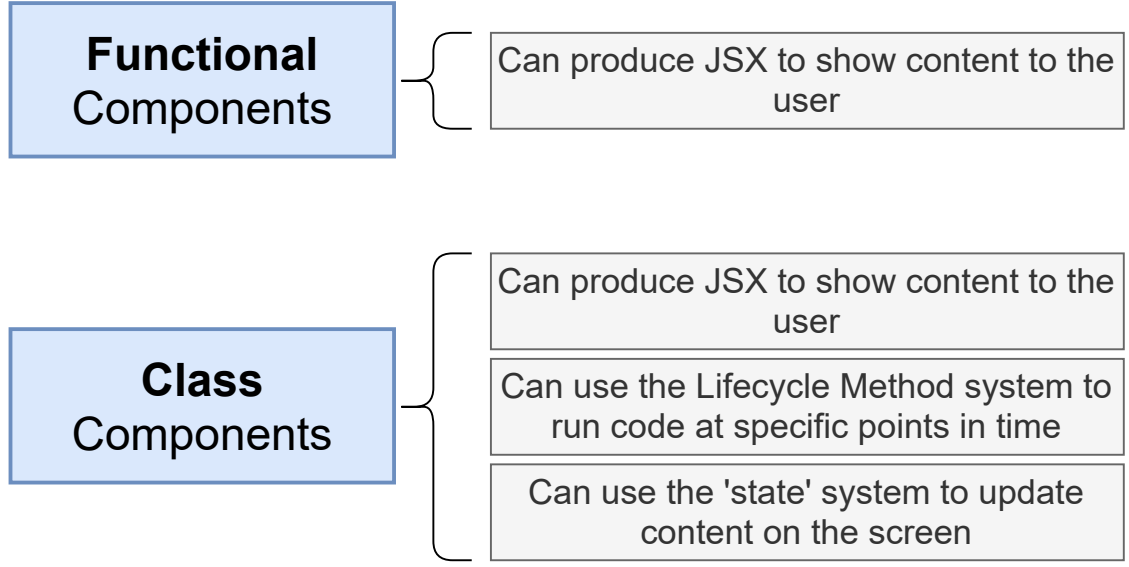
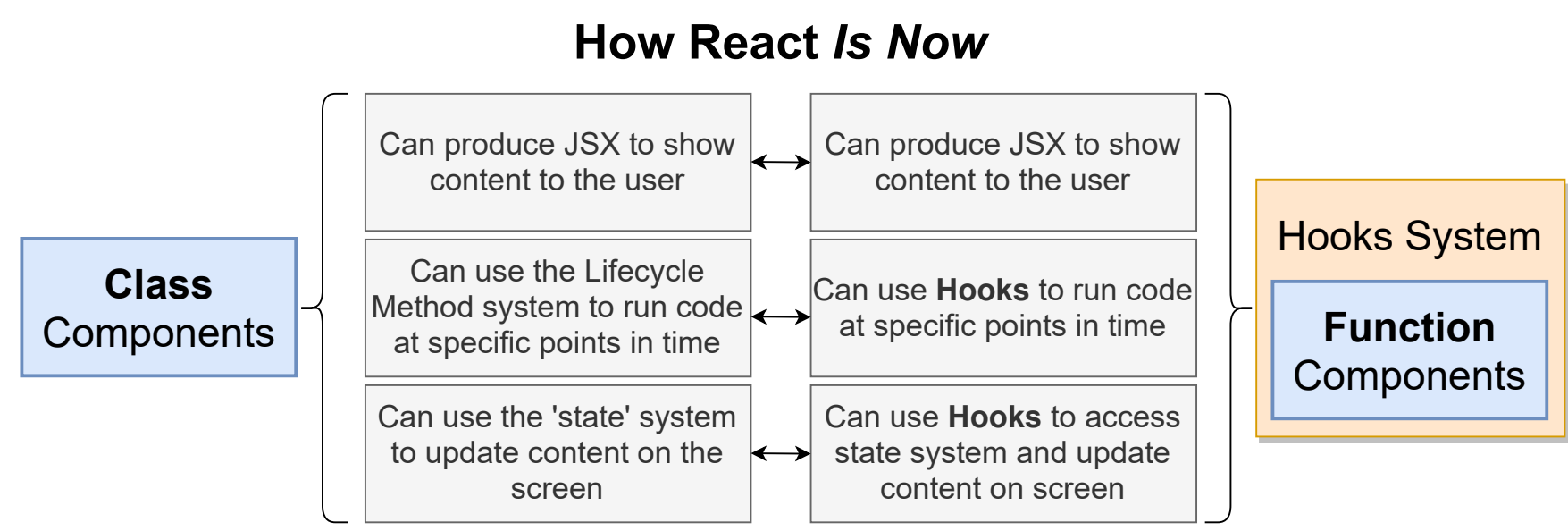
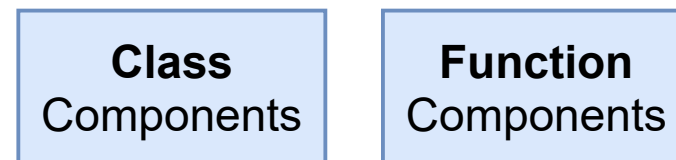
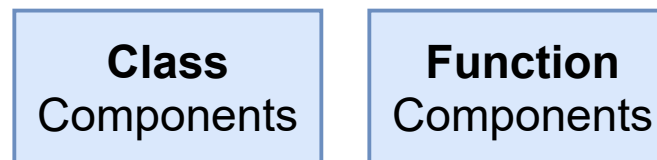


