

DANIEL GRAHAM PHD

REACT NATIVE HOOKS

HOOKS

- Introduced in 2018 in React 16.8.
- Introduced in 0.59 of React Native March , 2019

<https://facebook.github.io/react-native/blog/2019/03/12/releasing-react-native-059>

LET'S BUILD A TODO APP

- Code taken from <https://medium.com/crowdbotics/build-a-react-native-app-with-react-hooks-5498e1d5fdf6>

USING HOOKS TO SET STATE

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

export default function App() {
  const [count, setCount] = useState(0);

  return (
    <View style={styles.container}>
      <Text>You clicked {count} times.</Text>
      <Button
        onPress={() => setCount(count + 1)}
        title="Click me"
        color="red"
        accessibilityLabel="Click this button to increase count"
      />
    </View>
  );
}
```

You clicked 0 times.

CLICK ME

[HTTPS://SNACK.EXPO.IO/@PROFESSORXII/HOOKSPART1](https://snack.expo.io/@professorxii/hookspart1)

USING HOOKS TO SET STATE

STATE VARIABLE

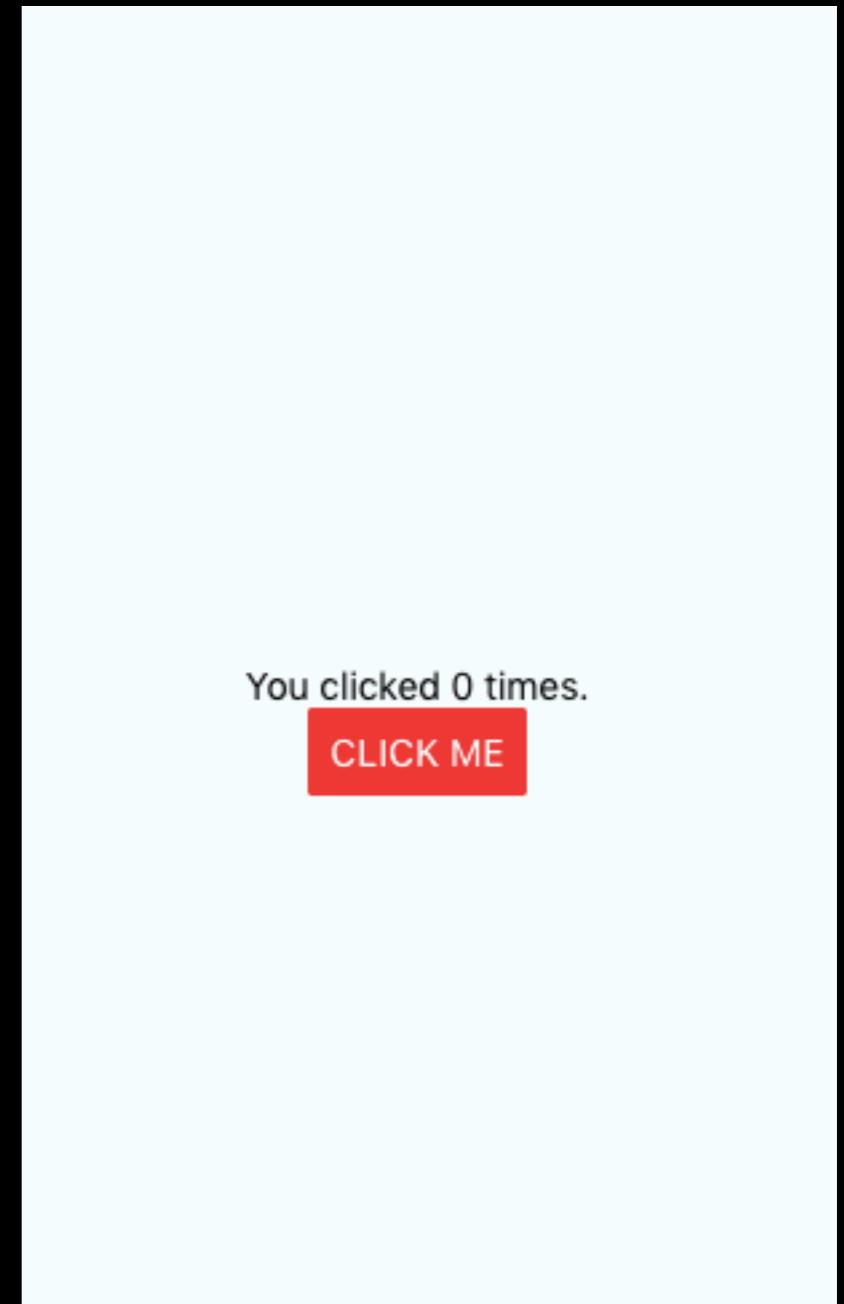
IMPORT

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

export default function App() {
  const [count, setCount] = useState(0);

  return (
    <View style={styles.container}>
      <Text>You clicked {count} times.</Text>
      <Button
        onPress={() => setCount(count + 1)}
        title="Click me"
        color="red"
        accessibilityLabel="Click this button to increase count"
      />
    </View>
  );
}
```

