

React JS – Cheat Sheet

COMPONENTES DE TELA

```
<div style={{}></div>
<p>Texto a ser mostrado na tela</p>
<img src={<objeto image>} ... </img>
<input type="text" placeholder="..." value={...}
onChangeText={...}/>
<button onClick = {...} type="button | submit"> <texto a
ser mostrado> </button>
<table>
  <tr>
    <td> Coluna 1 </td>
    <td> Coluna 2 </td>
  </tr>
</table>
```

DECLARAÇÃO DE VARIÁVEIS, FUNÇÃO, OPERADORES E CONTROLE DE FLUXO

```
const arrowFunction = () => { ... }
function Item( props ) { .... }
export default () => { ... }
const [id, setId] = useState("");
const obj = { id: "1", nomeVeiculo: "March", anoVeiculo:
2018}
var a = 1; let b = 2; const c = 3;
while ( <condição> ) { .... }
for (let i = <início>; i <<quantidade>; <incremento>) { ...
}
```

REDUCER

```
const estadoInicial = {
  <variavel de estado> : {},
  <função>( <parâmetros> ): ()=>{}
}

const <funcaoReducer> = ( <estadoAtual>, <action> ) =>
{
  ...
  return <nova versão do estado>;
};

const {type, payload} = <action>;
const [<estado atual>, <função dispatcher>] =
useReducer(<funcaoReducer>, <estadoInicial>);

// Invocando a função dispatcher
<função dispatcher>({
  type: "...", payload: { ... }
});
```

LISTAS

```
let arr = [1, 2, 3];
arr.push( 4 );
```

ESTILOS

```
marginTop: 10
backgroundColor: "cyan"
flex: <peso>
```

AVISOS

```
alert("Texto a ser exibido");
```

ACESSO A API REST

```
const api = axios.create({baseUrl: "https://xyz.com"});
api.post( <nome do recurso>, <objeto> )
api.delete( <nome do recurso> )
api.get( <nome do recurso> )
.then((response)=>{})
.catch((error)=>{})
```

CUSTOM HOOKS

```
const use<Nome da Funcao Hook> = () => {
  ...
  return { ... }
}
```

CONTEXT API

```
const <Componente Contexto> =
createContext(<contexto inicial>);
const <contexto inicial> = { ... }
const <contexto atual> = useContext( <Componente
Contexto> );
<ComponenteContexto.Provider
  value={ <contexto atual> }>
  ...
</ComponenteContexto.Provider/>
```

VALIDAÇÃO COM YUP

```
const <schema> = object( {
  <nome do campo>:
  <tipo>().<validacao1>().<validacaoN>();
} )
string()
required([<mensagem>])
number()
min(<tamanho ou menor valor valido>, [<mensagem>])
max(<tamanho ou maior valor valido>, [<mensagem>])
boolean()
email([<mensagem>])
date()
```