How React *Used to Be*



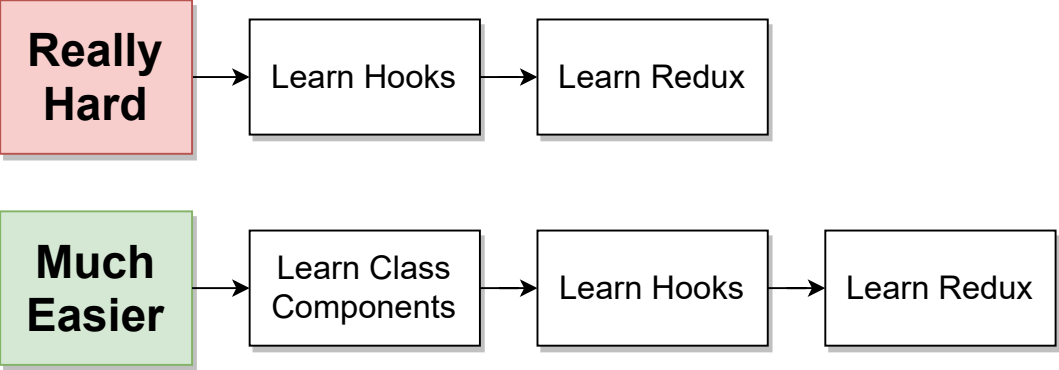






Companies with ***established projects*** are using Class-based components

Companies with ***newer projects*** may be using Class-based ***or*** Function-based components





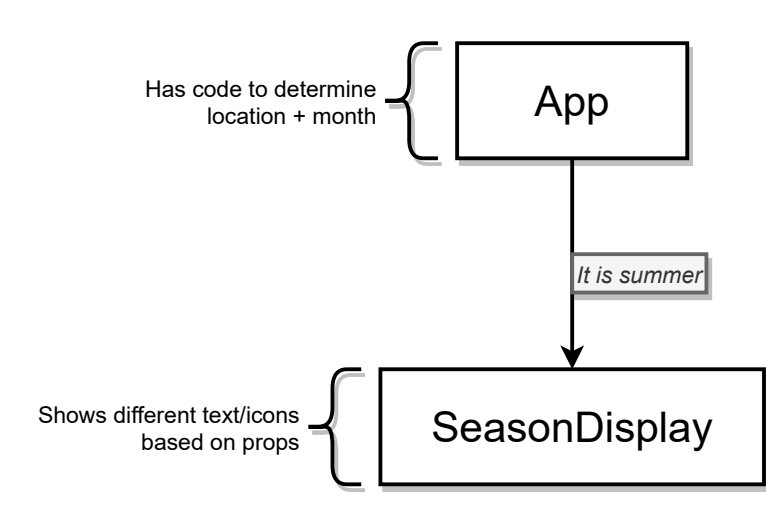
If the user is:	in the northern hemisphere	and	between October and March
	in the southern hemisphere	and	between March and October



