

# GUIDE TO DX

# conditional render

IN REACT 18.2

# CLEAN | DX

## conditional render

In React we can conditionally render components

This is different from passing a prop such as *isOpen*.

binary operator

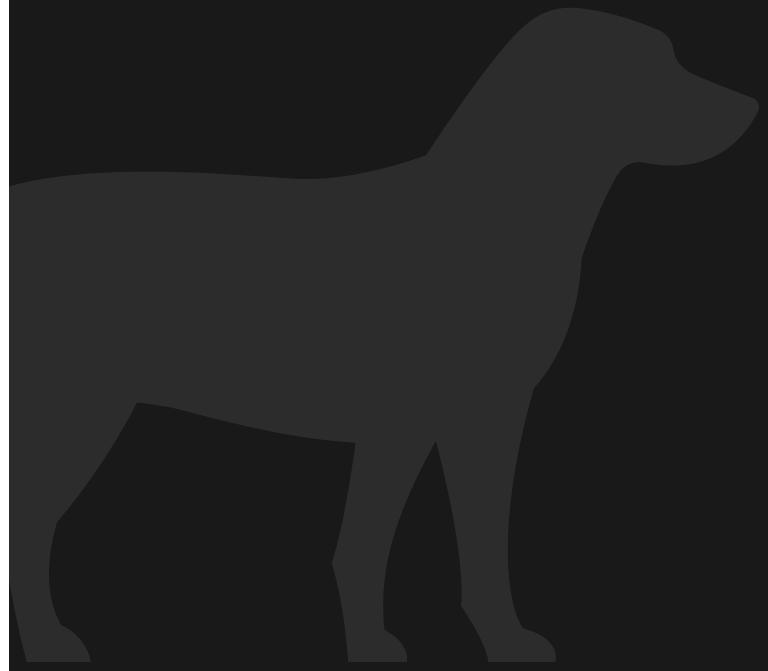
selective rendering

# CLEAN | DX

## conditional render

To conditionally render a component means that we only render it when we need it

This also means that when we do render it, all lifecycle hooks are triggered anew.



reset state  
clean slate

more predictable  
faster  
leaner

# CLEAN | DX

## conditional render

```
const [isOpen, setIsOpen] = useState(false);  
return <Component isOpen={isOpen} />
```



not good

# CLEAN | DX

## conditional render

Passing an *isOpen* equivalent prop triggers the lifecycle hooks of a component

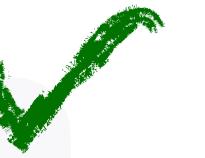
If *isOpen* is *false*, then we just wasted resources.

