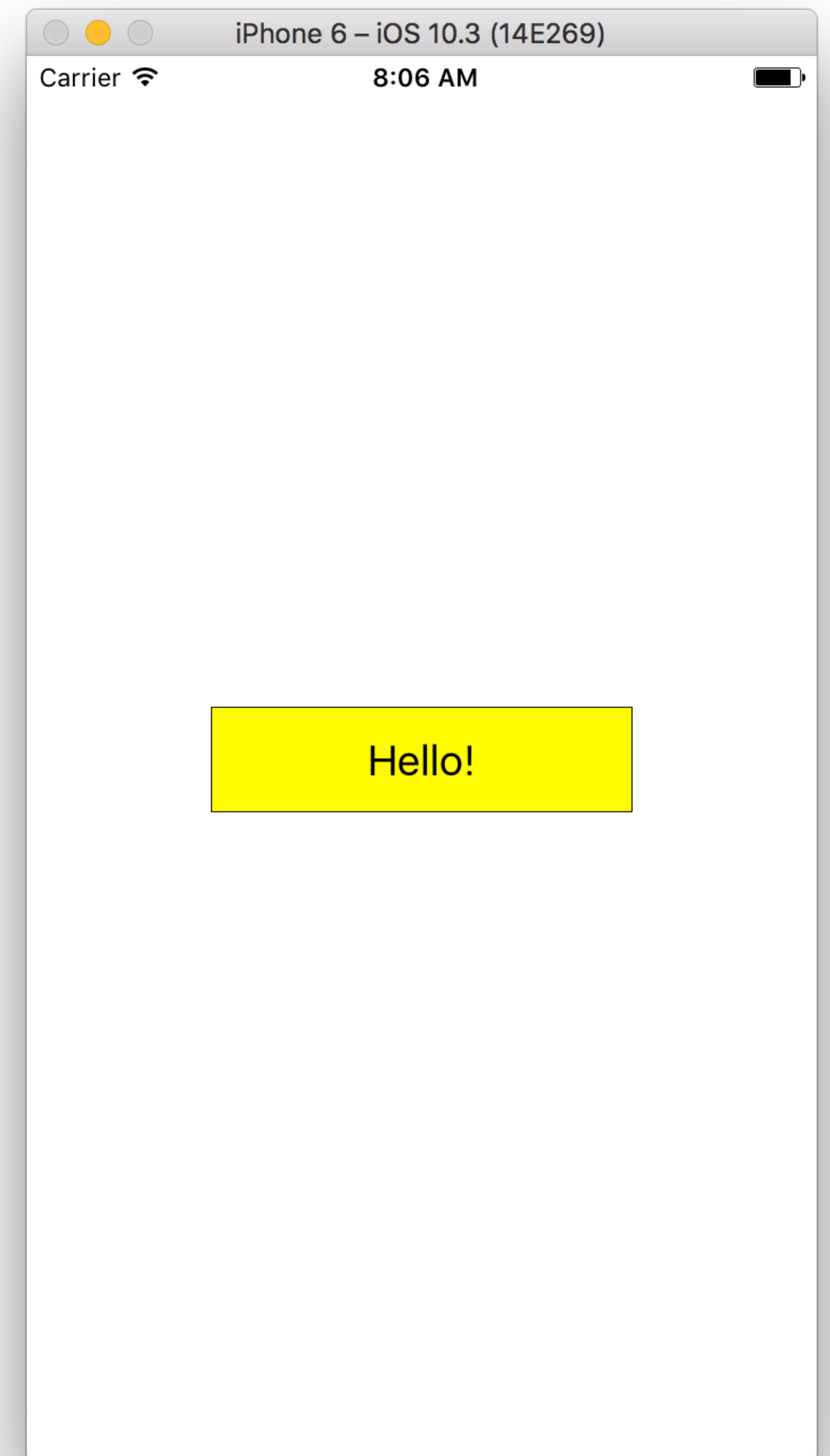


# Utils

With ES6 spread syntax

```
function rect(width, height) {  
  return {  
    width,  
    height  
  };  
}
```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
  box: {
    backgroundColor: 'yellow',
    ...rect(200, 50),
    borderWidth: StyleSheet.hairlineWidth,
    borderColor: 'black',
    alignItems: 'center',
    justifyContent: 'center',
  },
  title: {
    fontSize: 20
  }
});
```

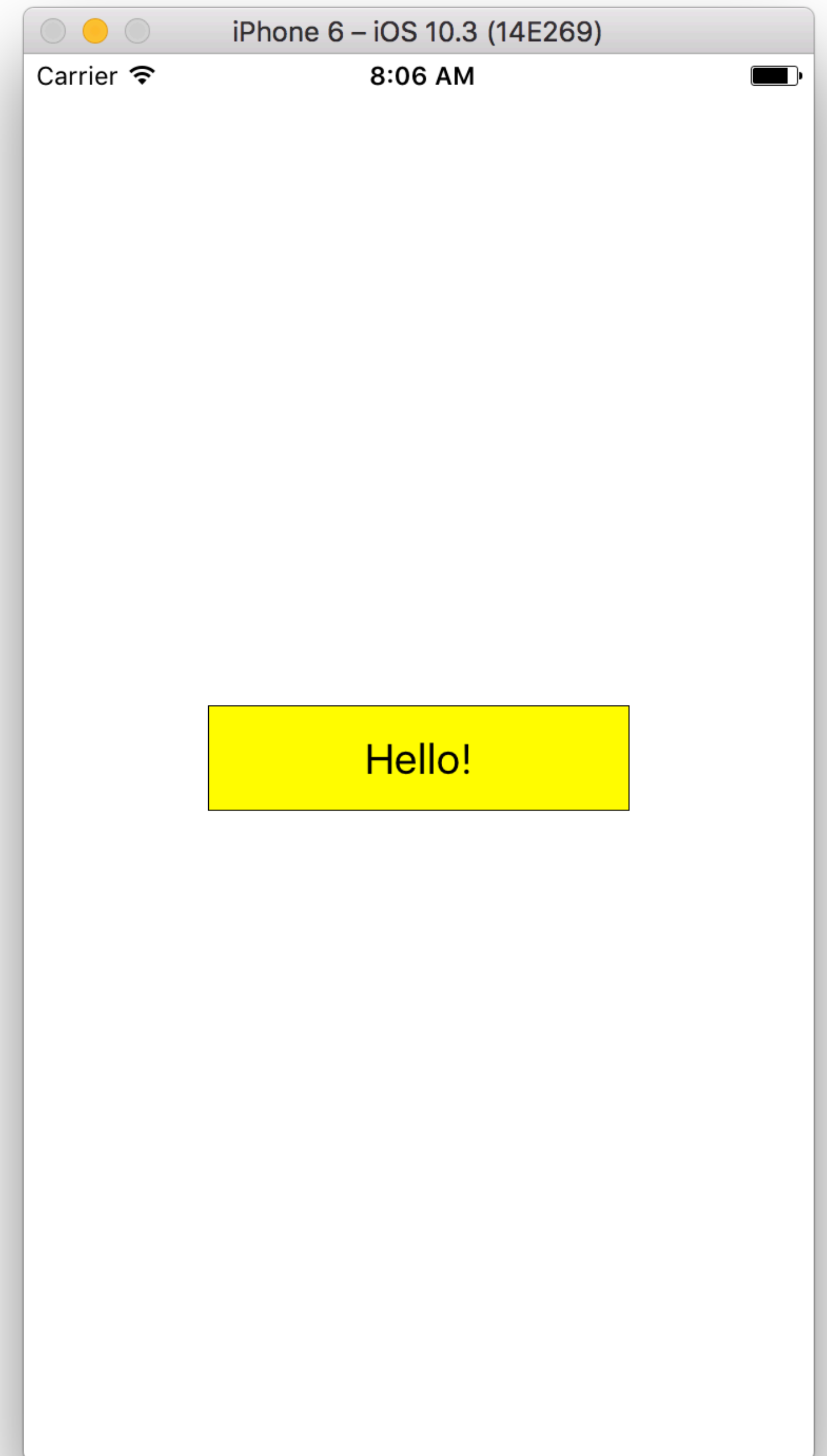


# Shorthand properties

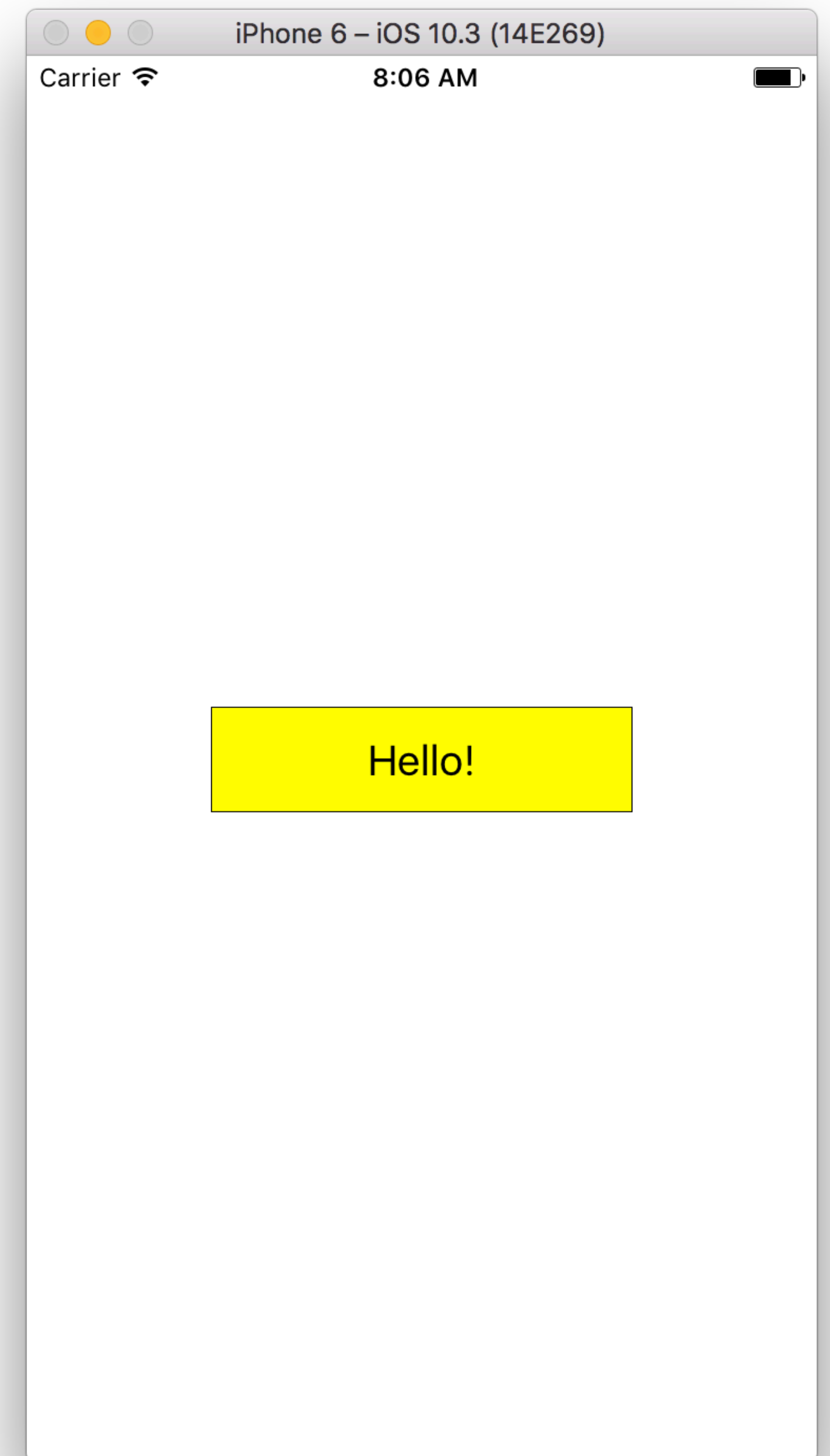
```
function contentPosition(a, j) {
  if (arguments.length === 1 && a === 'center') {
