Current Skill Variables

Basic examples

HTML

```
<p id=”my-name”></p>
```

JavaScript

```
let firstName = 'Will';
let lastName = 'Smith';
document.getElementById('my-name').innerHTML = firstName + ' ' + lastName;
```

JSX

```
let firstName = "Will";
let lastName = "Smith";

return <p>{firstName + " " + lastName}</p>;
```

To use an expression (operation or variable) inside JSX, you only need to add **curly brackets {}**.

JSX type

As we have already established, JSX is all about writing HTML inside of JavaScript files, so it can automatically inherit of JavaScript’s power.

In JavaScript, we can assign mostly everything into a variable. Similarly, when using JSX, that fact remains valid. We can create a JSX element and assign it to a variable. The example below will explain it more:

```
function App(){
let input = (<input type='text' placeholder='Name'/>);
let button = <button>Submit</button>;
let form = (
  <form>
    {input}
    {button}
  </form>
);

return form;
}
```

JSX type

Since JSX can be stored in a variable, that means it can also return it in a function.

Here’s how :

```
function App() {
 // We can put functions inside other functions when they are related
 function createForm() {
   let input = <input type="text" placeholder="Name" />;
   let button = <button>Submit</button>;
   return (
     <form>
       {input}
       {button}
     </form>
   );
 }
 return <div>{createForm()}</div>;
}
```

Calling Function

HTML

```
<p id=”my-age”></p>
```

JavaScript

```
document.getElementById(“my-age”)

       .innerHTML= getAge(1996)
```

JSX

```
<p>{getAge(1996)}</p>
```

If you need to call a function (any function), you can do so by calling it inside **curly brackets {}**.

Note: We didn’t declare the getAge() function for simplicity’s sake.

‹ Previous

next ›