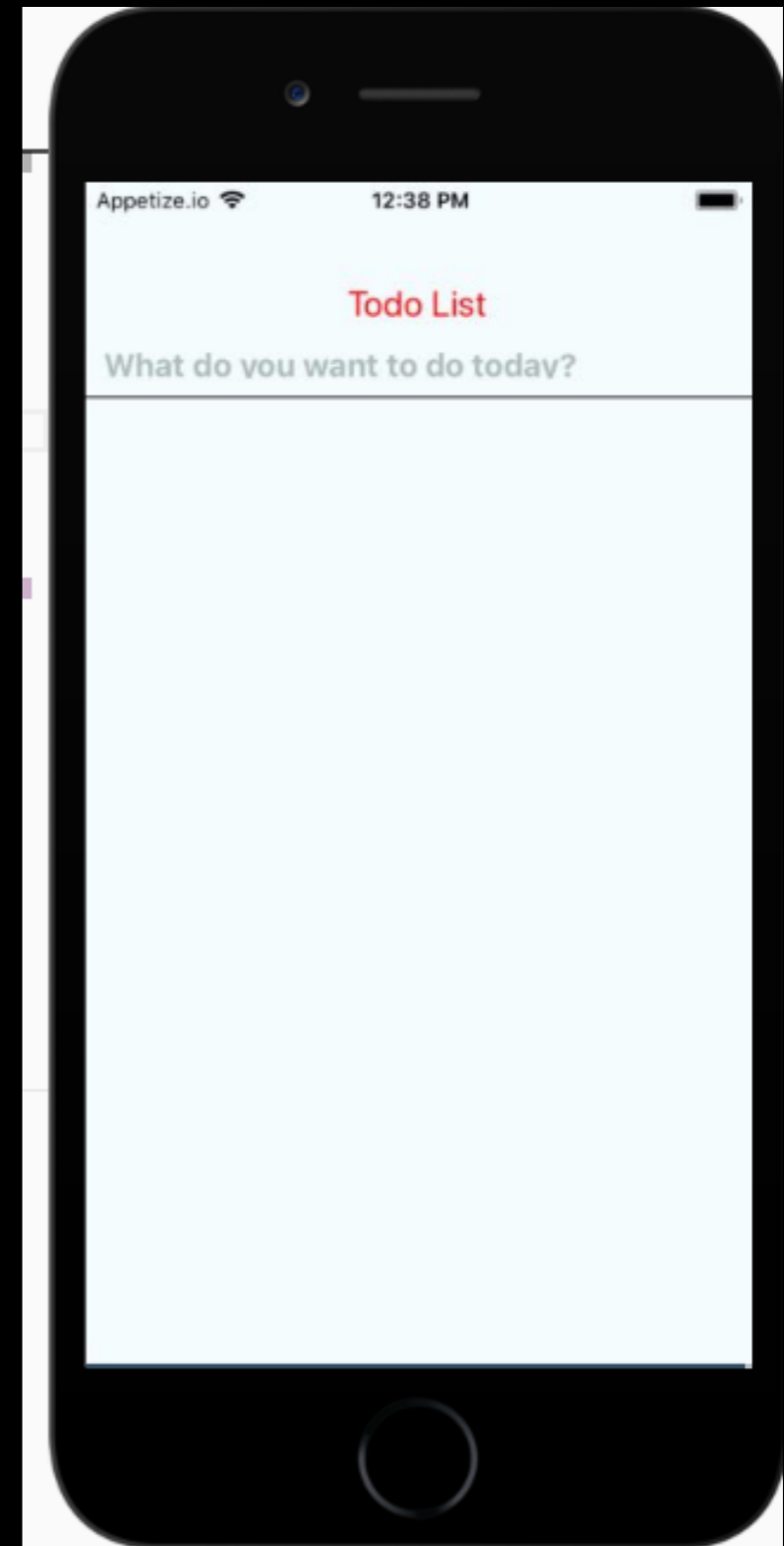
SETTER



[HTTPS://SNACK.EXPO.IO/@PROFESSORXII/HOOKSPARTI](https://snack.expo.io/@professorxii/hooksparti)