    return {
      alignItems: a,
      justifyContent: a
    };
  }
  return {
    alignItems: a,
    justifyContent: j
  };
}
```

```
function border(width, color = '#000', style = 'solid') {
  return {
    borderWidth: width,
    borderColor: color,
    borderStyle: style
  };
}
```

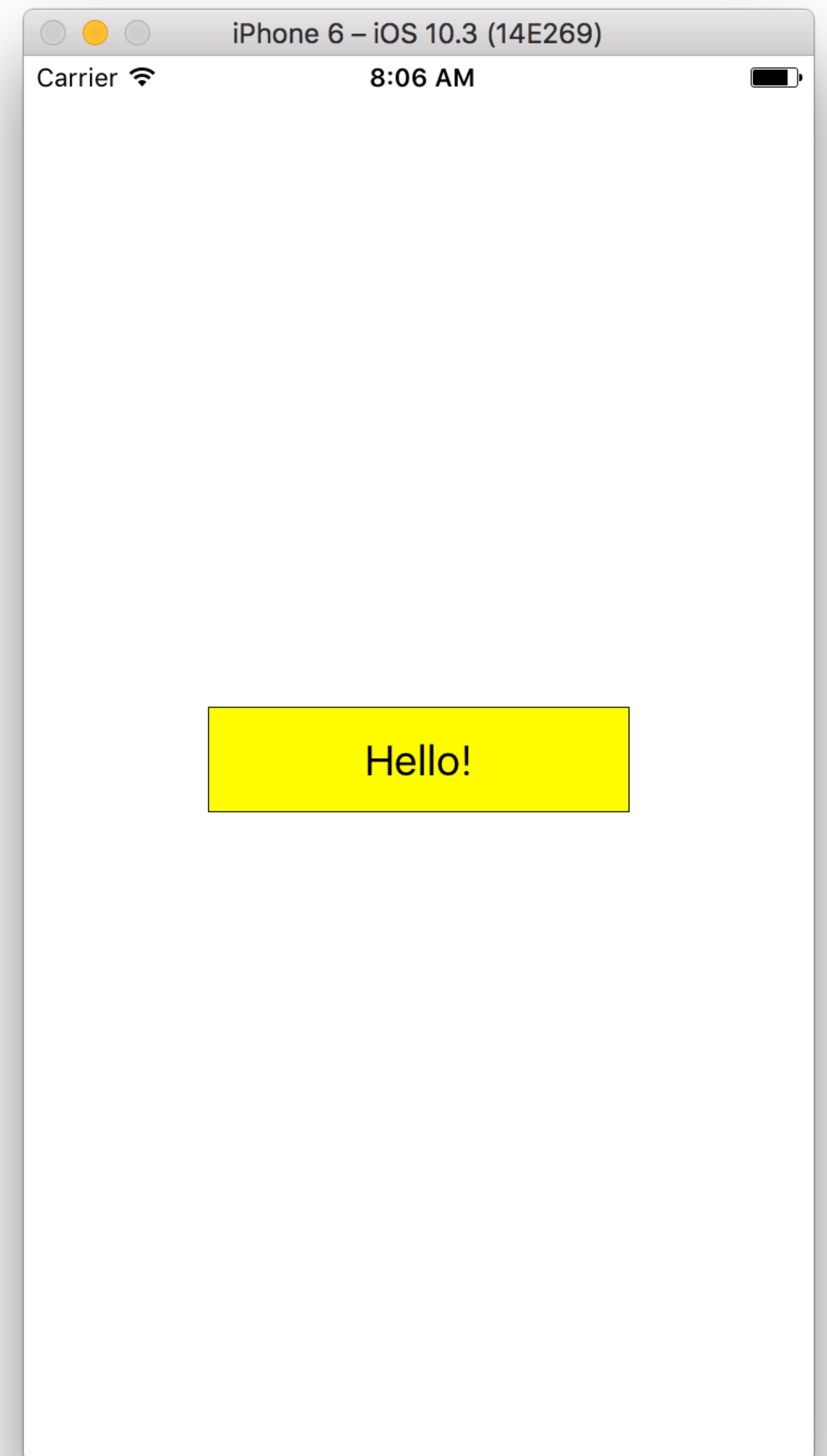
```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    ...contentPosition('center')
  },
  box: {
    backgroundColor: 'yellow',
    ...rect(200, 50),
    ...border(StyleSheet.hairlineWidth),
    ...contentPosition('center')
  },
  title: {
    fontSize: 20
  }
});
```



