

Skills
Network

Component Composition

IBM

What you will learn



Describe the purpose of component composition

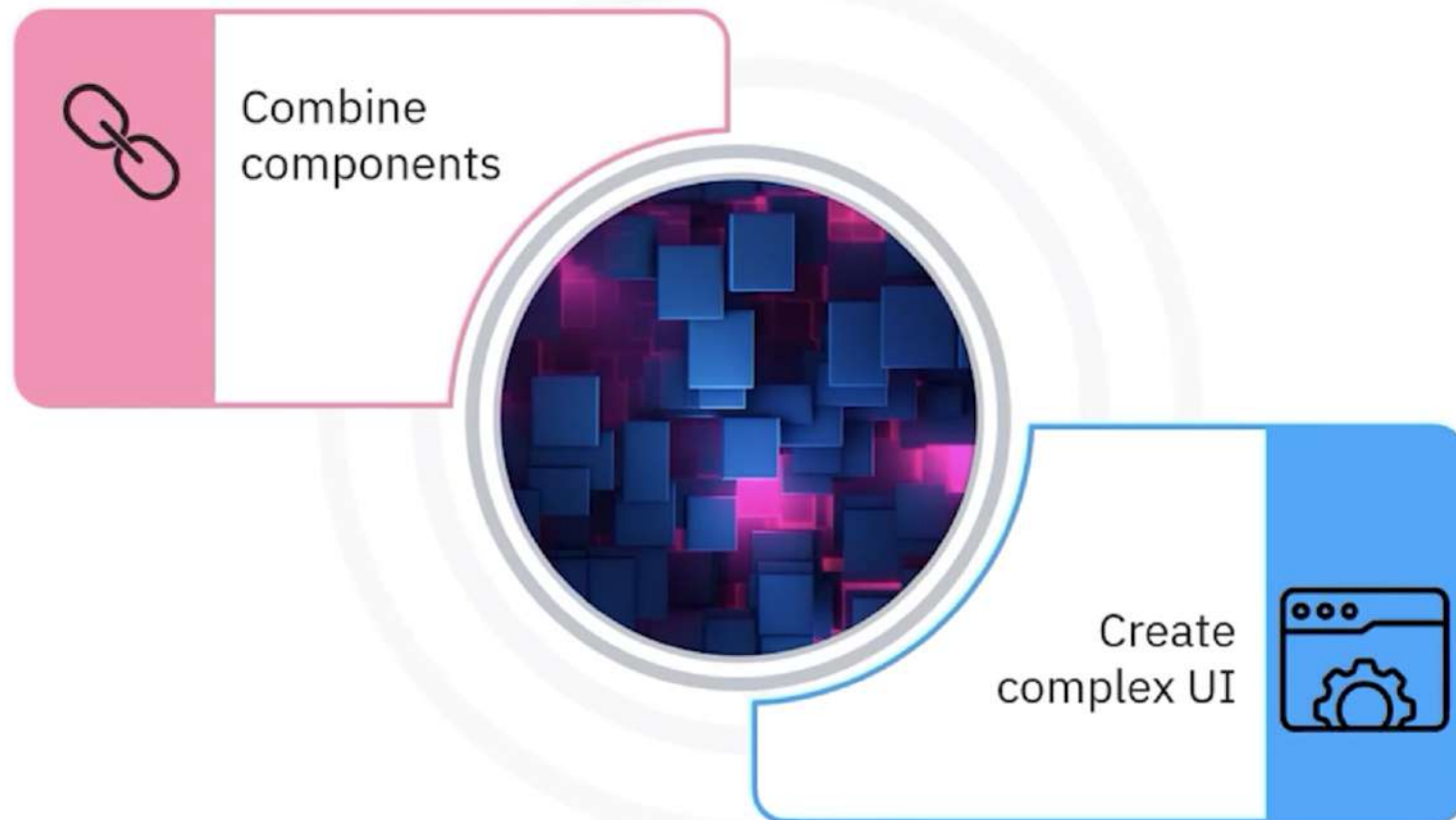


Explain principles of function component composition



Provide an example of component composition