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

export default function App() {
  return (
    <View style={styles.container}>
      <Text style={styles.header}>Todo List</Text>
      <View style={styles.textInputContainer}>
        <TextInput
          style={styles.textInput}
          multiline={true}
          placeholder="What do you want to do today?"
          placeholderTextColor="#abbabb"
        />
      </View>
    </View>
  );
}
```

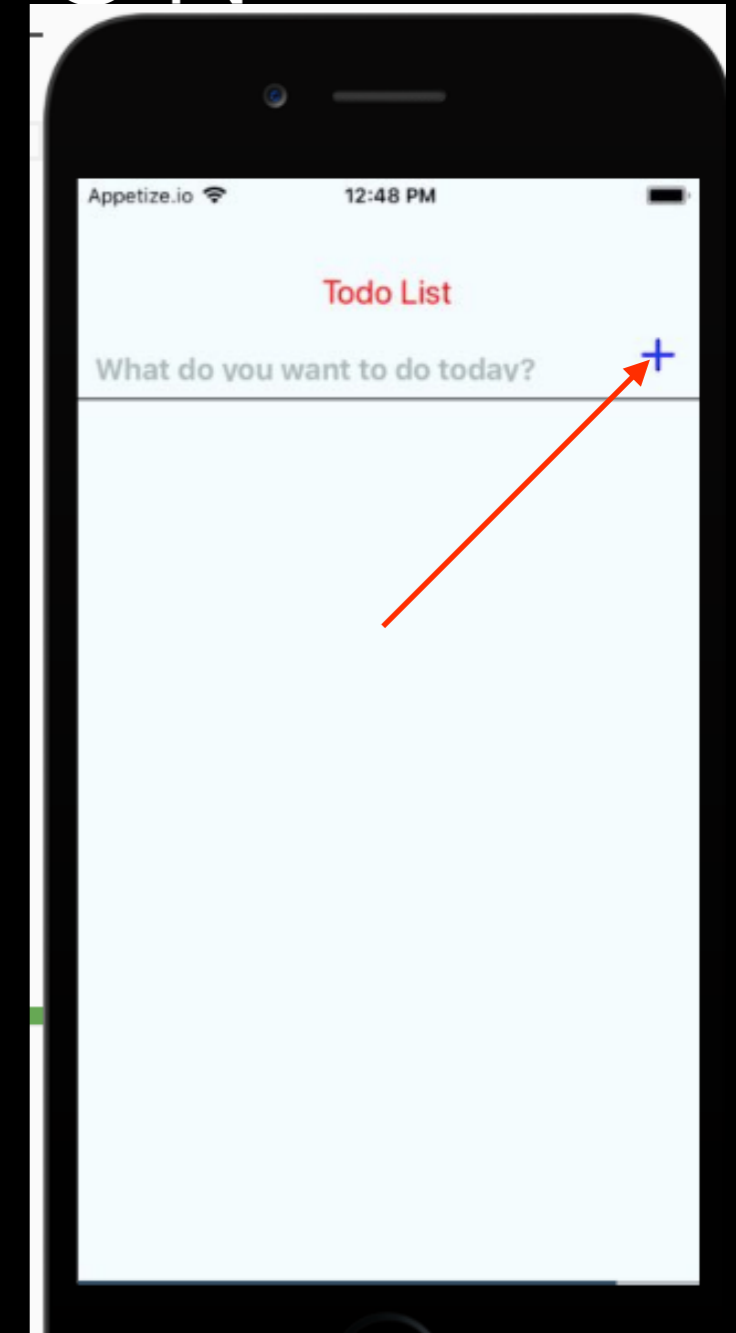


[HTTPS://SNACK.EXPO.IO/@PROFESSORXII/HOOKSPARTII](https://snack.expo.io/@professorxii/hookspartii)

ADDING A COOL + BUTTON