If the user is:	in the northern hemisphere	and	between March and October
	in the southern hemisphere	and	between October and March

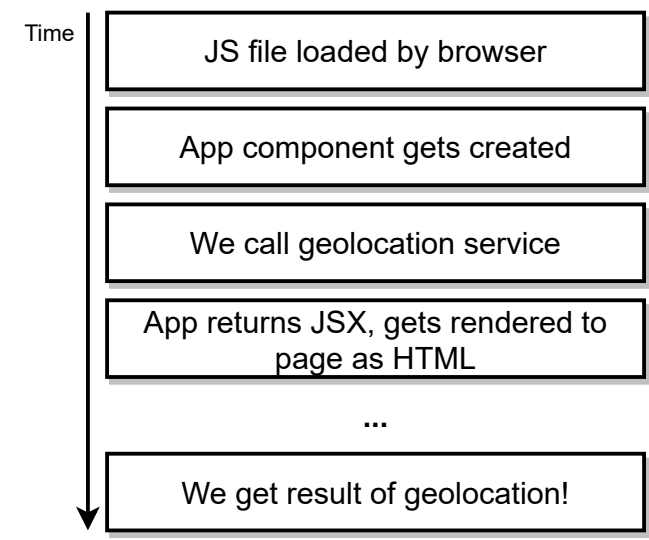
App Challenges

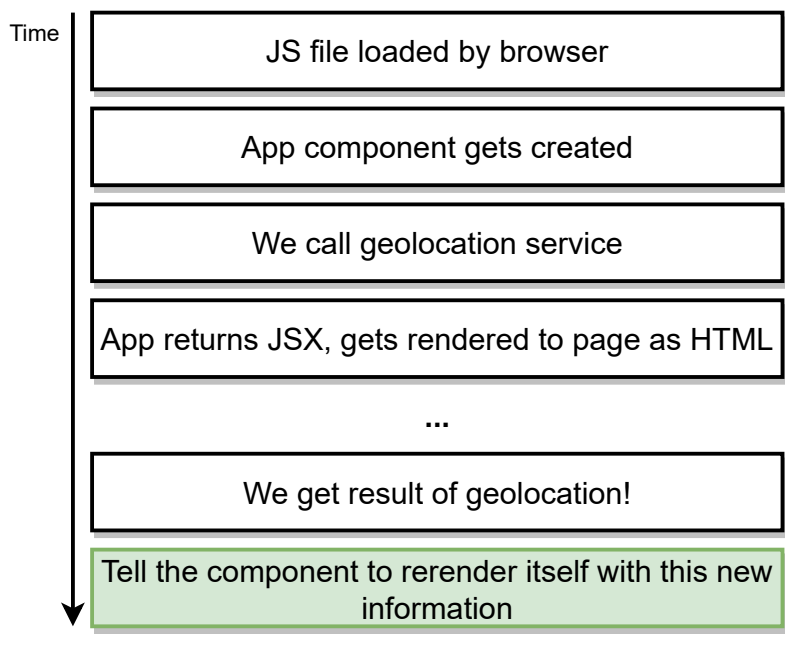
Need to get the users physical location
Need to determine the current month
Need to change text and styling based on location + month