```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
  box: {
    backgroundColor: 'yellow',
    width: 200,
    height: 50,
    borderWidth: StyleSheet.hairlineWidth,
    borderColor: 'black',
    alignItems: 'center',
    justifyContent: 'center',
  },
  title: {
    fontSize: 20
  }
});
```

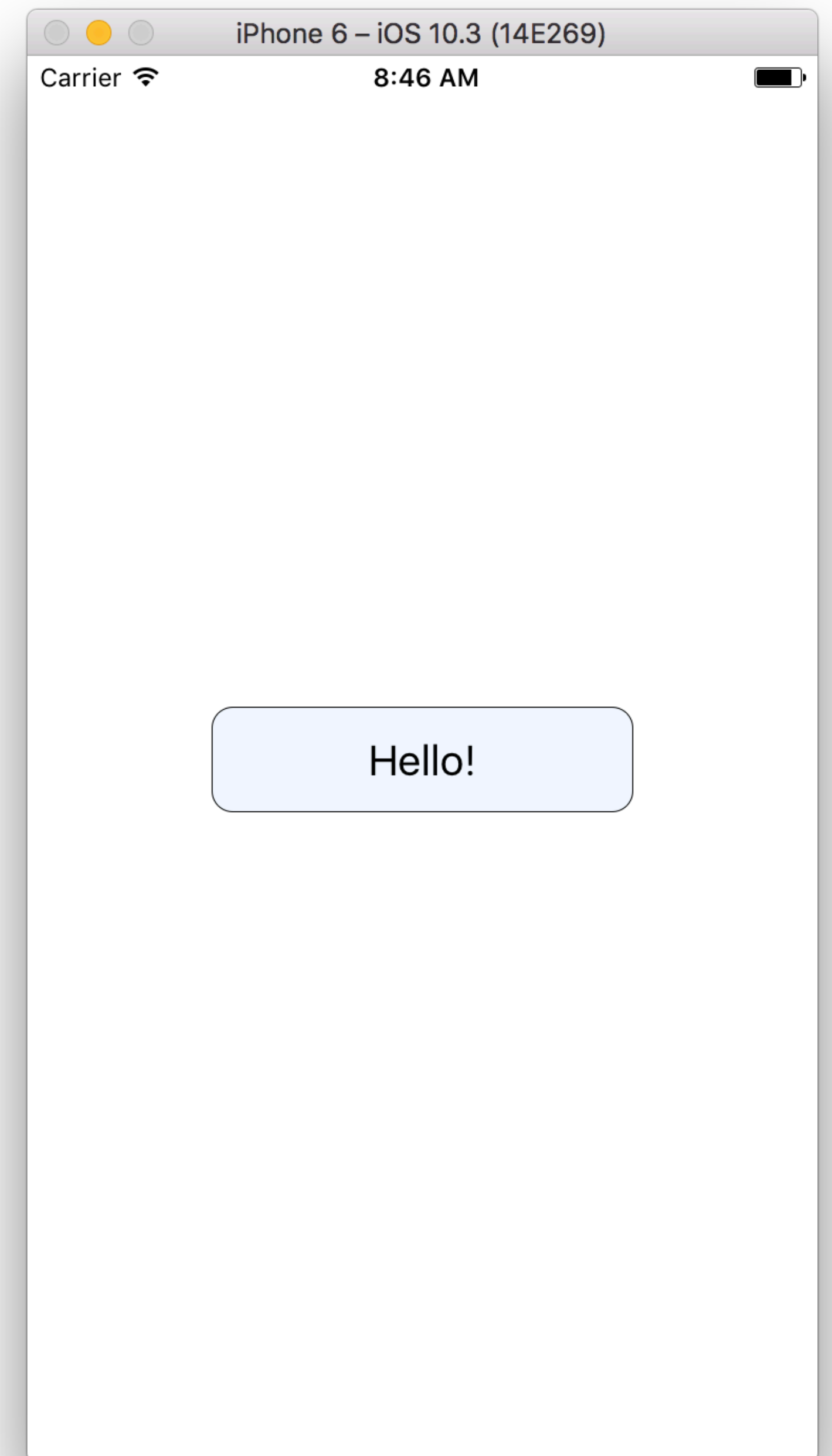


