

Syntax

Component Story Format (CSF)

Named story export

With CSF format, every named export in the Story file represents a story function by default.

```
// MyComponent.stories.js

import React from 'react';
import MyComponent from './MyComponent';

export default {
  title: 'Path/To/MyComponent',
  component: MyComponent,
};

export const Basic = () => <MyComponent />;
export const WithProp = () => <MyComponent prop="value" />;
```

```
const Template: Story<ButtonProps> = (args) => <Button {...args} />;

// 2. Each named export represents the state of your UI component
export const Primary = Template.bind({});
Primary.args = {
  background: '#ff0',
  label: 'Button',
};

export const Secondary = Template.bind({});
Secondary.args = {
  ...Primary.args,
  label: '👍👎👏👉',
};

export const Tertiary = Template.bind({});
Tertiary.args = {
  ...Primary.args,
  label: '🚗🚚🚛🚝',
};
```

Syntax

Component Story Format (CSF)

The key ingredients are the default export that describes the component, and named exports that describe the stories.

- The default export metadata controls how Storybook lists your stories and provides information used by addons
- Using named exports (such as Default) is how you define your component's stories. These show up underneath the Story and represent different states of the component
- To promote re-use, it's standard practice to create a "template"