**Injecting data to a JSX Element**

So far, we used static data on the JSX elements, but we can also pass different data types as a dynamic data. The dynamic data could be string, number, boolean, array or object. Let us see each of the data types step by step. To inject data to a JSX we use the {} bracket.

const welcome = 'Welcome to 30 Days Of React'

const title = 'Getting Started React'

const subtitle = 'JavaScript Library'

const authorFirstName = 'Asabeneh'

const authorLastName = 'Yetayeh'

const date = 'Oct 1, 2020'

// JSX element, header

const header = (

<header>

<div className='header-wrapper'>

<h1>{welcome}</h1>

<h2>{title}</h2>

<h3>{subtitle}</h3>

<p>

Instructor: {authorFirstName} {authorLastName}

</p>

<small>Date: {date}</small>

</div>

</header>

)

Similar to the header JSX element, we can implement data injection to main and footer JSX elements.

**Injecting a string to a JSX Element**

In this section we inject only strings

const welcome = 'Welcome to 30 Days Of React'

const title = 'Getting Started React'

const subtitle = 'JavaScript Library'

const firstName = 'Asabeneh'

const lastName = 'Yetayeh'

const date = 'Oct 2, 2020'

// JSX element, header

// JSX element, header

const header = (

<header>

<div className='header-wrapper'>

<h1>{welcome}</h1>

<h2>{title}</h2>

<h3>{subtitle}</h3>

<p>

Instructor: {firstName} {lastName}

</p>

<small>Date: {date}</small>

</div>

</header>

)

**Injecting a number to a JSX Element**

const numOne = 3

const numTwo = 2

const result = (

<p>

{numOne} + {numTwo} = {numOne + numTwo}

</p>

)

const yearBorn = 1820

const currentYear = new Date().getFullYear()

const age = currentYear - yearBorn

const personAge = <p> {age}</p>

As you can see in the example above, it is possible to do some arithmetic calculations and ternary operations.

**Injecting an array to a JSX Element**

To give an example for an array, let us change the HTML, CSS, JavaScript to an array and inject it to the main JSX element below. We will cover more in much detail later, in rendering lists section.

const techs = ['HTML', 'CSS', 'JavaScript']

// JSX element, main

const main = (

<main>

<div className='main-wrapper'>

<p>

Prerequisite to get started{' '}

<strong>

<em>react.js</em>

</strong>

:

</p>

<ul>{techs}</ul>

</div>

</main>

)

**Injecting an object to a JSX Element**

We can inject string, number, boolean, array data to JSX but we cannot directly inject an object. We should extract object values first or destructure the content of the object before we inject the data to the JSX element. For instance, let us write firstName and lastName inside an object and extract them to use them inside JSX.

Now, let us put everything together. Here, in the example below, the data is injected dynamically to the JSX. [Live on code pen](https://codepen.io/Asabeneh/full/YzXWgpZ)

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<link

href="https://fonts.googleapis.com/css?family=Montserrat:300,400,500|Roboto:300,400,500&display=swap"

rel="stylesheet"

/>

<title>30 Days Of React Challenge</title>

<style>

/\* == General style === \*/

\* {

box-sizing: border-box;

padding: 0;

margin: 0;

}

html,

body {

height: 100%;

line-height: 1.5;

font-family: 'Montserrat';

font-weight: 300;

color: black;

}

.root {

min-height: 100%;

position: relative;

}

.header-wrapper,

.main-wrapper,

.footer-wrapper {

width: 85%;

margin: auto;

}

.header-wrapper,

.main-wrapper {

padding: 10px;

margin: 2px auto;

}

h1 {

font-size: 70px;

font-weight: 300;

}

h2,

h3 {

font-weight: 300;

}

header {

background-color: #61dbfb;

padding: 10px;

}

main {

padding: 10px 10px 60px;

/\* Height of the footer \*/

}

ul {

margin-left: 15px;

}

ul li {

list-style: none;

}

footer {

position: absolute;

bottom: 0;

width: 100%;

height: 60px;

/\* Height of the footer \*/

background: #6cf;

}

.footer-wrapper {

font-weight: 400;

text-align: center;

line-height: 60px;

}

</style>

</head>

<body>

<div class="root"></div>

<script

crossorigin

src="https://unpkg.com/react@16/umd/react.development.js"

></script>

<script

crossorigin

src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"

></script>

<script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>

<script type="text/babel">

// To get the root element from the HTML document

const rootElement = document.querySelector('.root')

// JSX element, header

const welcome = 'Welcome to 30 Days Of React'

const title = 'Getting Started React'

const subtitle = 'JavaScript Library'

const author = {

firstName: 'Asabeneh',

lastName: 'Yetayeh',

}

const date = 'Oct 2, 2020'

// JSX element, header

const header = (

<header>

<div className='header-wrapper'>

<h1>{welcome}</h1>

<h2>{title}</h2>

<h3>{subtitle}</h3>

<p>

Instructor: {author.firstName} {author.lastName}

</p>

<small>Date: {date}</small>

</div>

</header>

)

const numOne = 3

const numTwo = 2

const result = (

<p>

{numOne} + {numTwo} = {numOne + numTwo}

</p>

)

const yearBorn = 1820

const currentYear = new Date().getFullYear()

const age = currentYear - yearBorn

const personAge = (

<p>

{' '}

{author.firstName} {author.lastName} is {age} years old

</p>

)

// JSX element, main

const techs = ['HTML', 'CSS', 'JavaScript']

// JSX element, main

const main = (

<main>

<div className='main-wrapper'>

<p>

Prerequisite to get started{' '}

<strong>

<em>react.js</em>

</strong>

:

</p>

<ul>{techs}</ul>

{result}

{personAge}

</div>

</main>

)

const copyRight = 'Copyright 2020'

// JSX element, footer

const footer = (

<footer>

<div className='footer-wrapper'>

<p>{copyRight}</p>

</div>

</footer>

)

// JSX element, app

const app = (

<div className='app'>

{header}

{main}

{footer}

</div>

)

// we render the JSX element using the ReactDOM package

ReactDOM.render(app, rootElement)

</script>

</body>

</html>