

A screenshot of a terminal window in Visual Studio Code (VS Code) running on Windows. The terminal is set to the 'powershell' profile. The title bar shows the search term 'Day 28'. The terminal tab is active, displaying the command `node ./index.js` which has produced the following output:

```
PS D:\My Workspace1\Web Development(Delta)\WEB CODE\Node js\Day 28\Figlet> node ./index.js
 Day 28
```

The output is a large, stylized text representation of the words "Day" and "28" using a font generator like Figlet. The "D" in "Day" is particularly prominent, featuring a vertical bar with a diagonal stroke and a horizontal bar extending from its right side.

At the bottom of the terminal, there is a small blue rectangular placeholder for a new command or input. The status bar at the bottom of the screen shows the current file path: 'D:\My Workspace1\Web Development(Delta)\WEB CODE\Node js\Day 28\Figlet', the line and column number 'Ln 3, Col 12', the character encoding 'UTF-8', the line endings 'CRLF', and the file type 'JavaScript'. There are also icons for Go Live and a bell通知.

\* What is templating?

EJS :- Embedded Javascript templates

EJS is a simple templating language that lets you generate HTML markup with plain Javascript

\* 1] npm init -y

This command is used to initialize a new node.js project & create a package.json file which default value without going through an interactive process  
-y : yes.

2] npm install express

3] npm install ejs

4] touch index.js

\* Using ejs

```
app.set("view engine", "ejs");
```

view → template

```
app.get("/", (req, res) => {
  res.render("home.ejs");
});
```

Explanation :-

The line app.set(); is used in node.js app with the express framework to set EJS as the template engine.

The `app.render()` method in express is used to render a view & return the rendered HTML content through a callback fn.

This method is particularly useful when you want to compile your template into template into HTML & either use it elsewhere or send it in the response.

\* How to handle error when you run code in parent directory?

① `const Path = require("Path");`

② `app.set("views", Path.join(__dirname, "/views"));`

\* Interpolation Syntax:

Interpolation refers to embedding expression into marked up text.

`<%` - 'scriptlet' tag for control-flow, no output

`<%-` 'whitespace slurping' scriptlet tag, strips all whitespace before it

`<%=` output the value into the template

`<%-` output the unescaped value into the template

`<%#` comment tag

`<%%` output a literal '`<%`'

`>%>` plain ending tag

`>%>` trim-mode tag, trims following newline,

`>%>` 'whitespace slurping' ending tag, remove all whitespace after it

## \* Conditional statement in EJS

```
<% if(dice == 6){ %>
  <h1> Nice! Roll dice again. </h1>
<% } %>
```

## \* Loop in EJS

```
<ul> for (user of follower) { %>
  <li> <%= user %> </li>
<% } %>
```

## \* include → subtemplate

```
<% - include("includes/head.ejs"); %>
```

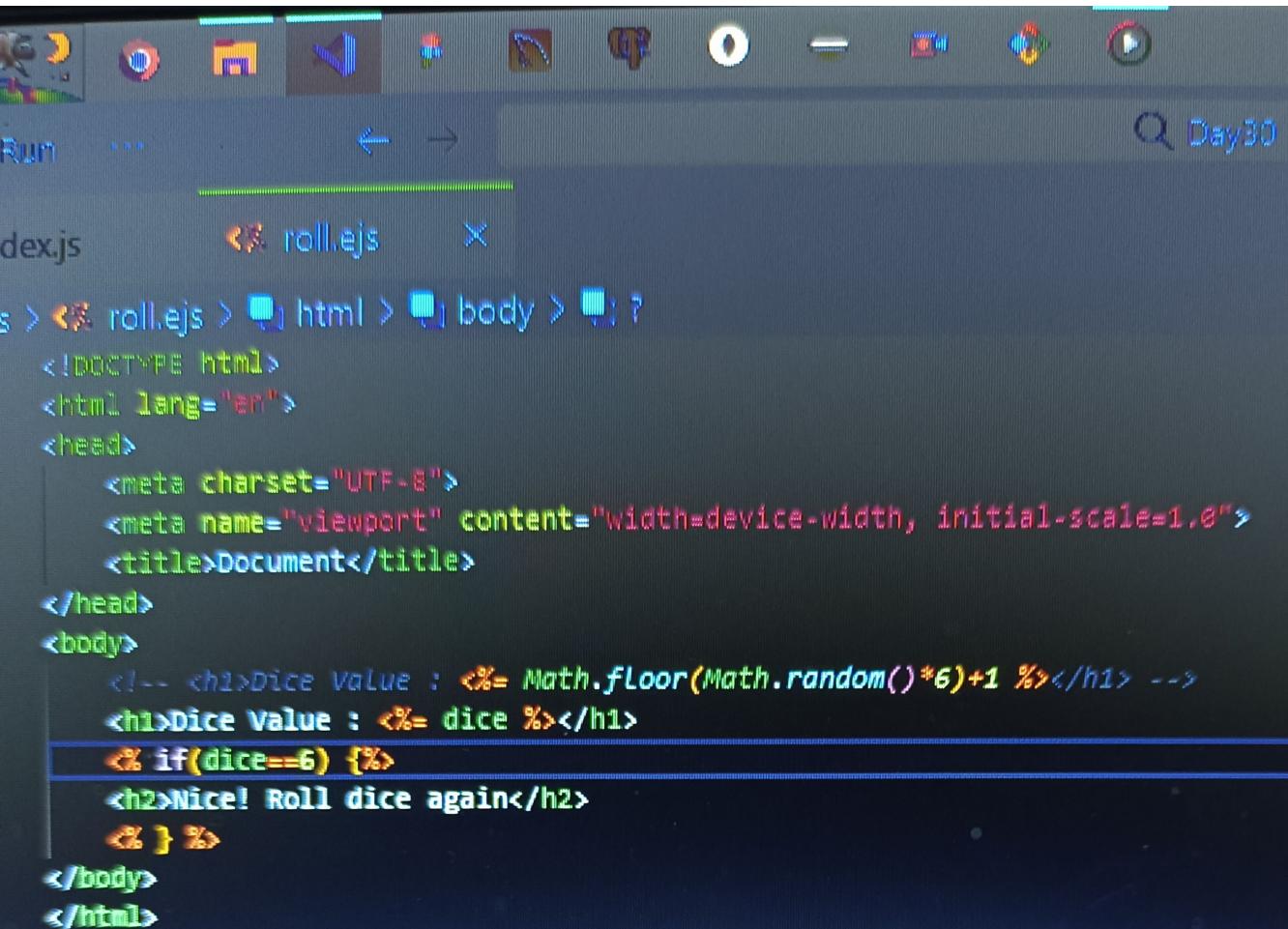
## \* Serving static file:

```
app.use(express.static(folder_name))
```

```
app.use(express.static(path.join(__dirname, "public")));
```