```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    ...contentPosition('center')
  },
  box: {
    backgroundColor: 'yellow',
    ...rect(200, 50),
    ...border(StyleSheet.hairlineWidth),
    ...contentPosition('center')
  },
  title: {
    fontSize: 20
  }
});
```



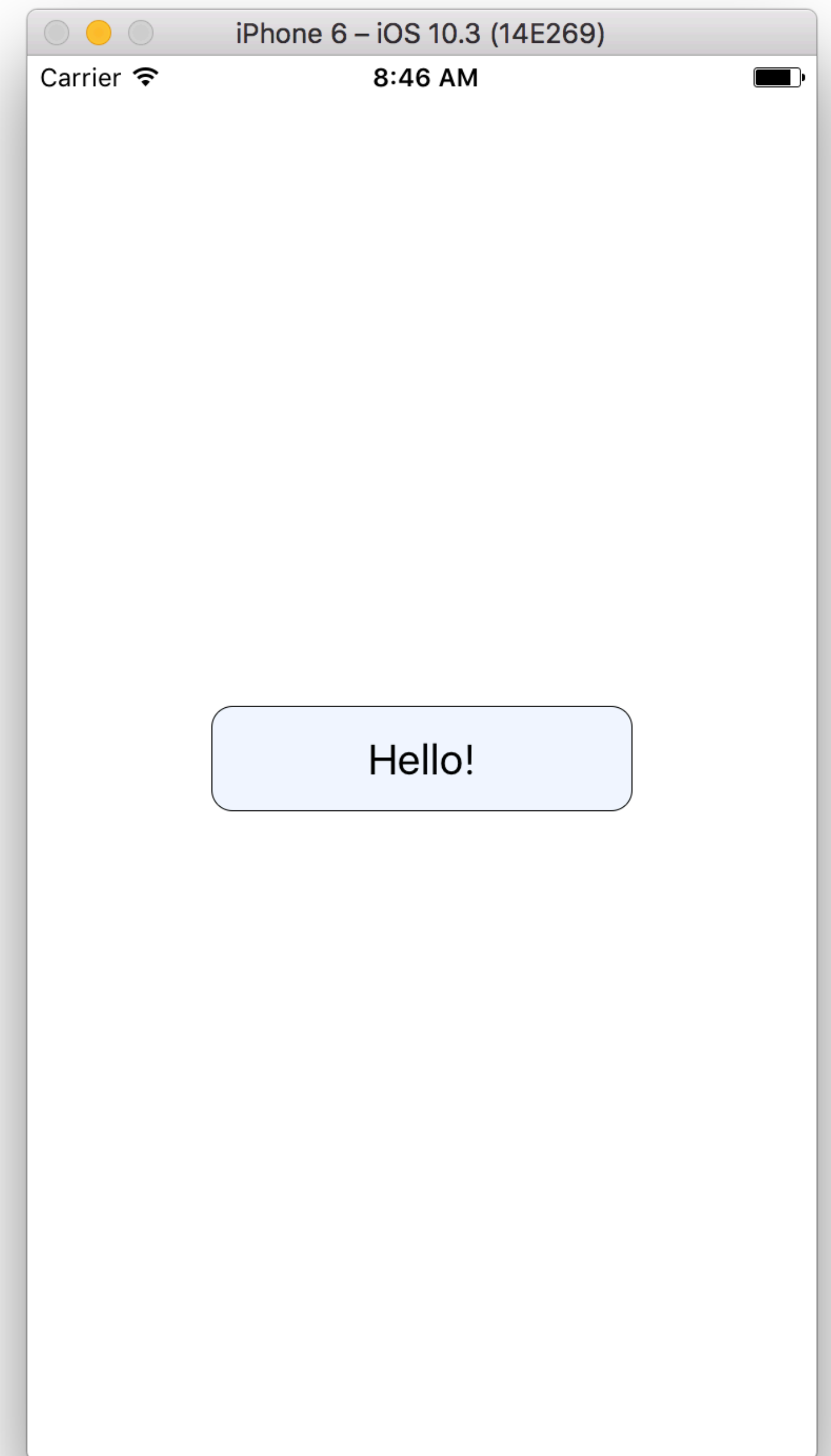
**Media\* queries**

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    ...contentPosition('center')
  },
  box: {
    backgroundColor: (Platform.OS === 'ios'
      ? 'aliceblue'
      : 'green'),
    borderRadius: Platform.OS === 'ios' ? 10: 0,
    ...rect(200, 50),
    ...border(StyleSheet.hairlineWidth),
    ...contentPosition('center')
  },
  title: {
    fontSize: 20
  }
});
```



```
function isIos(styles) {  
  return Platform.OS === 'ios' ? styles : null;  
}
```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    ...contentPosition('center')
  },
  box: {
    backgroundColor: 'green',
    ...isIos({
      backgroundColor: 'aliceblue',
      borderRadius: 10
    }),
    ...rect(200, 50),
    ...border(StyleSheet.hairlineWidth),
    ...contentPosition('center')
  },
  title: {
    fontSize: 20
  }
});
```

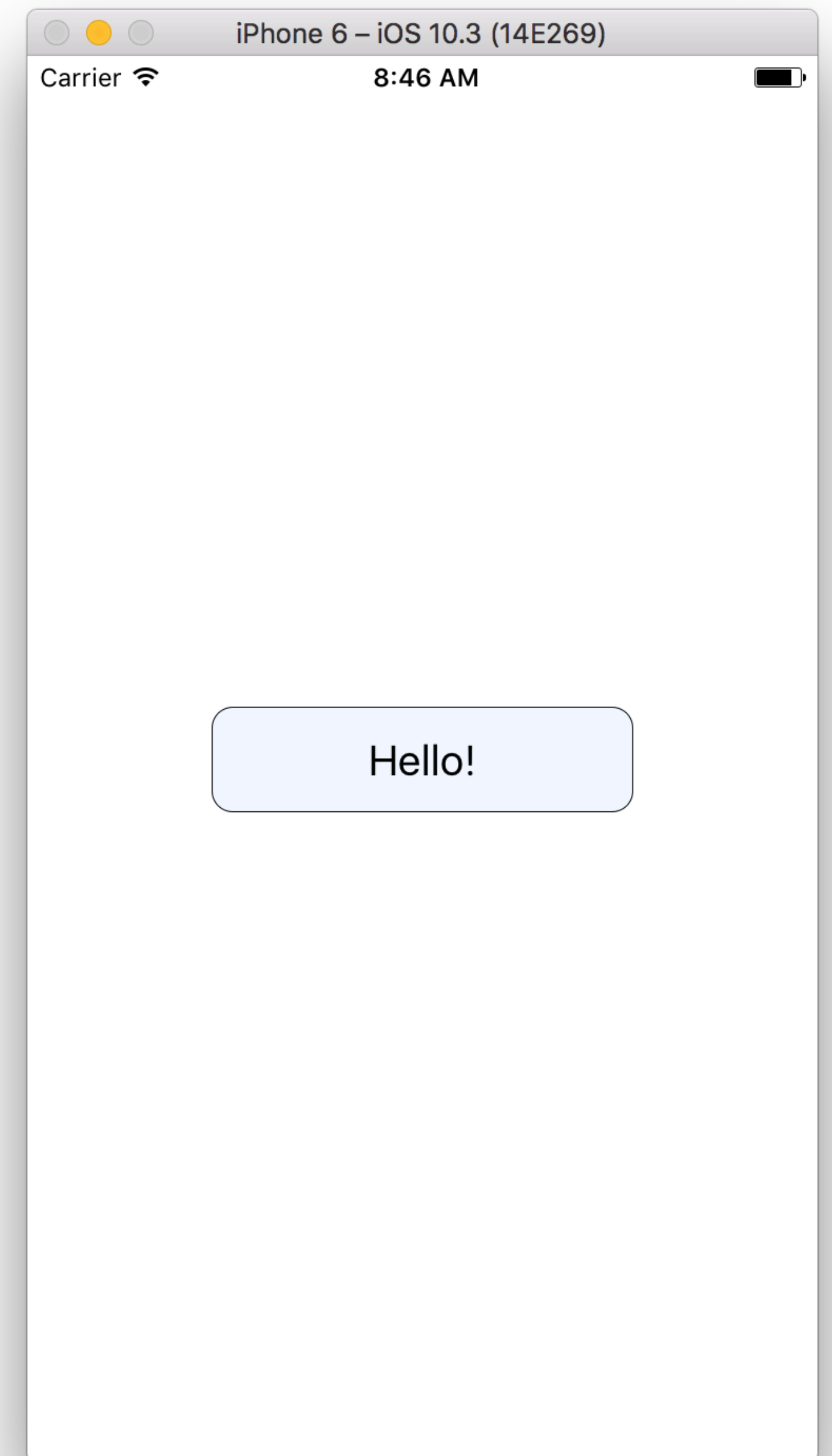


```
function isIos(styles) {  
  const ios = Platform.OS === 'ios'  
  return !styles ? ios : ios ? styles : null;  
}
```

```
function isLandscape(styles) {  
  const { width, height } = Dimensions.get('window');  
  const landscape = width > height;  
  return !styles ? landscape : landscape ? styles : null;  
}
```

```
function all(...args) {  
  const styles = args.pop();  
  return args.every(f => f()) ? styles : null;  
}
```