```
import Icon from 'react-native-vector-icons/Feather';
```

```
<View style={styles.textInputContainer}>  
  <TextInput  
    style={styles.textInput}  
    multiline={true}  
    placeholder="What do you want to do today?"  
    placeholderTextColor="#abbabb"  
  />  
  <TouchableOpacity>  
    <Icon name="plus" size={30} color="blue" style={{ marginLeft: 15 }} />  
  </TouchableOpacity>  
</View>
```



[HTTPS://SNACK.EXPO.IO/@PROFESSORXII/HOOKSPARTIII](https://snack.expo.io/@professorxii/hookspartiii)

DEFAULT VALUE

ANOTHER HOOK

```
export default function App() {  
  const [value, setValue] = useState('');  
  const [todos, setTodos] = useState([]);
```

```
  addTodo = () => {  
    if (value.length > 0) {  
      setTodos([...todos, { text: value, key: Date.now(), checked: false }]);  
      setValue('');  
    }  
  };  
}
```

CALLS FUNCTION
WHILE PRESERVING CONTEXT

```
<TouchableOpacity onPress={() => addTodo()}>  
  <Icon name="plus" size={30} color="blue" style={{ marginLeft: 15 }} />  
</TouchableOpacity>
```

[HTTPS://SNACK.EXPO.IO/@PROFESSORXII/HOOKSPARTIV](https://snack.expo.io/@professorxii/hookspartiv)

TODO LIST COMPONENT

- Create a Component that is going hold the todo list

```
export default function TodoList(props) {  
  return (  
    <View style={styles.listContainer}>  
      <Icon name="square" size={30} color="black" style={{ marginLeft: 15 }} />  
      <Text style={styles.listItem}>{props.text}</Text>  
      <Icon  
        name="trash-2"  
        size={30}  
        color="red"  
        style={{ marginLeft: 'auto' }}  
        onPress={props.deleteTodo}  
      />  
    </View>  
  );  
}
```

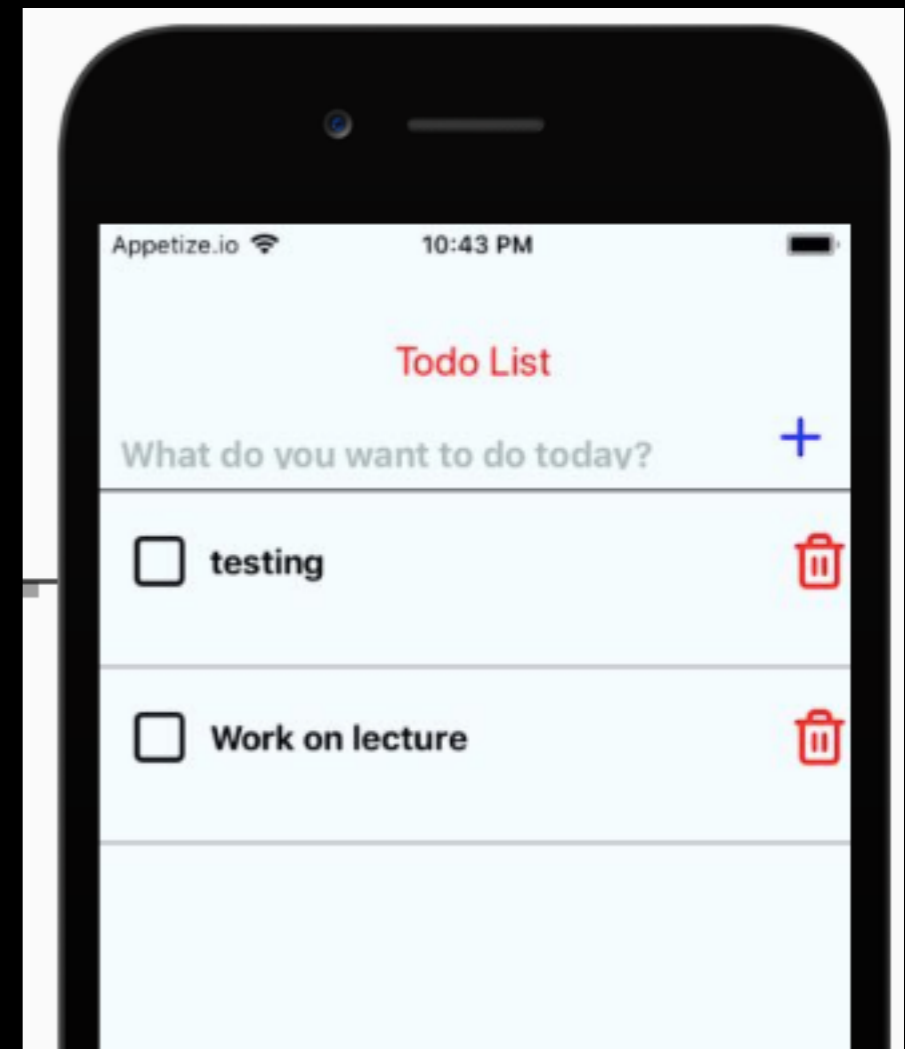


[HTTPS://SNACK.EXPO.IO/@PROFESSORXII/HOOKSPARTV](https://snack.expo.io/@professorxii/hookspartv)

DISPLAYING THE LIST