index.js

```
const express = require("express");
const app = express();
const path = require("path");
const port = 8080;
app.use(express.static(path.join(__dirname, "public")));
app.set("view engine", "ejs");
app.set("views", path.join(__dirname, "/views"));
app.get("/", (req, res) => {
    // res.send("This is home");
    res.render("home.ejs");
});
app.get("/home", (req, res) => {
    res.send("This is home");
    // res.render("home.ejs");
});
app.get("/roll", (req, res) => {
    let dice = Math.floor(Math.random() * 6) + 1;
    res.render("roll.ejs", { dice });
});
app.get("/ig/:username", (req, res) => {
    const {username} = req.params;
    const instadata = require("./data.json");
    const data = instadata[username];
    // console.log(data);
    if(data)
    {
        res.render("instagram.ejs", {data});
    }
    else{
        res.render("error.ejs");
    }
});
app.listen(port, () => {
    console.log(`Connect to Port: ${port}`);
});
```



The screenshot shows a code editor window with a dark theme. At the top, there's a toolbar with various icons, followed by a search bar containing the text "Day30". Below the toolbar, there are two tabs: "index.js" and "roll.ejs". The "roll.ejs" tab is currently active, displaying the following EJS code:

```
<!-- <h1>Dice Value : <%= Math.floor(Math.random()*6)+1 %></h1> -->
<h1>Dice Value : <%= dice %></h1>
<% if(dice==6) {%
<h2>Nice! Roll dice again</h2>
<% } %>
</body>
</html>
```

I

A screenshot of a Windows desktop environment. In the foreground, a code editor window is open, displaying an EJS template file named `home.ejs`. The file contains the following HTML and EJS code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Document</title>
7   </head>
8   <body>
9     <h1>This is home page</h1>
10    <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Nemo illo necessitatibus reprehenderit dignissimos iure molestiae quod, porro.</p>
11    <button>Click me</button>
12  </body>
13 </html>
```

The code editor's sidebar shows a project structure under the `DAY30` folder, including `node_modules`, `public`, `css` (with `style.css`), `js`, `views` (with `includes`, `error.ejs`, `home.ejs` selected), `instagram.ejs`, `roll.ejs`, `data.json`, `index.js`, `package-lock.json`, and `package.json`.

Search

File Edit Selection View Go Run ... ← → Q Day30

EXPLORER ... index.js roll.ejs error.ejs

DAY30 node\_modules public CSS style.css Js views includes error.ejs home.ejs instagram.ejs roll.ejs data.json index.js package-lock.json package.json

```
<% include("includes/head.ejs") %>
<body>
  <h1>No such page available</h1>
</body>
</html>
```

vivo V23 · RRK

Search

File Edit Selection View Go Run ... ← → 🔍 Day30

EXPLORER ... index.js roll.ejs instagram.ejs X

DAY30 node\_modules public CSS style.css Js views includes error.ejs home.ejs instagram.ejs X rollejs data.json index.js package-lock.json package.json

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Instagram</title>
    <link rel="stylesheet" href="/public/style.css">
</head>
<body>
    <h1>This page belong to @<%= data.name %></h1>
    <button>Follow</button>
    <button type="submit">Message</button>
    <br>
    <br>
    <p style="display: inline;">Followers:<%= data.followers %></p>&ampnbsp<p style="display: inline;"> Following :<%= data.following %></p>
    <hr><hr>

    <% for(let post of data.posts) {>
        
        <p>Likes :<%= post.likes %></p>
        <p>Comment : <%= post.comments %></p>
    <%}>
    </body>
</html>
```

A screenshot of a dark-themed code editor, likely Visual Studio Code, displaying EJS template code. The code is structured as follows:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
```

The code editor interface includes a top bar with various icons and a search bar containing "Day30". The left sidebar shows a project structure with files like index.js, roll.ejs, and head.ejs. The "includes" folder contains several EJS files: head.ejs, error.ejs, home.ejs, instagram.ejs, and roll.ejs. Other files visible include data.json, index.js, package-lock.json, and package.json.

Search

File Edit Selection View Go Run ...

Day30

EXPLORER ... index.js roll.ejs style.css

DAY30 node\_modules public CSS style.css

body{ background-color: pink; }

node\_modules

views

includes

head.ejs

error.ejs

home.ejs

instagram.ejs

roll.ejs

data.json

index.js

package-lock.json

package.json

OUTLINE

TIMELINE

Ln 3, Col 2 Spaces: 4 UTF