```
const styles = {
  container: {
    flex: 1,
    backgroundColor: '#fff',
    ...contentPosition('center')
  },
  box: {
    backgroundColor: 'green',
    ...all(isIos, isLandscape, {
      display: 'none'
    }),
    ...rect(200, 50),
    ...border(StyleSheet.hairlineWidth),
    ...contentPosition('center')
  },
  title: {
    fontSize: 20
  }
};
```



**Small bonus**

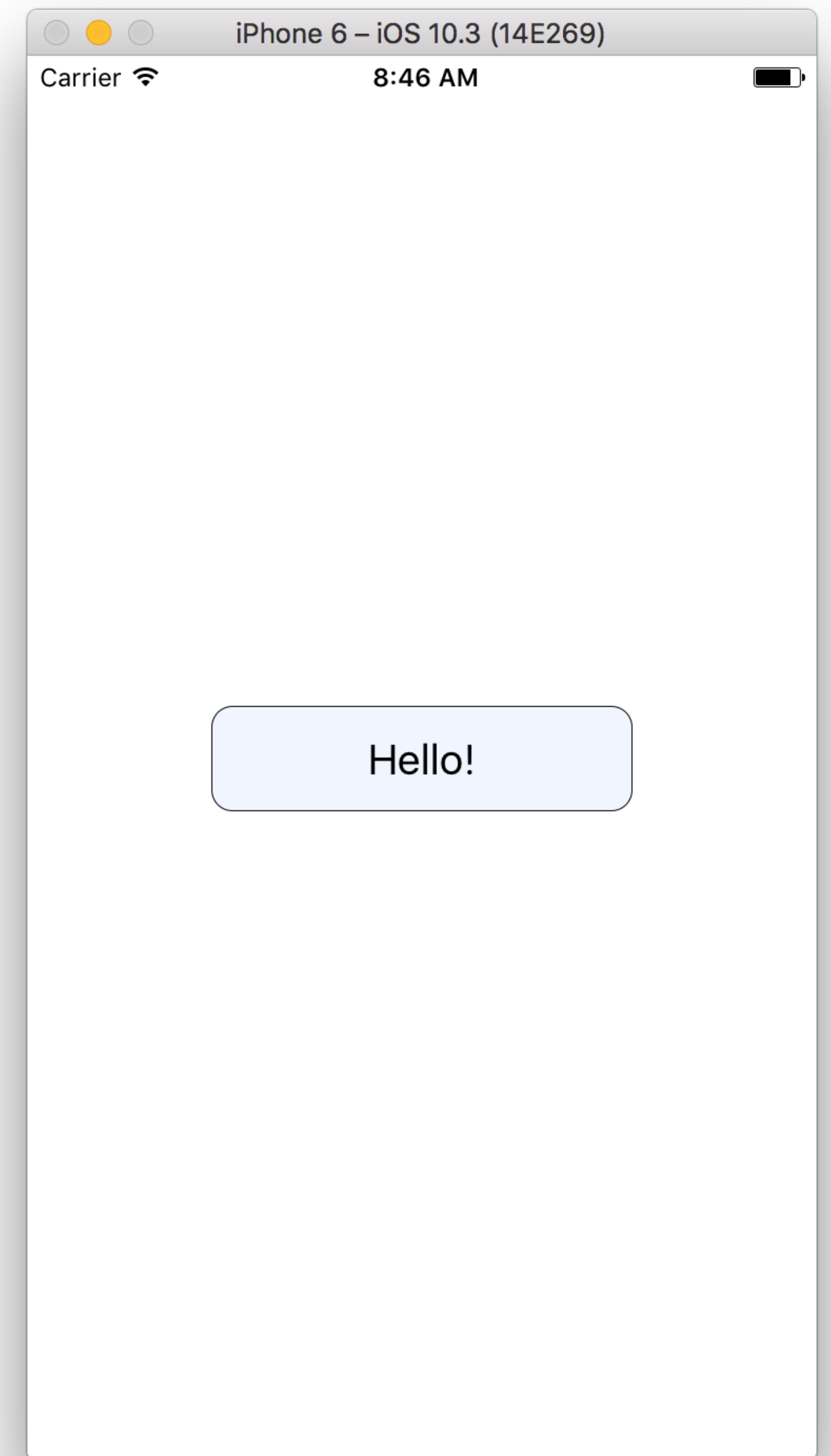
```
{  
  "presets": ["babel-preset-expo"],  
  "env": {  
    "development": {  
      "plugins": [  
        "transform-react-jsx-source",  
        "babel-plugin-transform-cstag"  
      ]  
    }  
  }  
}
```

```
import css from './ssx/src';
```

```
...
```

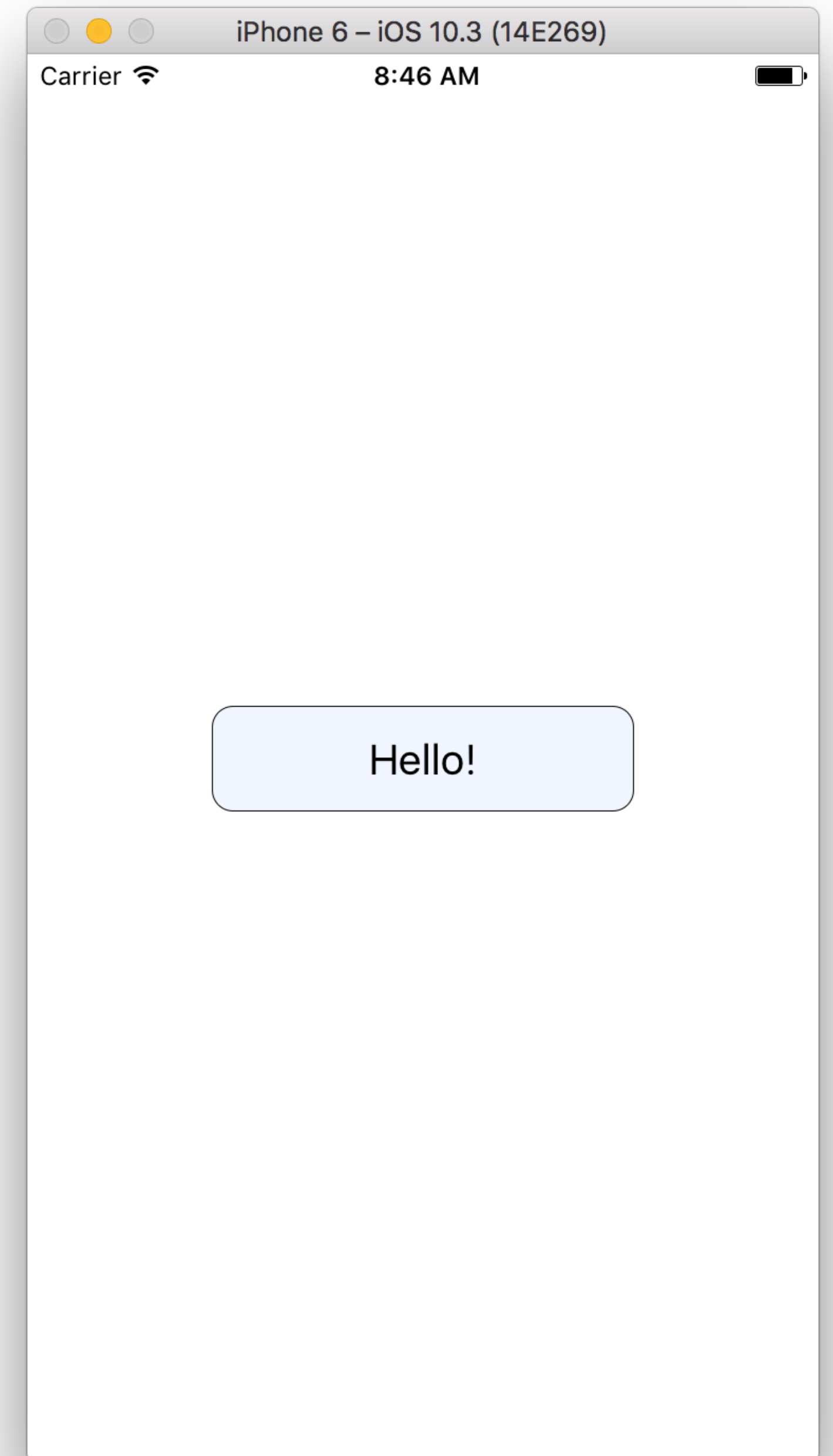
```
css.use({rect, border, contentPosition, isIos});
```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    ...contentPosition('center')
  },
  box: {
    backgroundColor: 'green',
    ...isIos({
      backgroundColor: 'aliceblue',
      borderRadius: 10
    }),
    ...rect(200, 50),
    ...border(StyleSheet.hairlineWidth),
    ...contentPosition('center')
  },
  title: {
    fontSize: 20
  }
});
```

