Are you a creative frontend web designer or are you looking for ways to make your website look more interactive and visually appealing? Would you like to advertise your products interestingly? Look no further. Today, we are going to learn how to insert 3D objects into a website.

In this tutorial, we will be creating a simple webpage with basic components, design its layers, and then inserting the 3D object into the webpage.

Key takeaways:
At the end of this tutorial, the user will get to learn about:

3D objects, formats, and some of their properties
How to create or acquire 3D objects for their website
Inserting a 3D object in a website
Formatting the 3D object inserted into the webpage
Prerequisites
Before we begin, some requirements of this tutorial include:

Basic knowledge and understanding of HTML and web development.
A basic understanding of CSS.
A basic web development IDE or a text editor installed on your machine. In our case, let's use Visual Studio Code.
A stable internet connection
3D Objects
Let us learn briefly about 3D Objects:

What are 3D objects?
If this question is in your mind, we can easily answer it by saying that 3D objects or 3D models are shapes with three dimensions; length, width, and height.

How to get 3D Objects
We can obtain 3D Objects by either creating one from scratch, scanning real-life objects, modifying templates to suit your need, and downloading online models. We are going to look briefly at each of the methods mentioned above.

Creating 3D objects from scratch
One may create 3D objects from scratch using 3D model creation software which is downloadable into the computer. One may also design it online at 3D modeling sites.

Scanning from real-life objects
You can create one by scanning the item with your scanner of choice.
Photogrammetry is another process one may use.

Photogrammetry can be simply defined as the art of acquiring reliable measurements from 2D images by overlapping them to form a 3D model.
This method is much simpler and more preferable since one can use 2D images generated with a smartphone and obtain live 3D models. An example is illustrated here.

Modifying existing templates
One can modify existing templates that are on the computer or were previously created for another project to suit your needs.

Online download
There are many sites with different 3D models. Examples that allow free downloads are Sketchfab and Google Poly among others.

3D Model Formats
These are used to store the appearance of a model, encoding the model animations,

encoding the geometry, and storing the physical appearance of the model. They are used in video games, 3D printing, engineering, movies and so much more.
Some popular formats available include FBX, STL, OBJ, glTF, and DAE among others. In our case, we will use the glTF model.

glTF, Graphics Language Transmission Format, is a 3D model format designed for efficient transmission and loading of 3D models by applications.

After visiting the above sites or any other, search for any preferable model and download it, in this case, we are going to search, download and use a Virtual Reality Headset (VR Headset) model which is in glTF format.

VR Headsets

3D Model Attributes
These are the characteristics of the viewed 3D Object on the website set by the frontend developers as preferred. Some attributes one can set to 3D models include:

Auto-Rotate
Alternative text(alt)
Model-Visibility
Augmented Reality
Touch-action
Camera-controls
Source(src)
ios-src and many more...
Let's now get into the tutorial:
First things first, let's set up our Visual Studio Code IDE. In the extensions tab, search for the "HTML Boilerplate" and "Live Server" extensions and install them. Create a folder and name it "3D Objects". Open the folder with Visual Studio Code and in it create other two folders namely "assets" and "css". Create an "index.html" file in the main directory and "style.css" in the CSS folder.

Create a webpage structure
We are going to create a simple HTML structure. We can simplify the process using boilerplate generators which may come pre-installed by default in Visual Studio Code or as downloadable extensions by typing in the index file, "html:5" or "html5-boilerplate" then key in the Enter key. Since we don't need additional elements in the boilerplate generated, we will delete the extra code and link it to our CSS file. Let's also link it with some online scripts which will load some Font Awesome fonts and icons. The outcome will be as shown below:

```
 !DOCTYPE html
 html lang="en"
 head
     meta charset="UTF-8"
     title 3D Object /title
     script src='https://kit.fontawesome.com/a076d05399.js'  /script
     link rel="stylesheet" href="css/style.css"

 /head
 body

 /body
 /html
```

Start the live server by either right-clicking on the "index.html" file in the explorer and selecting "Open with Live server" or on the status bar clicking on "Go live" to start the server. It will automatically open a preview of the webpage in the default browser which will be blank because we haven't added any content yet. I

recommend you use chrome since it supports most 3D objects. You can change the default browser in the settings. If it doesn't automatically open the webpage preview in the browser, check the port number in which the server uses at the Visual Studio Code status bar then add access it in the browser by typing in the URL:

http://127.0.0.1:5501/

Port numbers vary from one to another. In my case, it is port number 5501. If the server doesn't launch completely, stop any services with a similar address running in the background. If it still has errors please look for additional support online concerning the error it displays.

In the head section, add a link to import a model-viewer JavaScript code. This enables you to import the component into the webpage.

```
     !--Imports a model-viewer JavaScript code --
     !--It helps to handle how the 3D Object would be displayed --
     script type="module" src="https://unpkg.com/@google/model-viewer/dist/model-viewer.min.js"  /script
     script nomodule src="https://unpkg.com/@google/model-viewer/dist/model-viewer-legacy.js"  /script
```

Copy your logo picture in the assets folder.
When done, add a div element of id "container" to hold all our content in the body tags and also another in it for our navigation bar. Let us also add div elements with the following id; "aside" for holding our 3D object, "content" for some explanations on the object, and "icons" for some social media icons. Add a simple logo and some links to the webpage and some of their properties in css as shown below:

```
 !-- Main Container --
 div id="container"

     !-- Navbar container --
     div id="navbar"
             div id="logo"
                 img src="assets/logo.png" alt="logo"
             /div
             ul
                 li class="active"
                     a href="#" Home /a
                 /li
                 li  a href="#" Tech /a  /li
                 li  a href="#" About /a  /li
                 li  a href="#" Contact /a  /li
                 li  a href="#" Help /a  /li
             /ul
         /div

         !-- An aside --
         !-- This is for holding the 3D object --
         div id="aSide"

         /div

         !-- Content container --
         !-- This is for holding some brief description about the 3D object --
         div id="content"
             h2 3D VR LIVE /h2
```

```
            p The modern headsets, boost your pleasure to the MAX! /p
            button Shop /button
        /div

        !-- Social media icons --
        !-- This is for holding some company's social media icons and related
links --
        div id="icons"
            div id="iconsLogo"
                i class="fab fa-facebook-f"  /i
                i class="fab fa-instagram"  /i
                i class="fab fa-youtube"  /i
                i class="fab fa-twitter"  /i
            /div
        /div
 /div
```