unnecessary  
unpredictable  
slow

# CLEAN | DX

## conditional render

```
const [isOpen, setIsOpen] = useState(false);  
return isOpen && <Component />
```



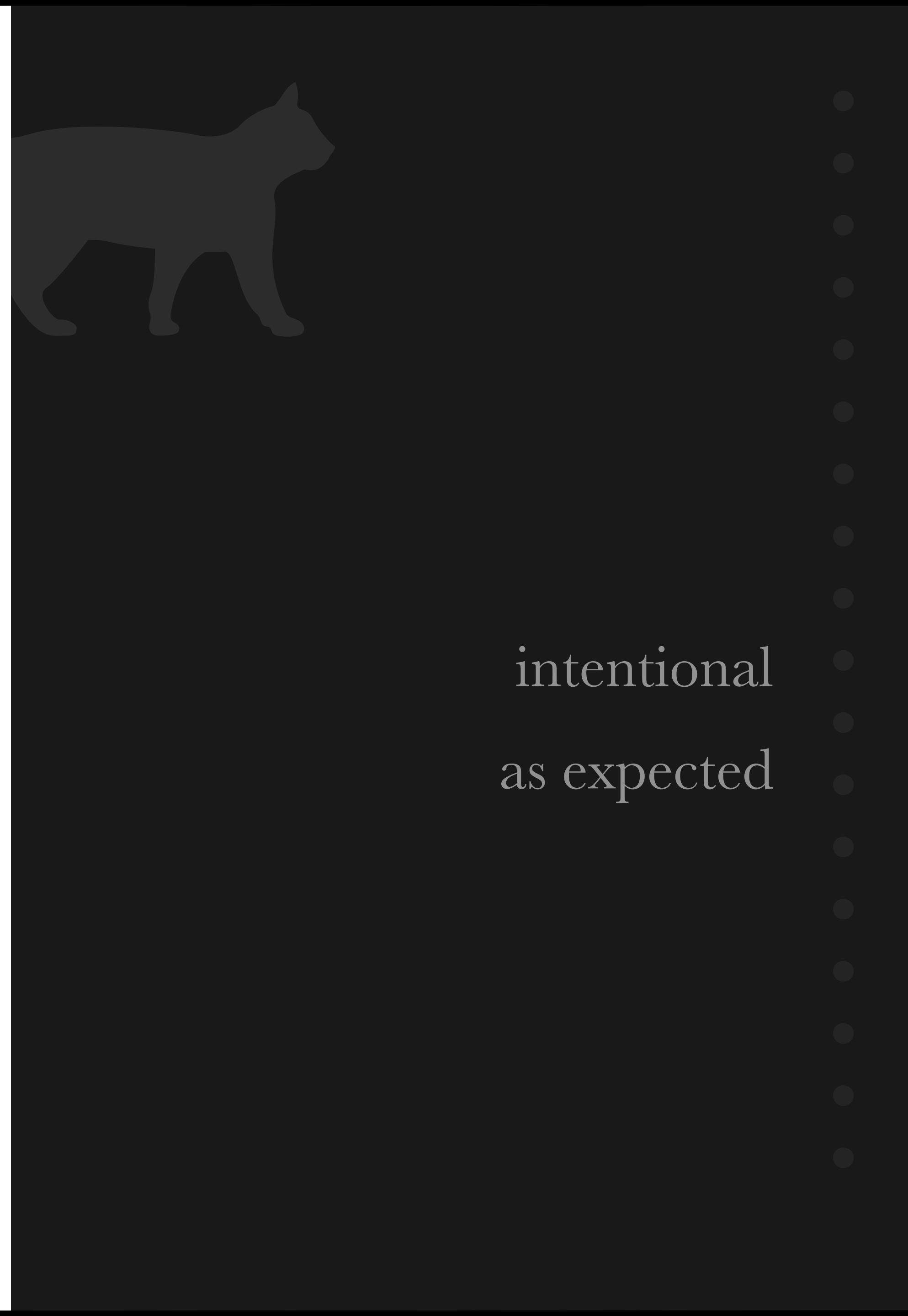
good

# CLEAN | DX

## conditional render

Placing the component behind a condition ensures the component will only execute when condition is met

This is usually what you want.