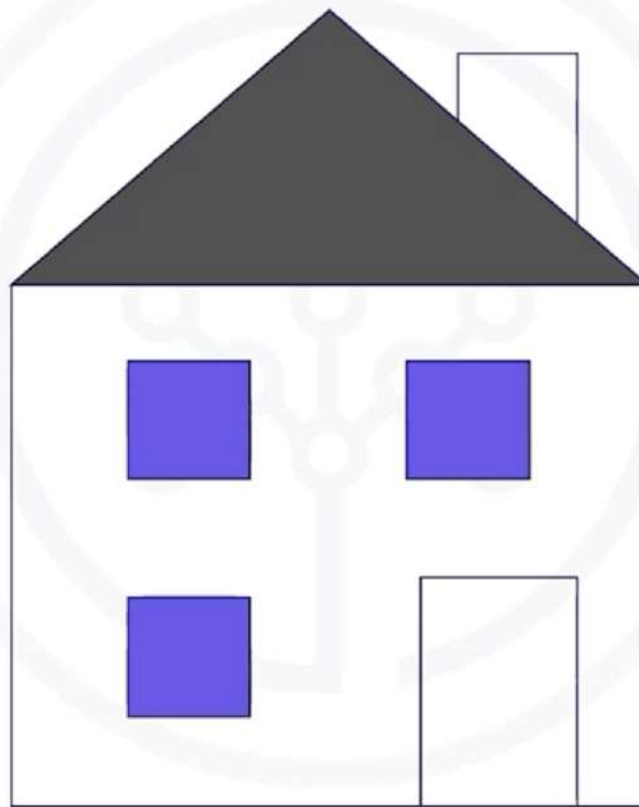
Component composition



Composition principles



Abstraction principle



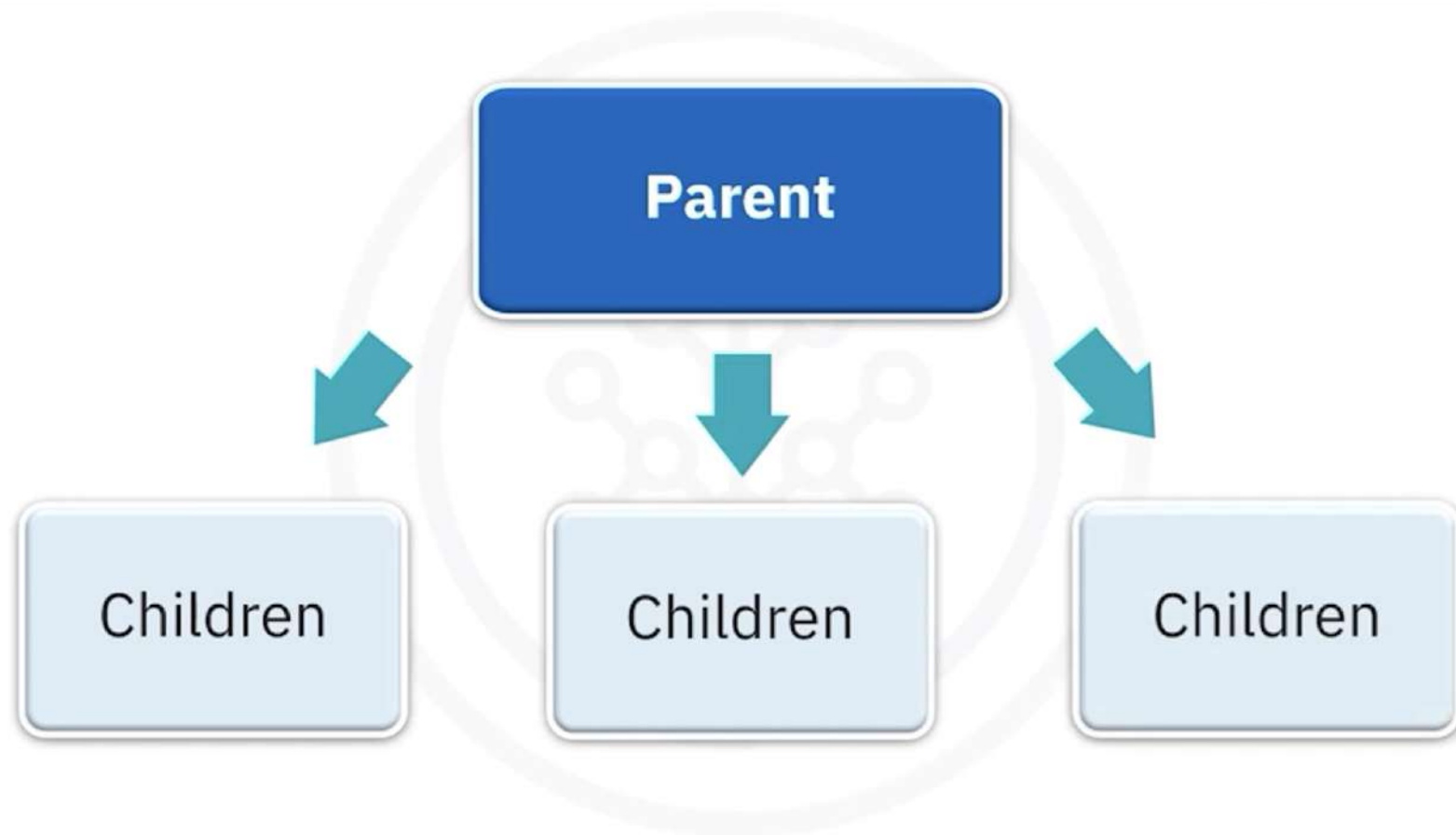
Abstraction principle



Reuse principle



Hierarchy principle



Props and children principle

Data flow:

From parent to child

Parent

Props
data

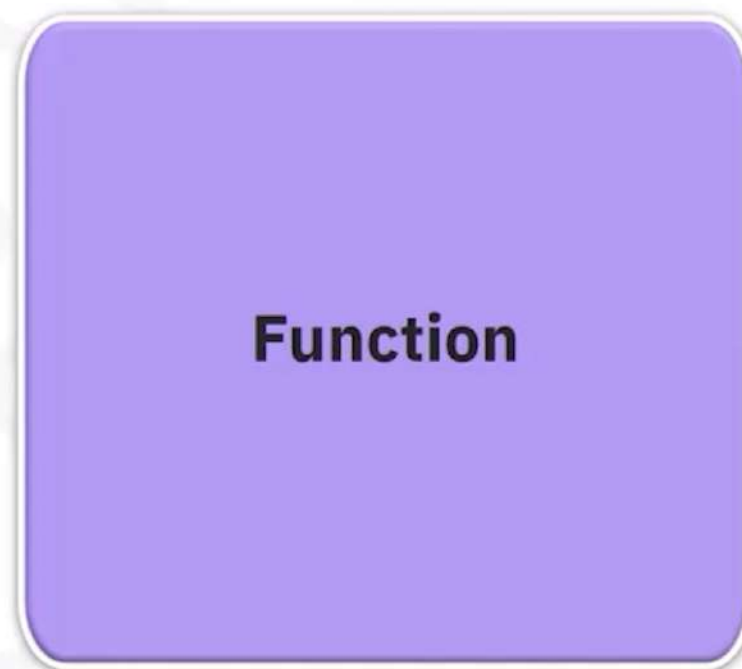
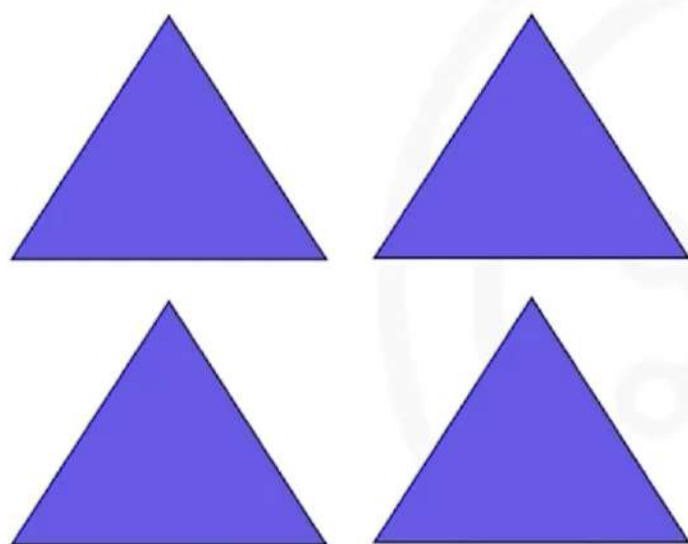
Child

Pass a component into
another one

Parent

Child

Higher-order components (HOCs)



Use case

- Blog for novels
- Features:
 - Title
 - Author
 - Type
 - Description



Component composition example

Title component

```
import React from 'react';  
function Title({ title }) {  
  return <h1>{title}</h1>;  
}  
export default Title;
```

Components for other features

Author component

```
import React from 'react';
function Author({ author }) {
  return <p>Written by: {author}</p>;
}
export default Author;
```

Title component (from previous slide)

```
import React from 'react';
function Title({ title }) {
  return <h1>{title}</h1>;
}
export default Title;
```

Components for other features

Description component

```
import React from 'react';
function Description ({ description
}) {
  return <div>{content}</div>;
}
export default Description;
```

Title component (from previous slide)

```
import React from 'react';
function Title({ title }) {
  return <h1>{title}</h1>;
}
export default Title;
```


Components for other features

Type component

```
import React from 'react';
function Type({ type }) {
  return <div> It is of {type}</div>;
}
export default Type
```

Title component (from previous slide)

```
import React from 'react';
function Title({ title }) {
  return <h1>{title}</h1>;
}
export default Title;
```

HOC example

```
//import required components
function NovelBlog({ title, author, description, type }) {
  return (
    <div className="novel-post">
      <Title title={title} />
      <Author author={author} />
      <Description content={ description } />
      <Type content={ type } />

    </div>
  );
}

export default NovelBlog;
```


HOC example (continued)

```
        .  
        .  
        .  
    return (  
      <div className="app">  
        < NovelBlog  
          title={ novel.title}  
          author={ novel.author}  
          description = { novel.description }  
          type={ novel.type}  
        />  
      </div>  
    );  
  }
```

Main application

```
import React from 'react';
import NovelBlog from './NovelBlog';
function App() {
  const novel = {
    title: 'Harry Potter',
    author: 'J. K. Rowling',
    description: 'A world full of wizards and magic',
    type: 'Fiction'
  };
  return (
    <div className="app">
      <NovelBlog title={novel.title} author={novel.author}
        description={novel.description} type={novel.type}
      />
    </div>
  );
}
export default App;
```

Recap

In this lesson, you learned that:

- You use component composition to combine smaller components to create a complex UI
- Principles of component composition include:
 - Abstraction
 - Reusability
 - Hierarchy
 - Props and children
 - Higher-order components