https://academind.com/tutorials/reference-vs-primitive-values/

Javascript :

Primitive datatypes:

String, number, boolea, undefined, null , => store to stack

Referance types:

Object: => array =>store to heap

**What are Primitives?**

This article and videos is named “Reference vs Primitive Values”.

So let’s start - what are “Primitives”?

Here’s an example:

**var age = 28**

The **age** variable (you could also use **let** or **const** by the way) stores a number value. The number **28**.

Number values are called “primitive values” because they’re very simple building blocks of JavaScript apps.

Other simple core building blocks are:

**var name = 'Max' // strings are primitives, too!**

**var isMale = true // so are booleans**

So numbers, string, booleans - these are probably very well-known to you. **undefined** and **null** are additional primitive types.

* Adding img from folder to component:
  + First import that img to component
  + Then add to src

import React from "react";

import mealImage from '../../assets/meals.jpg';

const Image = () => {

  return <React.Fragment>

        <img src={mealImage} alt=""/>

  </React.Fragment>;

};

export default Image;

* Is in \_.module.css file class name contain – then to use that class in component.

import styles from "./Header.module.css";

<div className={styles['main-image']}>

        <img src={mealImage} alt="A table full of delicious food!"/>

      </div>

* Creating Input Component and use it:

import React from 'react';

import styles from './Input.module.css';

const Input = (props) =>{

    return<div className={styles.input}>

        <label htmlFor={props.input.id}>{props.label}</label>

        <input {...props.input}/>

    </div>

}

export default Input;

 <Input

        label="Amount"

        input={{

          id: 'amount\_' + props.id,

          type: "number",

          min: "1",

          max: "5",

          step: "1",

          defaultValue: "1",

        }}

      />

* Whenever we update state eg. setCondition( ) use previous state data to update data

SetCondition( (value) => { return !value } );

* Modal using react portals :

import React from "react";

import ReactDOM from "react-dom";

import styles from "./Modal.module.css";

const Backdrop = (props) => {

  return <div className={styles.backdrop}></div>;

};

const ModalOverlay = (props) => {

  return (

    <div className={styles.modal}>

      <div className={styles.content}>{props.children}</div>

    </div>

  );

};

const portalElement = document.getElementById('overlays')

const Modal = (props) => {

  return (

    <React.Fragment>

      {ReactDOM.createPortal(<Backdrop />,portalElement)}

      {ReactDOM.createPortal(<ModalOverlay>{props.children}</ModalOverlay> , portalElement)}

    </React.Fragment>

  );

};

export default Modal;

.backdrop {

  position: fixed;

  top: 0;

  left: 0;

  width: 100%;

  height: 100vh;

  z-index: 20;

  background-color: rgba(0, 0, 0, 0.75);

}

.modal {

  position: fixed;

  top: 20vh;

  left: 5%;

  width: 90%;

  background-color: white;

  padding: 1rem;

  border-radius: 14px;

  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.25);

  z-index: 30;

  animation: slide-down 300ms ease-out forwards;

}

@media (min-width: 768px) {

  .modal {

    width: 40rem;

    left: calc(50% - 20rem);

  }

}

@keyframes slide-down {

  from {

    opacity: 0;

    transform: translateY(-3rem);

  }

  to {

    opacity: 1;

    transform: translateY(0);

  }

}

Using modal

import React from "react";

import Modal from "./Modal";

const Cart = (props) => {

  return (

    <Modal>

       //Content in modal

    </Modal>

  );

};

export default Cart;