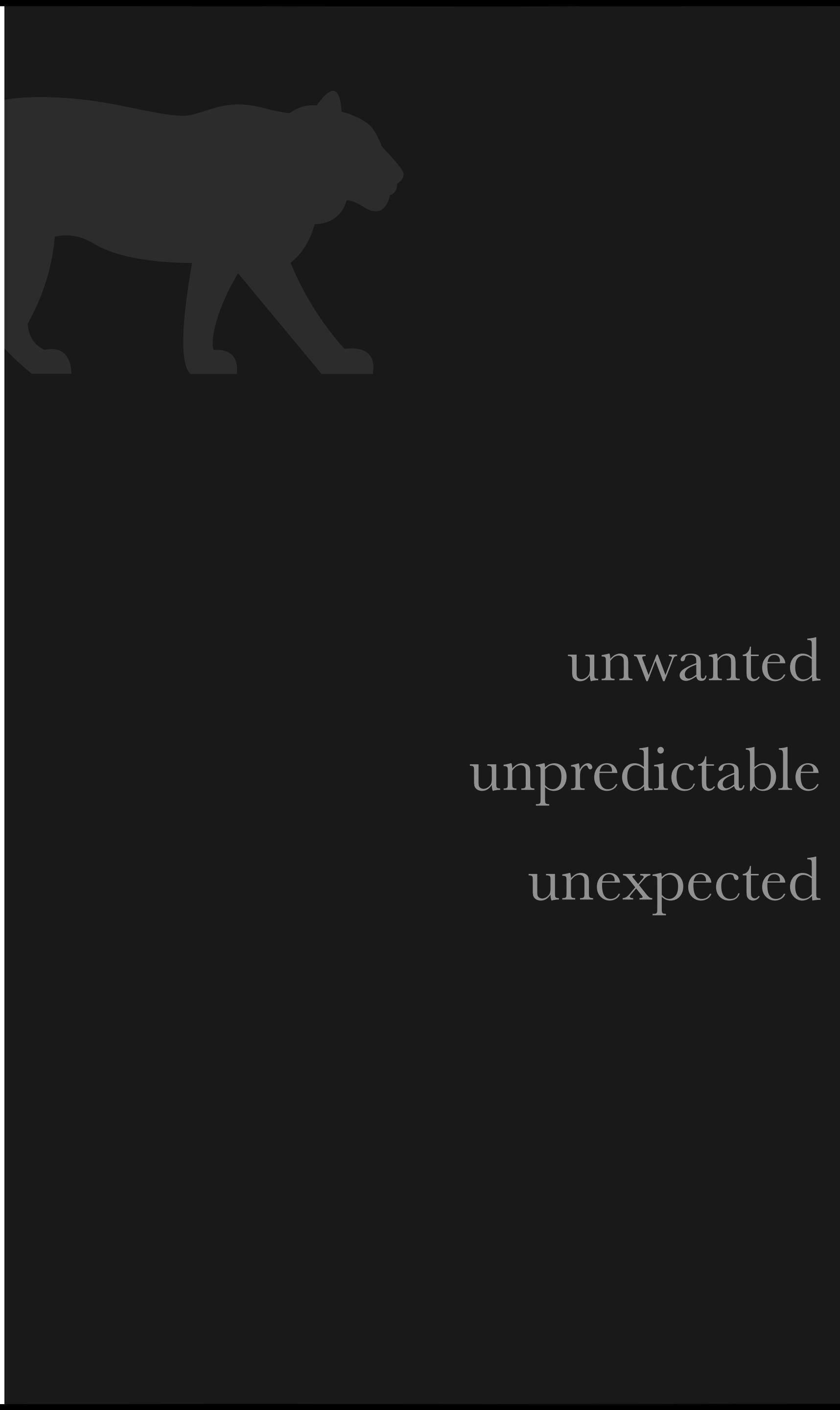


# CLEAN | DX

## conditional render

The conditional should resolve to a boolean.

Any other *falsey* value will render instead of the component.



