

```

Hello.propTypes = {
  name: PropTypes.string.isRequired,
  age: PropTypes.number.isRequired,
  isMarried: PropTypes.bool.isRequired,
  friends: PropTypes.array.isRequired,
  address: PropTypes.object.isRequired,
  handleClick: PropTypes.func.isRequired
};

//VS

Hello.propTypes = {
  name: PropTypes.string,
  age: PropTypes.number,
  isMarried: PropTypes.bool,
  friends: PropTypes.array,
  address: PropTypes.object,
  handleClick: PropTypes.func
};

// with isRequired vs Without isRequired vs both, which is best?
explain with example like note

```

## Difference Between isRequired and Without isRequired in PropTypes

### 1. With isRequired

- Adding .isRequired ensures that the prop must be provided when using the component.
- If the required prop is not passed, React will log a warning in the console during development.

Example:

```

javascript
Copy code

Hello.propTypes = {
  name: PropTypes.string.isRequired, // Must provide 'name'
  age: PropTypes.number.isRequired, // Must provide 'age'
  isMarried: PropTypes.bool.isRequired, // Must provide 'isMarried'
  friends: PropTypes.array.isRequired, // Must provide 'friends'
  address: PropTypes.object.isRequired, // Must provide 'address'
  handleClick: PropTypes.func.isRequired // Must provide 'handleClick'
};

```

Usage:

```

jsx
Copy code

<Hello
  name="John"
  age={30}
  isMarried={true}
  friends={['Alice', 'Bob']}
  address={{ city: 'New York', zip: 12345 }}
  handleClick={() => console.log('Clicked!')}
/>

```

- If you forget to pass any prop (e.g., age ), React will throw a warning:

```

python
Copy code

Warning: Failed prop type: The prop 'age' is marked as required in 'Hello', but its va

```

### 2. Without isRequired

- When you do not use isRequired , the prop becomes optional.
- If the prop is not provided, React won't log a warning, and the prop's value will default to undefined unless you provide a default value.

Example:

```

javascript
Copy code

Hello.propTypes = {
  name: PropTypes.string, // Optional
  age: PropTypes.number, // Optional
  isMarried: PropTypes.bool, // Optional
  friends: PropTypes.array, // Optional
  address: PropTypes.object, // Optional
  handleClick: PropTypes.func // Optional
};

```

Usage:

```

jsx
Copy code

<Hello name="John" />

```

- Missing props won't trigger a warning, but accessing props.age will return undefined unless a default value is set.

### 3. Combination: Some Required, Some Optional

- Best practice is to make only critical props required (isRequired ) while leaving others optional based on your component's design.

Example:

```

javascript
Copy code

Hello.propTypes = {
  name: PropTypes.string.isRequired, // Critical
  age: PropTypes.number, // Optional
  isMarried: PropTypes.bool, // Optional
  friends: PropTypes.array.isRequired, // Critical
  address: PropTypes.object, // Optional
  handleClick: PropTypes.func.isRequired // Critical
};

```

Usage:

```

jsx
Copy code

<Hello
  name="John"
  friends={['Alice', 'Bob']}
  handleClick={() => console.log('Clicked!')}
/>

```

- In this case:
  - name , friends , and handleClick are required.
  - Omitting age , isMarried , or address won't log any warnings.

## Which Is Best?

- With isRequired for Critical Props: Use isRequired for props essential to the component's functionality. For example, a button component may require a onClick handler, so it should be marked as required.
- Without isRequired for Optional Props: Leave props that have default values or are not critical optional.
- Combination for Real-World Projects: Best practice is to mix required and optional props based on the design. Only mark props as required if their absence would break the functionality.

## Note for Default Props

When using optional props, it's a good idea to define default values using defaultProps :

```

javascript
Copy code

Hello.defaultProps = {
  age: 25,
  isMarried: false,
  friends: [],
  address: { city: 'Unknown', zip: null },
  handleClick: () => {}
};

```

This ensures that even if the optional props are not passed, your component has sensible defaults.