

```
n = 10
sum_all = 0

for i in range(1, n + 1):
    sum_all = sum_all + i

print(sum_all)  # 55
```

```
n = 10
sum_all = sum(range(1, n + 1))

print(sum_all)  # 55
```

```
def create_user(user_ids):
    for id in user_ids:
        make_xyz_api_request(id)
        time.sleep(2)
```

```
def create_user(user_ids):
    for id in user_ids:
        make_xyz_api_request(id)
        time.sleep(2) # NOTE: serv
```

```
// Dirty
import Title from './Title';
export const Thingie = ({ description }) => (
  <div class="thingie">
    <div class="description-wrapper">
      <Description value={description} />
    </div>
  </div>
);
export const ThingieWithTitle = ({ title, description }) => (
  <div>
    <Title value={title} />
    <div class="description-wrapper">
      <Description value={description} />
    </div>
  </div>
);
```

```
// Clean
import Title from './Title';
export const Thingie = ({ description, children }) => (
  <div class="thingie">
    {children}
    <div class="description-wrapper">
      <Description value={description} />
    </div>
  </div>
);
export const ThingieWithTitle = ({ title, ...others }) => (
  <Thingie {...others}>
    <Title value={title} />
  </Thingie>
);
```

```
// Dirty
const Icon = ({ className, onClick }) => {
  const additionalClasses = className || 'icon-large';
  return (
    <span
      className={`icon-hover ${additionalClasses}`}
      onClick={onClick}>
    </span>
  );
};
```

```
// Clean
const Icon = ({ className = 'icon-large', onClick }) => (
  <span className={`icon-hover ${className}`} onClick={onClick} />
);
```

```
// Dirty
class User extends Component {
  state = { loading: true };

  render() {
    const { loading, user } = this.state;
    return loading
      ? <div>Loading...</div>
      : <div>
        <div>
          First name: {user.firstName}
        </div>
        <div>
          First name: {user.lastName}
        </div>
        ...
      </div>;
  }

  componentDidMount() {
    fetchUser(this.props.id)
      .then((user) => { this.setState({ loading: false, user }) })
  }
}
```

```
// Clean
import RenderUser from './RenderUser';
class User extends Component {
  state = { loading: true };

  render() {
    const { loading, user } = this.state;
    return loading ? <Loading /> : <RenderUser user={user} />;
  }

  componentDidMount() {
    fetchUser(this.props.id)
      .then(user => { this.setState({ loading: false, user })})
  }
}
```