**Snap Scrolling**

html {

    scroll-snap-type: y mandatory;

    scroll-behavior: smooth;

}

.section {

    height: 100vh;

    scroll-snap-align:center;

}

**Create a Shortcut in Visual Studio Code:**

CTRL + Shift + p 🡪 type “snippets” then select “Snippets: Configure Snippets” then type and choose “javascriptreact.json”

Copy and Paste this code in the json file:

{

    // Place your snippets for javascriptreact here. Each snippet is defined under a snippet name and has a prefix, body and

    // description. The prefix is what is used to trigger the snippet and the body will be expanded and inserted. Possible variables are:

    // $1, $2 for tab stops, $0 for the final cursor position, and ${1:label}, ${2:another} for placeholders. Placeholders with the

    // same ids are connected.

    // Example:

    // "Print to console": {

    //  "prefix": "log",

    //  "body": [

    //      "console.log('$1');",

    //      "$2"

    //  ],

    //  "description": "Log output to console"

    // }

    "fcc": {

        "prefix": "fcc",

        "body": [

            "import './${TM\_FILENAME\_BASE/^(.)/${1:/downcase}/}.css'""""const ${1:${TM\_FILENAME\_BASE/(.)(.\*)/${1:/capitalize}${2}/}} = () => {",

            "  return (",

            "    <div className='${TM\_FILENAME\_BASE/^(.)/${1:/downcase}/}'>${1:${TM\_FILENAME\_BASE/(.)(.\*)/${1:/capitalize}${2}/}}</div>",

            "  )",

            "}",

            "",

            "export default ${1:${TM\_FILENAME\_BASE/(.)(.\*)/${1:/capitalize}${2}/}}"

        ],

        "description": "Create an arrow component with css"

    },

    "fcs": {

        "prefix": "fcs",

        "body": [

            "import './${TM\_FILENAME\_BASE/^(.)/${1:/downcase}/}.scss'""""function ${1:${TM\_FILENAME\_BASE/(.)(.\*)/${1:/capitalize}${2}/}}(){",

            "  return (",

            "    <div className='${TM\_FILENAME\_BASE/^(.)/${1:/downcase}/}'>${1:${TM\_FILENAME\_BASE/(.)(.\*)/${1:/capitalize}${2}/}}</div>",

            "  )",

            "}",

            "",

            "export default ${1:${TM\_FILENAME\_BASE/(.)(.\*)/${1:/capitalize}${2}/}}"

        ],

        "description": "Create a functional component with Sass"

    },

    "acs": {

        "prefix": "acs",

        "body": [

            "import './${TM\_FILENAME\_BASE/^(.)/${1:/downcase}/}.scss'""""const ${1:${TM\_FILENAME\_BASE/(.)(.\*)/${1:/capitalize}${2}/}} = () => {",

            "  return (",

            "    <div className='${TM\_FILENAME\_BASE/^(.)/${1:/downcase}/}'>${1:${TM\_FILENAME\_BASE/(.)(.\*)/${1:/capitalize}${2}/}}</div>",

            "  )",

            "}",

            "",

            "export default ${1:${TM\_FILENAME\_BASE/(.)(.\*)/${1:/capitalize}${2}/}}"

        ],

        "description": "Create an arrow component with Sass"

    },

    "comp": {

        "prefix": "comp",

        "body": [

            "const ${1:${TM\_FILENAME\_BASE/(.)(.\*)/${1:/capitalize}${2}/}} = () => {",

            "  return (",

            "    <div>${1:${TM\_FILENAME\_BASE/(.)(.\*)/${1:/capitalize}${2}/}}</div>",

            "  )",

            "}",

            "",

            "export default ${1:${TM\_FILENAME\_BASE/(.)(.\*)/${1:/capitalize}${2}/}}"

        ],

        "description": "Create a component"

    },

    "compt": {

        "prefix": "compt",

        "body": [

            "const ${1:${TM\_FILENAME\_BASE/(.)(.\*)/${1:/capitalize}${2}/}} = () => {",

            "  return (",

            "    <div className=''>${1:${TM\_FILENAME\_BASE/(.)(.\*)/${1:/capitalize}${2}/}}</div>",

            "  )",

            "}",

            "",

            "export default ${1:${TM\_FILENAME\_BASE/(.)(.\*)/${1:/capitalize}${2}/}}"

        ],

        "description": "Create a component with tailwind classname"

    }

}

After that just create a new component and type “compt” then tab.

## **Create 3D image**

download ‘gltf’ 3d image from:

<https://sketchfab.com/>

install: npm install -g gltf-pipeline

to create glb file from gltf type: gltf-pipeline -i scene.gltf -b

convert glb to jsx type: npx gltfjsx car.glb