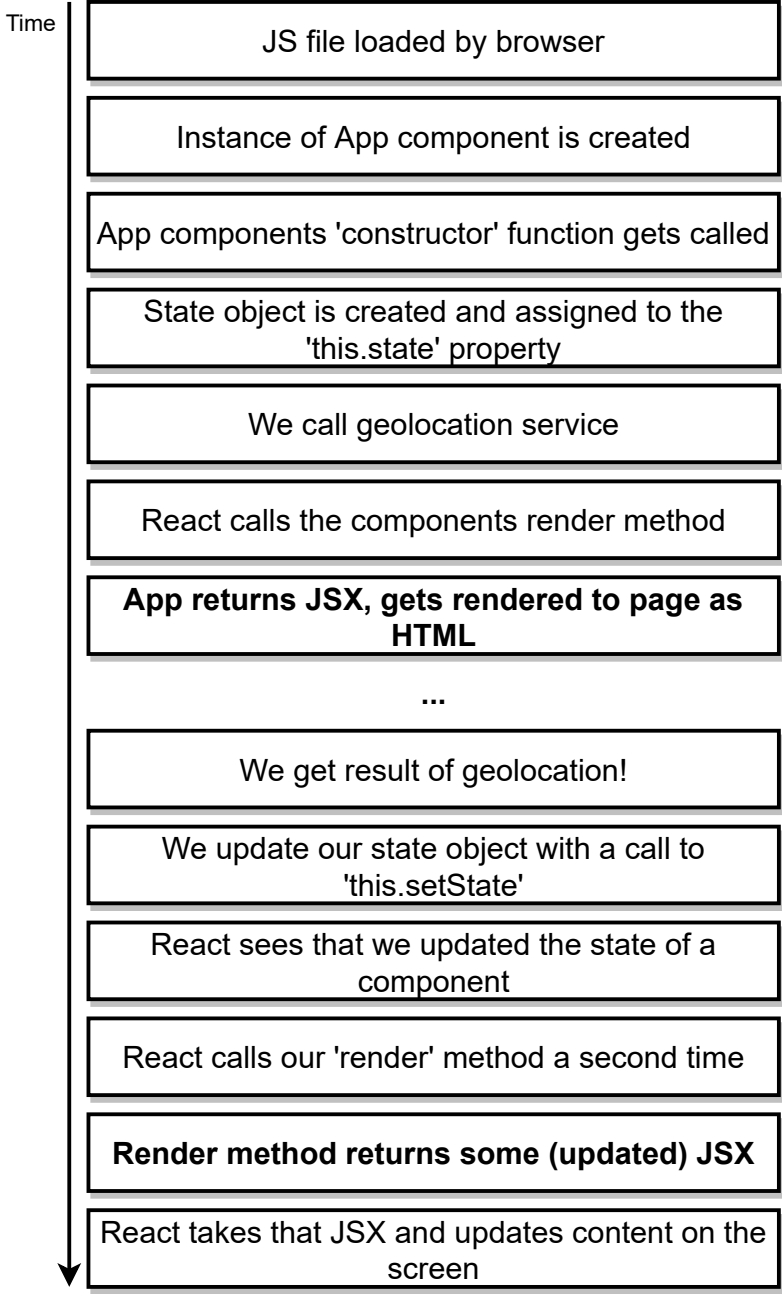


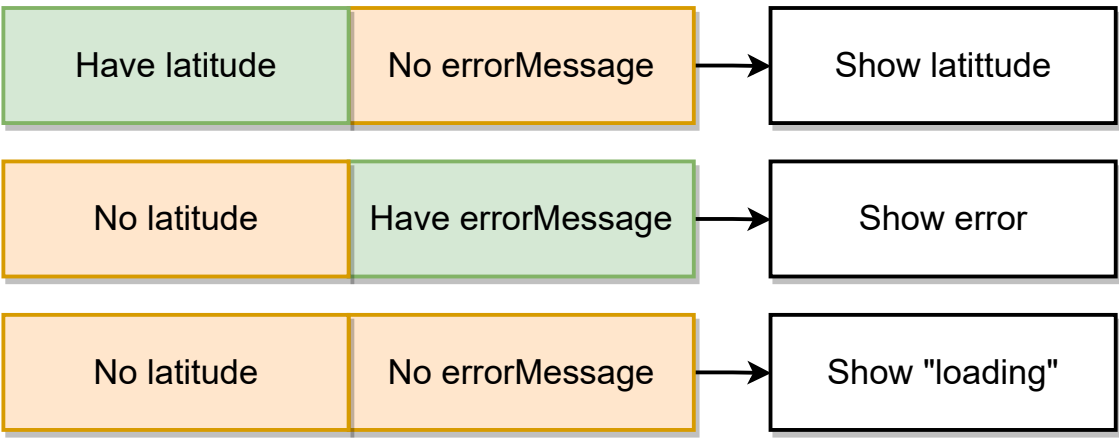
Geolocation API