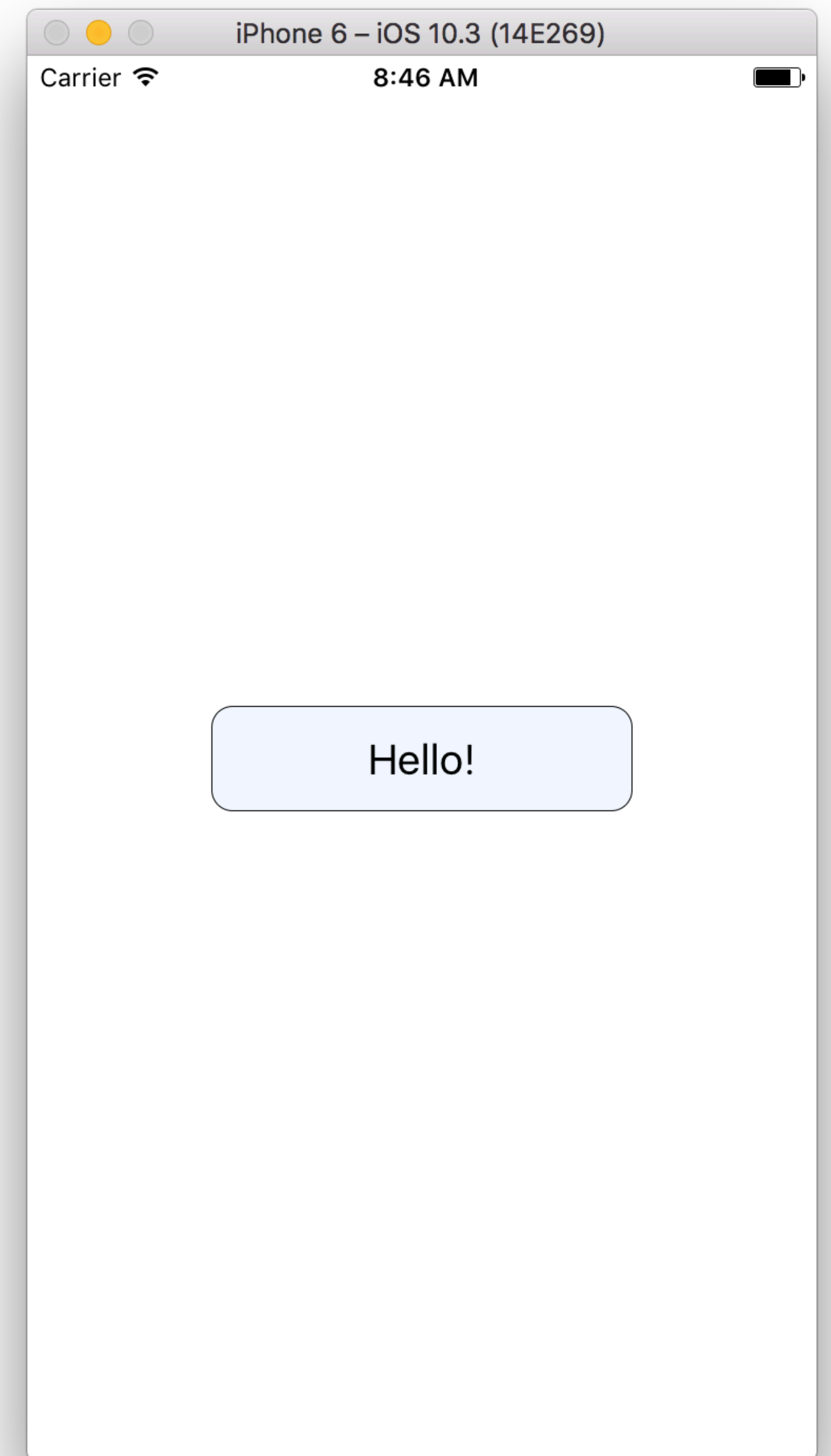


**CSS-like syntax**

```
const styles = css`
  .container {
    flex: 1;
    background-color: #fff;
    content-position: center;
  }
  .box {
    background-color: green;
    rect: 200 50;
    border: 1px;
    content-position: center;
    @media ios {
      background-color: aliceblue;
      border-radius: 10;
    }
  }
  .title {
    font-size: 20;
  }
`
```



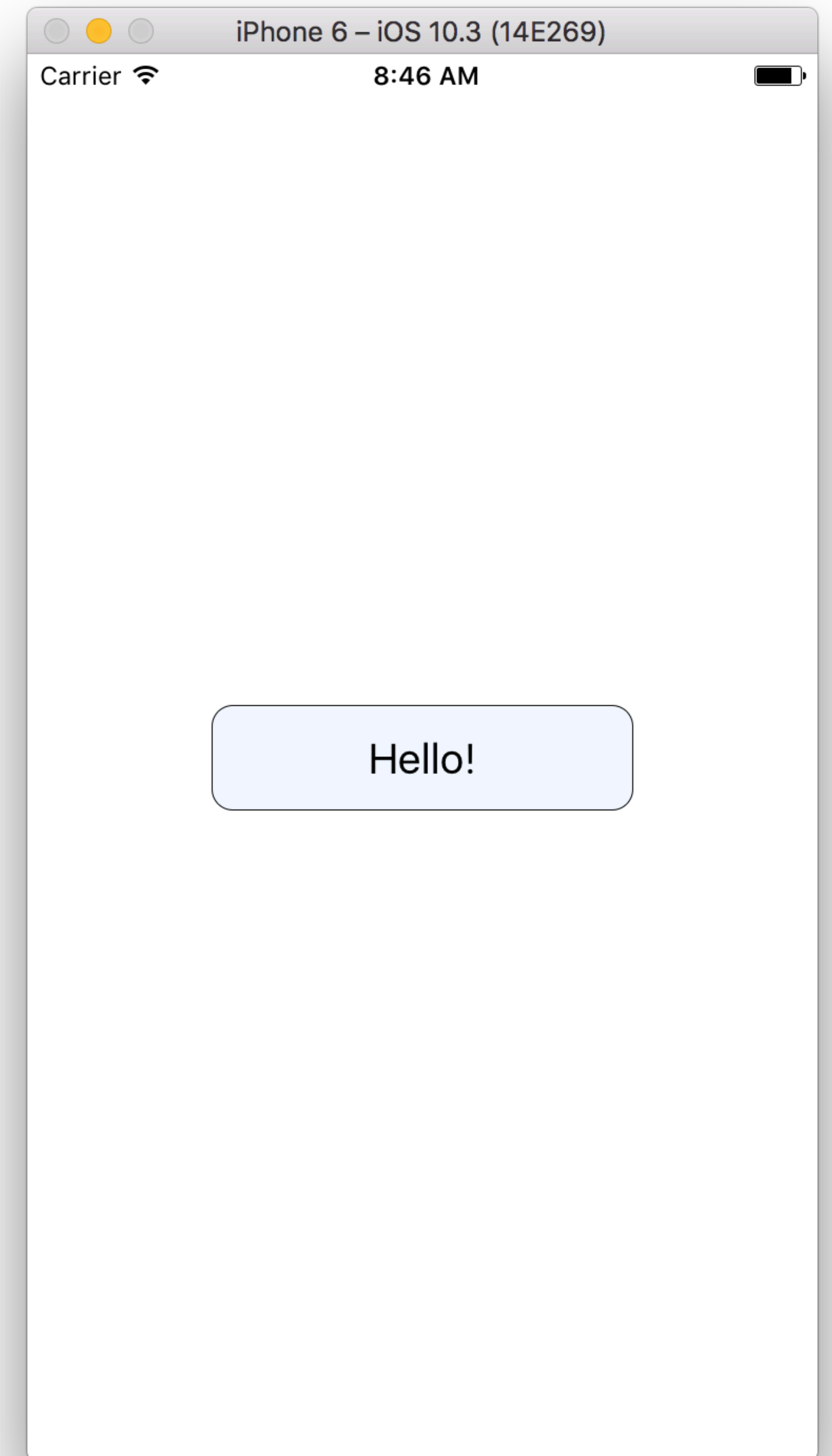
```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    ...contentPosition('center')
  },
  box: {
    backgroundColor: 'green',
    ...isIos({
      backgroundColor: 'aliceblue',
      borderRadius: 10
    }),
    ...rect(200, 50),
    ...border(StyleSheet.hairlineWidth),
    ...contentPosition('center')
  },
  title: {
    fontSize: 20
  }
});
```



```
export default class App extends React.Component {  
  render() {  
    return (  
      <View style={styles.container}>  
        <Button>Hello!</Button>  
      </View>  
    );  
  }  
}
```

```
class Button extends React.Component {
  render() {
    const { background, title } = styles();
    return (
      <View style={background}>
        <Text style={title}>
          {this.props.children}
        </Text>
      </View>
    );
  }
}
```

```
const styles = css`
  .background {
    background-color: green;
    rect: 200 50;
    border: 1px;
    content-position: center;
    @media ios {
      background-color: aliceblue;
      border-radius: 10;
    }
  }
  .title {
    font-size: 20;
  }
`;
```