unwanted

unpredictable

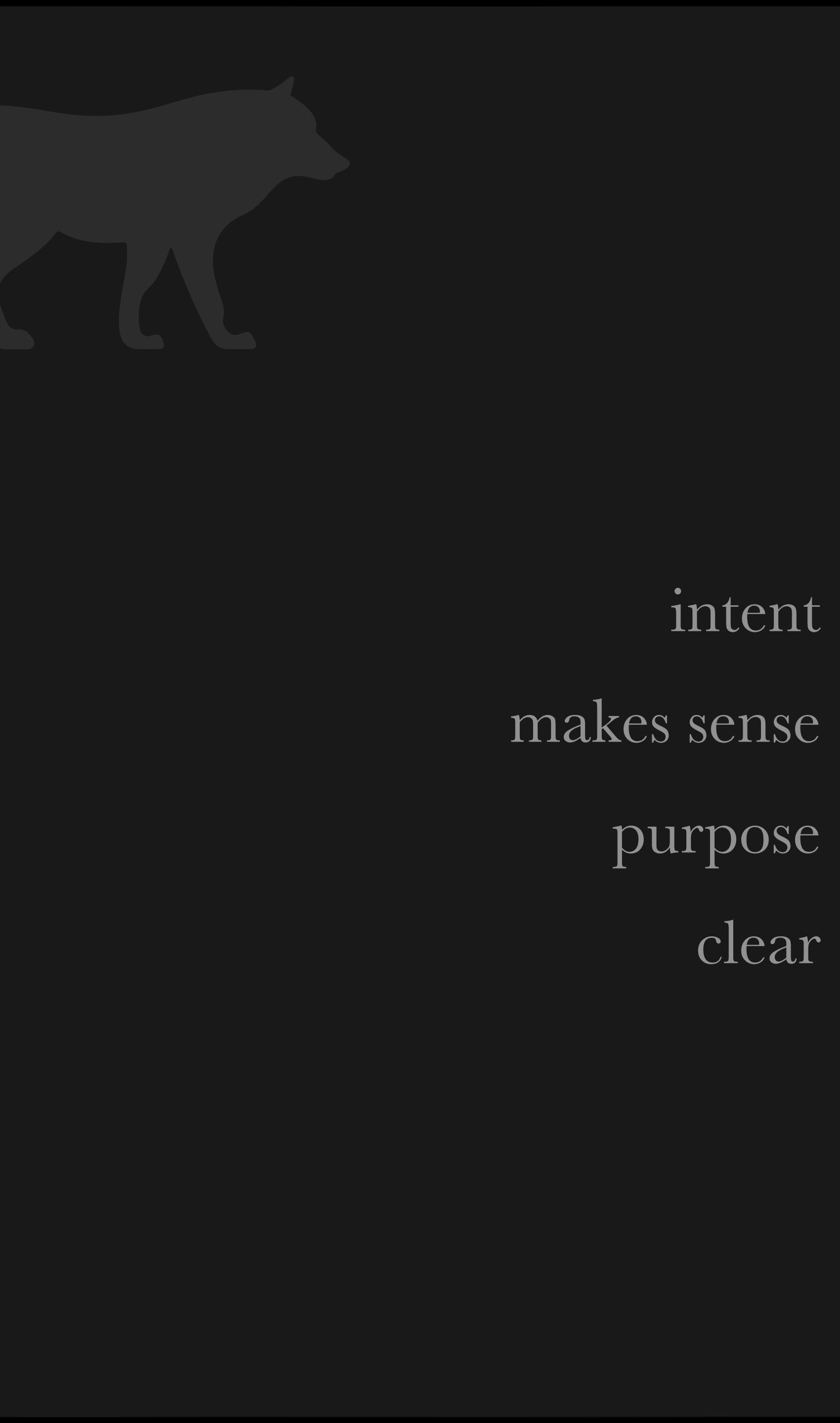
unexpected

# CLEAN | DX

## conditional render

The conditional identifier should be semantically correct and communicate purpose

Try to keep identifier generic when possible and more specific when necessary.



intent

makes sense

purpose

clear

# CLEAN | DX

## conditional render

isOpen

isModalOpen

isVisible



isNotOpen

show

i



positive

communicate intent

hint at boolean

# CLEAN | DX

## conditional render

We want to keep our identifiers on the positive side so that we can avoid double negative logic

It is easier to understand `isOpen` than `!isNotOpen`, so the cognitive complexity is lower.



