



REACT-NATIVE JSX

DIN

2 DAM

Què és?

Què és?

- Segons apareix a la documentació de React, és una **‘extensió de Javascript’**, o una manera d’escriure els teus components fàcilment.
- Exemple:

```
const element = <h1>Hola, món!</h1>;
```
- El que tenim a la dreta del = no és una cadena de text o HTML, és JSX.
- JSX és javascript amb alguna funcionalitat ‘extra’.
- Ens permet escriure codi semblant a l’HTML o XML però amb l’avantatge d’afegit-li mètodes i variables javascript:
- Amb JSX, el codi queda molt més clar i entenable, encara que al principi ens coste una miqueta acostumar-nos a utilitzar-lo.

JSX

- *JSX necessita ser 'transpilat' a Javascript abans d'ésser renderitzat. És per això, que per provar els exemples necessitem engegar un projecte React Native.*
- Podem aprofitar el que crearem l'altre dia anomenat **prova**.

JSX (Expressions bàsiques)

- *Modificarem la funció App per a que se semble al següent codi:*

```
Const App = ()=> {  
  return (  
    <>  
      //Ací posarem el nostre codi  
    </>  
  );  
}
```

Codi Inicial

- Anem a fer el típic 'hola món'

```
const saluda = () => {  
  return <Text>Hola món</Text>  
};  
  
const App = () => {  
  return (  
    <>  
      {Saluda()}  
    </>  
  );  
}
```

JSX (Expressions bàsiques)

- Passant paràmetres.

```
const saluda = (unNom, unCognom) => {  
  return (  
    <View>  
      <Text>Hola món</Text>  
      <Text>Sóc {unNom} {unCognom}</Text>  
    </View>  
  );  
};
```

funció saluda()

```
class App extends Component {  
  render() {  
    return (  
      <View>  
        { //Crida a la funció saluda en una classe  
          saluda('Manel', 'Viel') }  
      </View>  
    );  
  }  
}
```

```
const App = () => {  
  return (  
    <>  
      { //Crida a la funció saluda des d'App  
        saluda('Manel', 'Viel') }  
    </>  
  );  
}
```

JSX (Expressions bàsiques)

Conditionals.

- Es permeten conditionals utilitzant l'operador terciari `condition ? exprIfTrue : exprIfFalse`
- **No** es pot utilitzar el `if`

```
let loggedIn = true;
```

```
return (  
  <View>  
    {  
      !loggedIn?  
        <Text style={styles.text_salutacio}> Primer t'has de loggear </Text>  
        : saluda('Manel', 'Viel')  
    }  
  </View>  
);
```

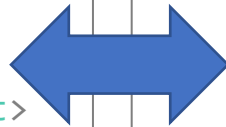
JSX (Expressions bàsiques)

Conditionals.

- I si no tenim opció 'else' per a l'operador terciari `condition ? exprIfTrue : exprIfFalse`
- **No** es pot utilitzar el **if**

```
let loggedIn = true;
```

```
return (  
  <View>  
    {  
      loggedIn?  
        <Text>{ saluda('Manel', 'Viel') }</Text>  
        : null  
    }  
  </View>  
);
```



```
return (  
  <View>  
    {  
      loggedIn &&  
        <Text>{ saluda('Manel', 'Viel') }</Text>  
    }  
  </View>  
);
```

Opció molt habitual si mireu codi d'altres desenvolupadors

JSX (Expressions bàsiques)

Bucles.

- **No** es pot utilitzar el **for**
- Es permeten funcions de recorregut d'arrays com [map\(\)](#)

```
const fruites = ['peres', 'pomes', 'taronges', 'plàtans'];
```

```
<>
  <StatusBar barStyle="dark-content" />
  <Text style={{fontSize:28}}>Llistat de fruites:</Text>
  {fruites.map((item,pos) => {
    return(<Text key={pos}>{pos+1}. - {item}</Text>);
  })}
</>
```

Quan creem una llista d'elements des de JSX, cal que afegim un identificador únic a cada element, mitjançant la propietat key.

- Es poden utilitzar altres mètodes per a manipular arrays, però recordeu separar el que és mostrar la informació, dels càlculs i la lògica. Potser caldria pensar una altra estratègia en compte de fer-ho tot en el mètode App().

JSX. Exercicis

1. Crea una nova branca anomenada JSX
- Modifica el fitxer App.js, per a que estiga buit amb un component <PaperProvider>.

```
import {Provider as PaperProvider} from 'react-native-paper';  
const App = () => {  
  return (  
    <PaperProvider>  
      </PaperProvider>  
  );  
}
```

1. Crea una funció (anomenada nom()) que mostre un component Text amb el teu nom complet
2. Modifica-la per a que el contingut del text el reba com a paràmetre **textAMostrar**
3. Torna-la a modificar per a poder passar-li l'estil a aplicar al text
4. Crea una funció (anomenada dades()) que mostre un component TextInput (de la llibreria react-native-paper).
5. Modifica la funció dades() per a que torne tants components com elements hi hagen en un array que rebrà per paràmetre amb el següent aspecte: ['Email','Nom']

Components que torna la funció dades()


Manel Viel

Email

Email

Nom

JSX. Exercicis

6. Crea els següents estils en l'objecte styles: 
7. Crearem una constant anomenada estil, i que Inicialment tindrà el valor de 'florida'.
Sobre el component pare que torna la funció dades()
Li aplicarem l'estil florida o upv, en funció del valor de la constant estil.

8. El text de cada Input, queda molt fosc, averigua
Per a que serveixen les propietats

placeholderTextColor
~~color~~

I aplica'ls un valor 'white' si l'estil és 'florida' i 'orange' si l'estil és upv.

9. Ho podries reescriure amb funcions?

```
upv: {
  backgroundColor: 'purple',
  fontSize: 10,
  fontWeight: '600',
  padding: 4,
  paddingLeft: 12,
  textAlign: 'left',
  color: 'grey',
},
florida: {
  backgroundColor: 'red',
  fontSize: 12,
  fontWeight: '600',
  padding: 4,
  paddingRight: 12,
  textAlign: 'right',
},
```

JSX. Exercicis

10. Crea una constant anomenada `isAdmin`, amb el valor a `true`.

Crea una funció que mostre un botó com este:



La icona és 'format-list-bulleted'.

Si `isAdmin` és `true`, caldrà mostrar el botó, sinó, no s'ha de mostra res.

11. Crea una constant anomenada `moduls2dam`

amb els següents valors:

```
const moduls2Dam = [  
  { nom: 'DIN', professor: 'Manel', hores: 120 },  
  { nom: 'ADA', professor: 'Roberto', hores: 120 },  
  { nom: 'PMDM', professor: 'Paco', hores: 100 },  
  { nom: 'PSP', professor: 'Roberto', hores: 60 },  
  { nom: 'SGE', professor: 'Belén', hores: 100 },  
  { nom: 'Anglés', professor: 'Caterina', hores: 40 },  
  { nom: 'EIE', professor: 'Ana', hores: 60 },  
];
```

Fes que recorreguent l'array ho mostre com en la figura: Utilitza els colors indicats:

`isAdmin true`

`isAdmin false`

1	DIN	Manel	120 hores	#F48FB1
2	ADA	Roberto	120 hores	#F8BBD0
3	PMDM	Fran	100 hores	
4	PSP	Roberto	60 hores	
5	SGE	Belén	100 hores	
6	Anglés	Caterina		

Recordeu sol·licitar una
tutoria si no teniu clars
alguns conceptes.