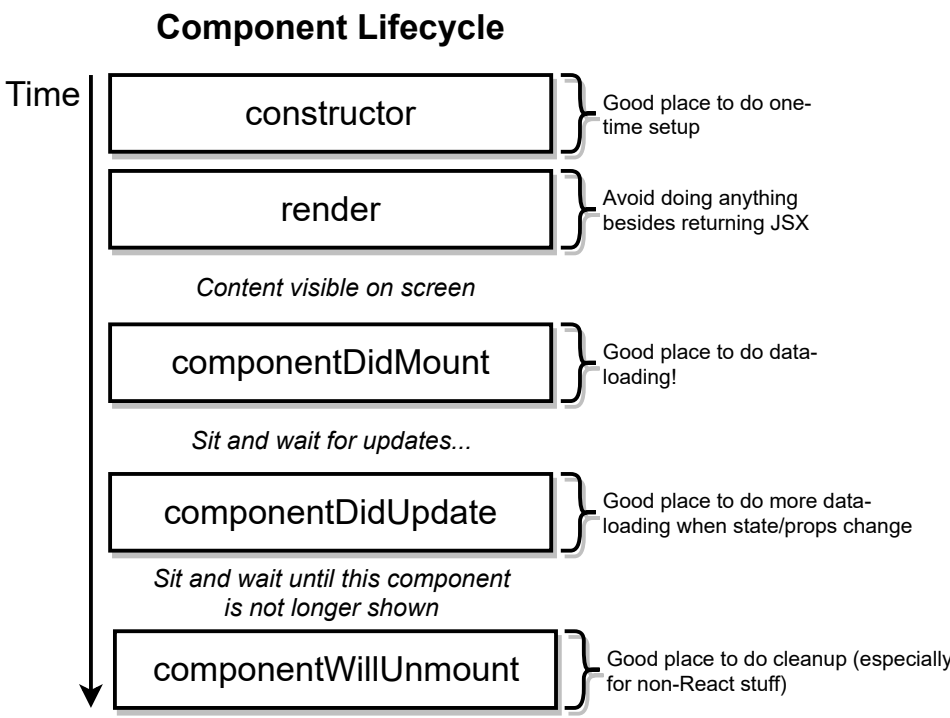
developer.mozilla.org/en-US/docs/Web/API/Geolocation_API