```
import TodoList from './TodoList';

// ...
return (
  <View style={styles.container}>
    {/* ... */}
    <ScrollView style={{ width: '100%' }}>
      {todos.map(item => (
        <TodoList text={item.text} key={item.key} />
      ))}
    </ScrollView>
  </View>
);
```



[HTTPS://SNACK.EXPO.IO/@PROFESSORXII/HOOKSPARTVI](https://snack.expo.io/@professorxii/hookspartvi)

DELETE AND COMPLETE

```
checkTodo = id => {  
  setTodos(  
    todos.map(todo => {  
      if (todo.key === id) todo.checked = !todo.checked;  
      return todo;  
    })  
  );  
};
```

```
deleteTodo = id => {  
  setTodos(  
    todos.filter(todo => {  
      if (todo.key !== id) return true;  
    })  
  );  
};
```

```

export default function TodoList(props) {
  return (
    <View style={styles.listContainer}>
      <Icon
        name={props.checked ? 'check' : 'square'}
        size={30}
        color="black"
        style={{ marginLeft: 15 }}
        onPress={props.setChecked}
      />
      <View>
        {props.checked && <View style={styles.verticalLine} />}
        <Text style={styles.listItem}>{props.text}</Text>
      </View>
      <Icon
        name="trash-2"
        size={30}
        color="red"
        style={{ marginLeft: 'auto' }}
        onPress={props.deleteTodo}
      />
    </View>
  );
}

```

```

<ScrollView style={{ width: '100%' }}>
  {todos.map(item => (
    <TodoList
      text={item.text}
      key={item.key}
      checked={item.checked}
      setChecked={() => checkTodo(item.key)}
      deleteTodo={() => deleteTodo(item.key)}
    />
  ))}
</ScrollView>

```

- FINAL DEMO

[HTTPS://SNACK.EXPO.IO/@PROFESSORXII/HOOKSPARTVII](https://snack.expo.io/@professorxii/hookspartvii)

LET'S LOOK AT REACT VERSION
OF ANOTHER HOOK
USE EFFECTS

WHY CREATE THE USE EFFECTS HOOK

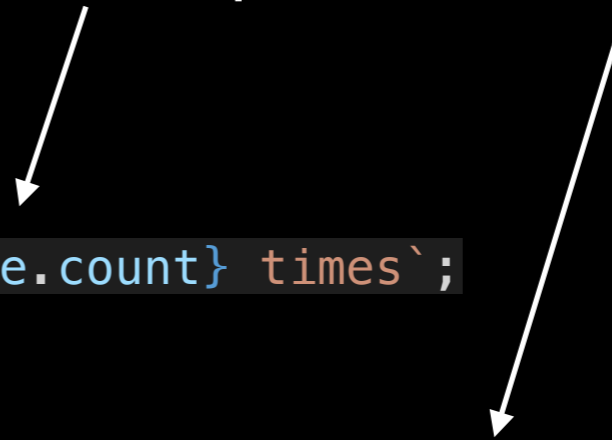
```
class Example extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      count: 0
    };
  }

  componentDidMount() {
    document.title = `You clicked ${this.state.count} times`;
  }

  componentDidUpdate() {
    document.title = `You clicked ${this.state.count} times`;
  }

  render() {
    return (
      <div>
        <p>You clicked {this.state.count} times</p>
        <button onClick={() => this.setState({ count: this.state.count + 1 })}>
          Click me
        </button>
      </div>
    );
  }
}
```

Reduce duplicate Code



you can think of the useEffect Hook as componentDidMount, componentDidUpdate, and componentWillUnmount combined

