**React**

**Impresión de variables en el HTML.**

Documentación: <https://es.reactjs.org/>

<https://es.reactjs.org/docs/getting-started.html>

Dentro de lo que es la pieza del componente en el segmento, fragmento, div o lo que sea que haga el return. Si nosotros queremos reemplazar algo por alguna expresión cual sea de JavaScript, debemos utilizar los **“{}”** dentro de estas llaves, podemos colocar una expresión permitida de JavaScript que no sea un objeto. Una expresión de JS podría ser lo siguiente: **{ 1 + 1 } = 2**, **{ 1 + 2 } = 3** , **{ 1 + 3 } = 4, { 2 \* 3 } = 6**

Siempre que sea una expresión y no un objeto React lo va a interpretar. **¿Pero por qué un objeto no?** Los objetos no son permitidos como un hijo jsx de React y si nosotros necesitamos imprimir un objeto, podemos utilizar el **JSON.stringify**. Ejemplo de impresión de variables:

/\*\*

 \* Si la variable que estamos usando, no tiene relación con ningún Hook o algo así,

 \* entonces no tiene sentido que este dentro del componente y se podría sacar del mismo.

 \* Ejemplo quedando así:

 \*/

const $Mensaje = "Muchas gracias por venir Benjamín, que te vaya bien";

const $Lista = [1,2,3,4,5,6,7,8,9];

// Los objetos no se pueden imprimir dentro de unos "{}", por que no son permitidos como un hijo jsx de React y si nosotros necesitamos imprimir un objeto, podemos utilizar el JSON.stringify

const $Persona = {

    Nombres: "Benjamín",

    Apellidos: "Ayala",

}

export const Despido3 = () => {

    {/\*

        NOTA IMPORTANTE.

        Todo código que nosotros pongamos aquí, en este espacio dentro del componente, al momento de renderizar el componente

        también se estaría renderizando, como la función $Suma, que esta abajo...

¿CUÁNDO PONER EL CODIGO ADENTRO O AFUERA?

Cuando necesitemos que el componente haga un cambio en el estado, pero si no hay

ninguna dependencia del código al componente, es mejor sacarlo...

    \*/}

    const $Sumar = ( $Valor1 , $Valor2 ) => ($Valor1 + $Valor2);

    // Hay que tener cuidado cuando utilizamos los async - await en las funciones y los pongamos dentro de los componentes, recordemos que una función asincrónica devuelve un objeto

    return (

        <>

            { $Mensaje} - { $Lista } - { $Persona.Nombres + ' ' + $Persona.Apellidos  }

            <code>{ JSON.stringify( $Persona ) }</code>

            {/\*

                Esto es un comentario en React, esto no va a aparecer en el DOM.

            \*/}

            Suma de 4 + 5 = { $Sumar( 4 , 5) }

        </>

    );

}