Hola Ariel! Mira, se puede hacer de muchas formas. Lo que hice yo personalmente fue esta función, que recibe el tipo de pokemon por parámetros. Eso lo cree en un archivo javascript y lo exporté, así como si fuera un componente. Por cada tipo de pokemon, esta función devolverá un string que será el hexagesimal  de un color

const ValidateColor = (type) => {

if(type === 'normal' || type === 1){

return "#735159";

} else if(type === 'fighting' || type === 2){

return "#973f26";

} else if(type === 'flying' || type === 3){

return "#48677b";

} else if(type === 'poison' || type === 4){

return "#5b2d86";

} else if(type === 'gound' || type === 5){

return "#a37324";

} else if(type === 'rock' || type === 6){

return "#46180b";

} else if(type === 'bug' || type === 7){

return "#8bc34a";

} else if(type === 'ghost' || type === 8){

return "#31336a";

} else if(type === 'steel' || type === 9){

return "#5d736c";

} else if(type === 'fire' || type === 10){

return "#fb6c6c";

} else if(type === 'water' || type === 11){

return "#70b7fa";

} else if(type === 'grass' || type === 12){

return "#48d0b0";

} else if(type === 'electric' || type === 13){

return "#e2e02d";

} else if(type === 'phychic' || type === 14){

return "#a12b6a";

} else if(type === 'ice' || type === 15){

return "#86d2f4";

} else if(type === 'dragon' || type === 16){

return "#448a94";

} else if(type === 'dark' || type === 17){

return "#030706";

} else if(type === 'fairy' || type === 18){

return "#981844";

} else if(type === 'shadow' || type === 19){

return "#000000";

} else {

return "#ffeb3b";

}

}

export default ValidateColor;

Después en el componente lo importaba y lo ponía en la propiedad style:

import ValidateColor from "../ValidateColor.js";

const PokedexItem = ({ url }) => {

...

return(

<Link

style={{background: ValidateColor(pokemon.type)}}

>

...

</Link>

)

}

export default PokedexItem;