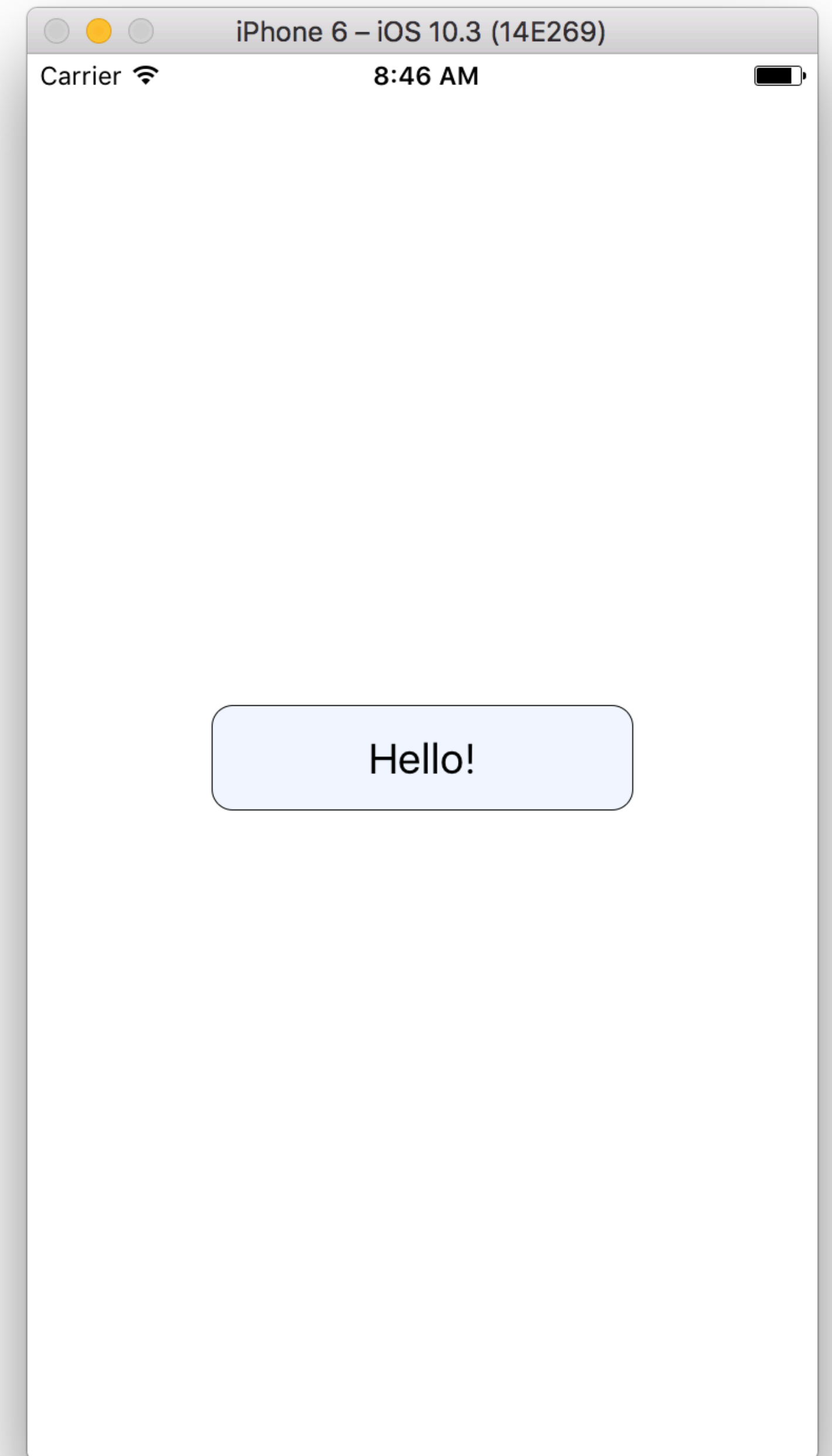


**Context**

```
export default class App extends React.Component {  
  render() {  
    return (  
      <View style={styles.container}>  
        <Button size="20">Hello!</Button>  
      </View>  
    );  
  }  
}
```