```
import React, { useState, useEffect } from 'react';
```

```
function Example() {  
  const [count, setCount] = useState(0);
```

```
  // Similar to componentDidMount and componentDidUpdate:  
  useEffect(() => {  
    // Update the document title using the browser API  
    document.title = `You clicked ${count} times`;  
  });
```

```
  return (  
    <div>  
      <p>You clicked {count} times</p>  
      <button onClick={() => setCount(count + 1)}>  
        Click me  
      </button>  
    </div>  
  );  
}
```

Fires Asynchronously

Data fetching, setting up a subscription, and manually changing the DOM in React components are all examples of side effects. - facebook documentation

USE EFFECTS

- Does useEffect run after every render? **Yes!** By default, it runs both after the first render and after every update- facebook React native doc

USE EFFECTS

```
import React, { useState, useEffect } from 'react';
```

```
function Example() {  
  const [count, setCount] = useState(0);
```

```
  // Similar to componentDidMount and componentDidUpdate:  
  useEffect(() => {  
    // Update the document title using the browser API  
    document.title = `You clicked ${count} times`;  
  });
```

```
  return (  
    <div>  
      <p>You clicked {count} times</p>  
      <button onClick={() => setCount(count + 1)}>  
        Click me  
      </button>  
    </div>  
  );  
}
```

YOU CAN ALSO
INCLUDE MULTIPLE
USE EFFECTS

IF USEEFFECTS RETURNS A FUNCTION IT
WILL RUN WHEN THE COMPONENT UNMOUNT

TWO RULES FOR HOOKS

- Don't call Hooks inside loops, conditions, or nested functions
- Only call hooks for react functions: Don't call hooks from regular functions.

RULE OF THUMB KEEP HOOKS AT THE TOP LEVEL.

```
// 🚫 We're breaking the first rule by using a Hook in a condition
if (variable !== '') {
  useEffect(function someFunction() {
    //Some Code
  });
}
```

```
useEffect(function someFunction() {
  // 👍 We're not breaking the first rule anymore
  if (variable !== '') {
    //Some Code
  }
});
```

[HTTPS://REACTJS.ORG/DOCS/HOOKS-RULES.HTML](https://reactjs.org/docs/hooks-rules.html)

LETS REWRITE JOKE APP WITH HOOKS

```
export default function APP() {
  const [joke, setJoke] = useState('Wait on joke');

  useEffect(() => {
    const fetchData = async () => {
      let response = await fetch(
        'http://api.icndb.com/jokes/random?limitTo=[nerdy]&firstName=Prof&lastName=X'
      );
      let parseObject = await response.json();
      setJoke(parseObject.value.joke);
    };
    fetchData();
  });

  return (
    <View style={styles.container}>
      <Text>{joke}</Text>
    </View>
  );
}
```

But does not update

USE EFFECT DOES NOT SUPPORT TOP LEVEL ASYNC

[HTTPS://SNACK.EXPO.IO/@PROFESSORXII/JOKE-SNACK-HOOKS-PART1](https://snack.expo.io/@professorxii/joke-snack-hooks-part1)

BUILD YOUR OWN HOOKS

- Building your own hooks lets you share functionality across components.
- A custom Hook is a JavaScript function whose name starts with "use" and that may call other Hooks

BUILD USE INTERVAL

- READ MORE ABOUT IT HERE: <https://overreacted.io/making-setinterval-declarative-with-react-hooks/>

Custom Hook



```
useInterval(() => {
  const fetchData = async () => {
    let response = await fetch(
      'http://api.icndb.com/jokes/random?limitTo=[nerdy]&firstName=Prof&lastName=X'
    );
    let parseObject = await response.json();
    setJoke(parseObject.value.joke);
  };
  fetchData();
}, 7000);
```

[HTTPS://SNACK.EXPO.IO/@PROFESSORXII/JOKE-SNACK-HOOKS-
PART2](https://snack.expo.io/@professorxii/joke-snack-hooks-part2)


```
useEffect(() => {  
  document.title = `You clicked ${count} times`;  
}, [count]);
```

Dependencies, Will not apply the Effect