intent

makes sense

purpose

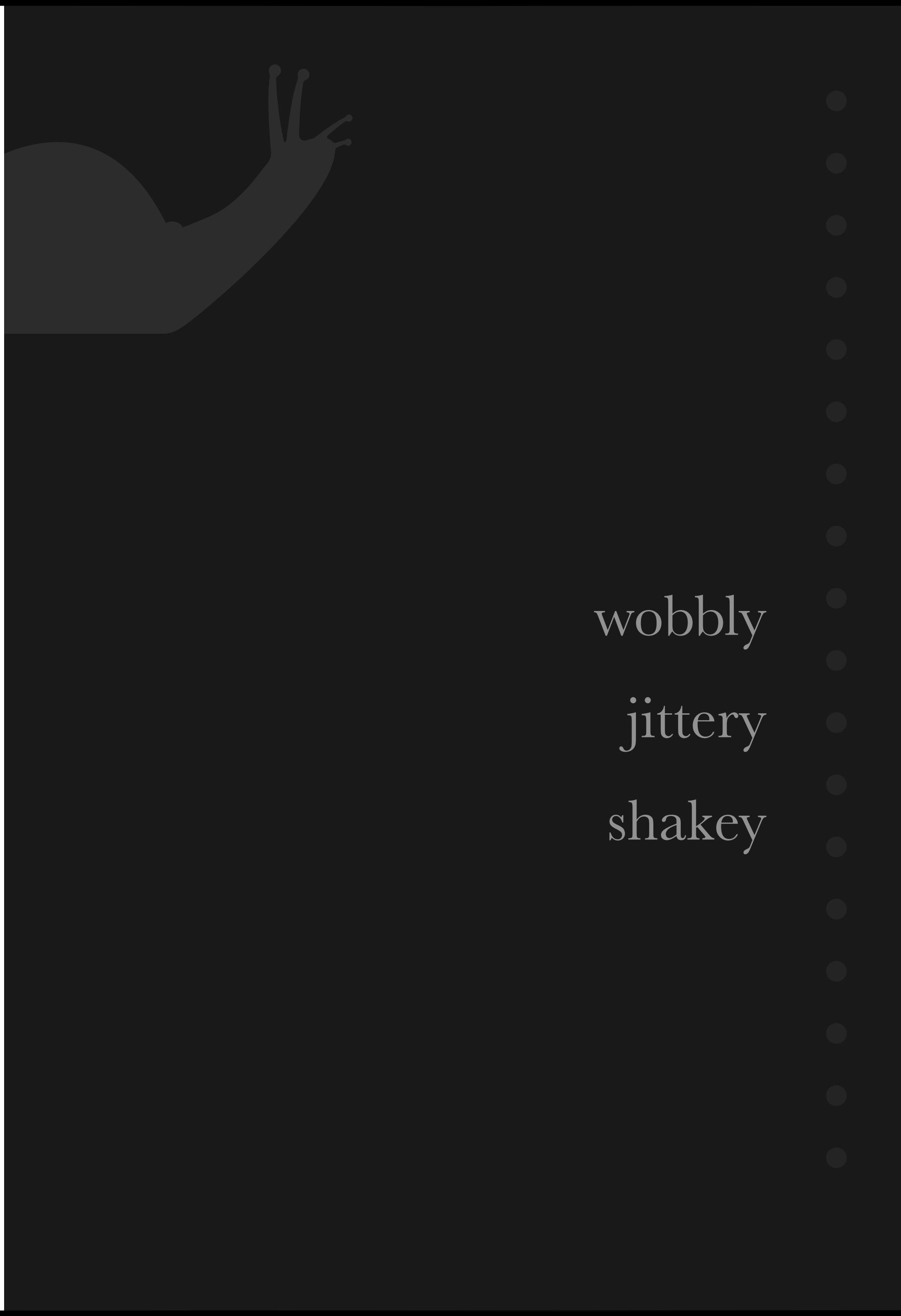
clear

# CLEAN | DX

## conditional render

There can be slow visual removals of components when using the *isOpen* approach rather than conditional rendering.

Use conditional rendering in JSX.



# CLEAN | DX

## conditional render

Milan has over 27 years of professional experience

Presently he focuses on front-end technologies, especially React

Contact him for engagements.



Author: Milan Adamovsky

Copyright © 2022

[milan.adamovsky@gmail.com](mailto:milan.adamovsky@gmail.com)



# CLEAN | DX

## conditional render

Website coming soon...

[www.cleandx.com](http://www.cleandx.com)