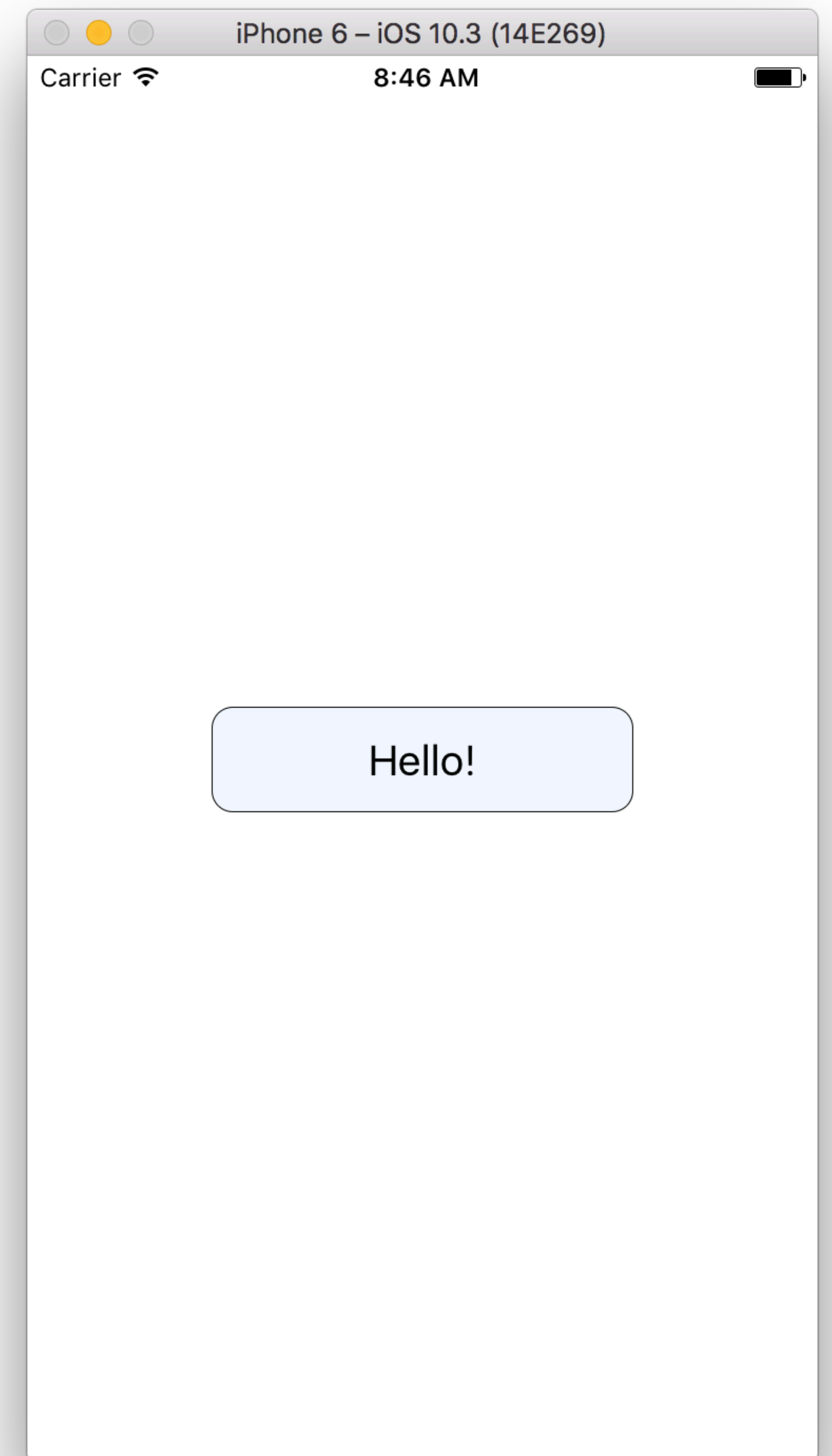
```
class Button extends React.Component {
  render() {
    const { background, title } = styles(this.props);
    return (
      <View style={background}>
        <Text style={title}>
          {this.props.children}
        </Text>
      </View>
    );
  }
}
```

```
const styles = css`
  .background {
    background-color: green;
    rect: 200 50;
    border: 1px;
    content-position: center;
    @media ios {
      background-color: aliceblue;
      border-radius: 10;
    }
  }
  .title {
    font-size: `${props => props.size}`;
  }
`;
```



**Tags**

```
const styles = css`
  .background {
    background-color: green;
    rect: 200 50;
    border: 1px;
    content-position: center;
    @media ios {
      background-color: aliceblue;
      border-radius: 10;
    }
  }
  Text.title {
    font-size: ${props => props.size};
  }
`;
```



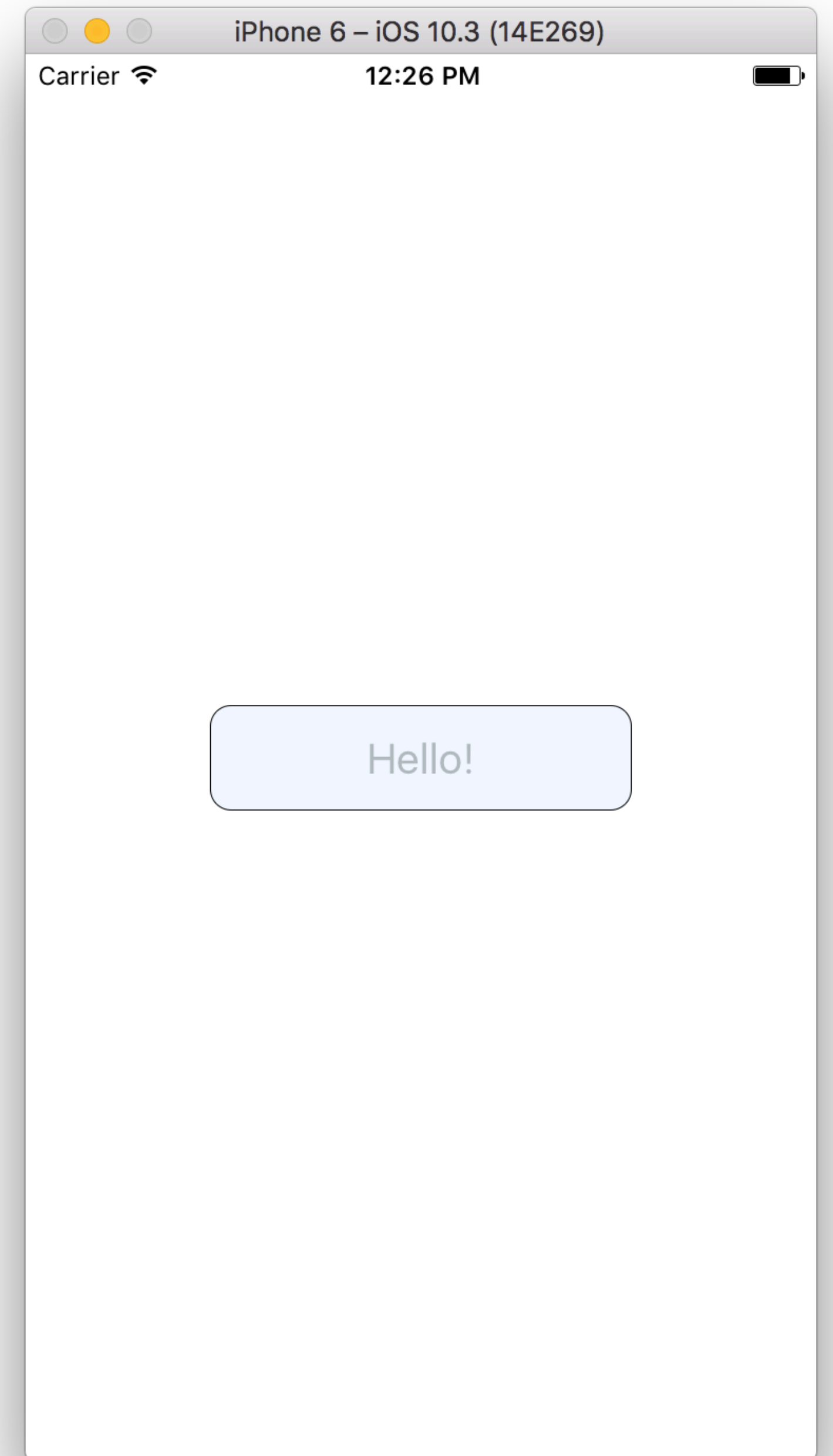
```
class Button extends React.Component {  
  render() {  
    const { background, Title } = styles(this.props);  
    return (  
      <View style={background}>  
        <Title>{this.props.children}</Title>  
      </View>  
    );  
  }  
}
```

# Statuses

```
export default class App extends React.Component {
  render() {
    return (
      <View style={styles.container}>
        <Button size="20" disabled>Hello!</Button>
      </View>
    );
  }
}
```

```
class Button extends React.Component {
  render() {
    const { background, Title } = styles(this.props);
    return (
      <View style={background}>
        <Title>{this.props.children}</Title>
      </View>
    );
  }
}
```

```
const styles = css`
  .background {
    background-color: green;
    rect: 200 50;
    border: 1px;
    content-position: center;
    @media ios {
      background-color: aliceblue;
      border-radius: 10;
    }
  }
  Text.title {
    font-size: ${props => props.size};
    :disabled {
      opacity: 0.5;
    }
  }
`;
```



# Thanks