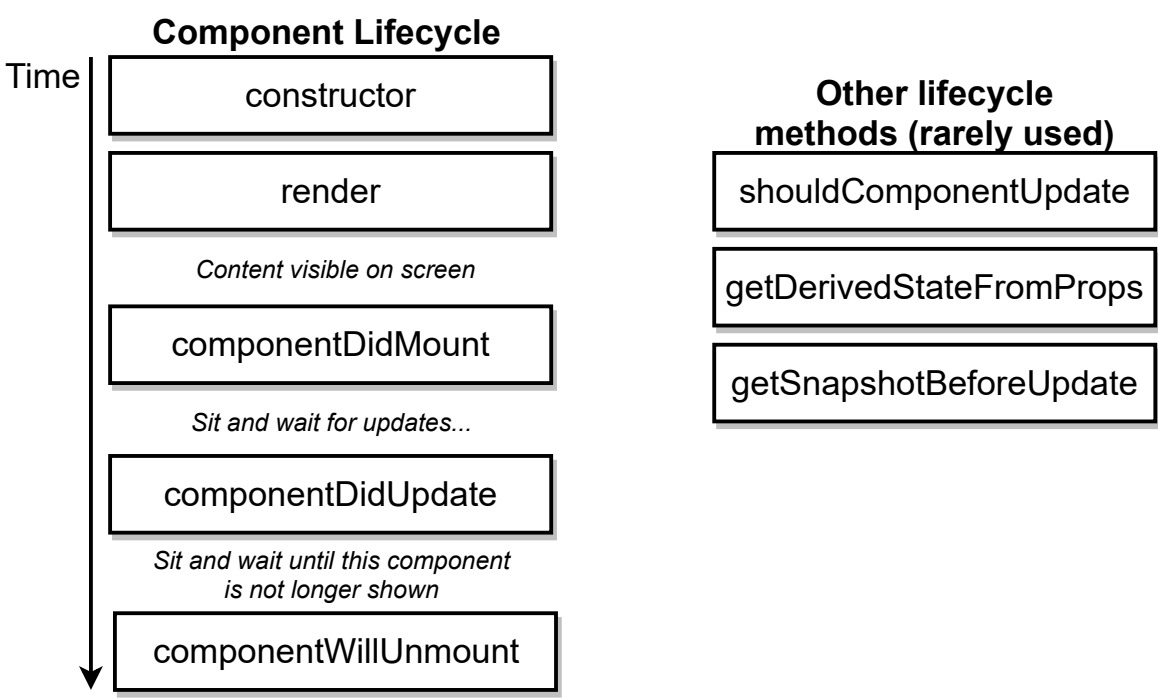










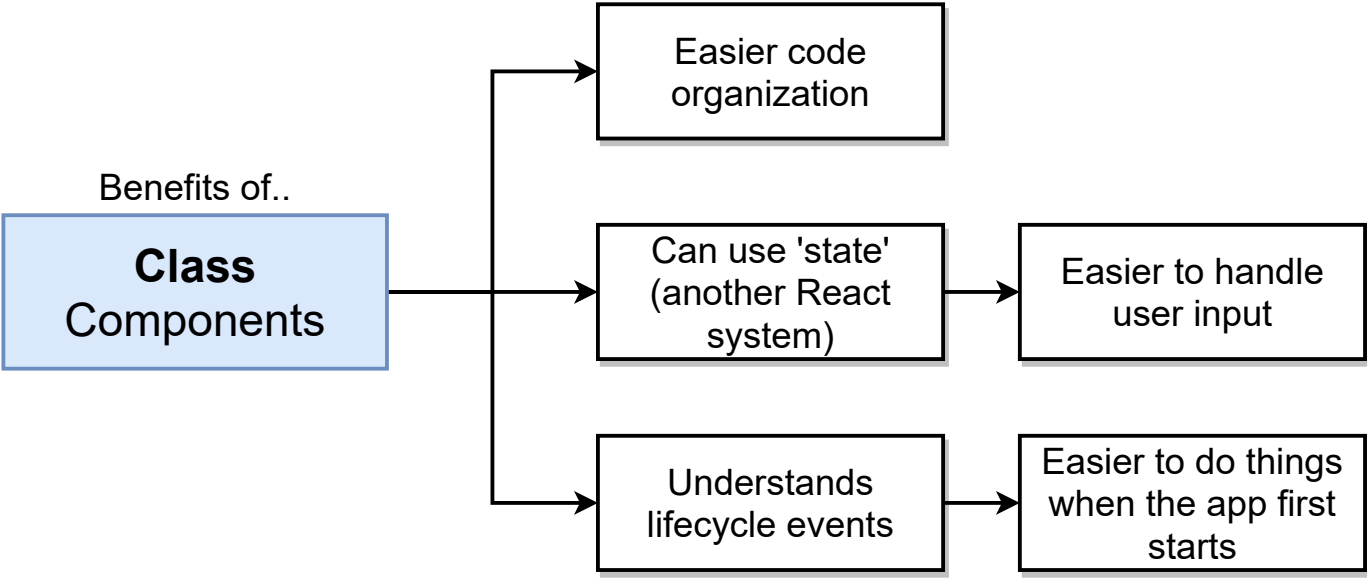


Northern Hemisphere

0	1	2	3	4	5	6	7	8	9	10	11
Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sept	Oct	Nov	Dec

Southern Hemisphere

0	1	2	3	4	5	6	7	8	9	10	11
Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sept	Oct	Nov	Dec



**Rules of Class
Components**

Must be a Javascript Class
Must <i>extend</i> (subclass) React.Component
Must define a 'render' method that returns some amount of JSX

Rules of State

Only usable with class components
You will confuse props with state :(
'State' is a JS object that contains data relevant to a component
Updating 'state' on a component causes the component to (almost) instantly rerender
State must be initialized when a component is created
State can only be updated using the function 'setState'

Technically can be used with functional components using the 'useState' system (going to discuss later)

