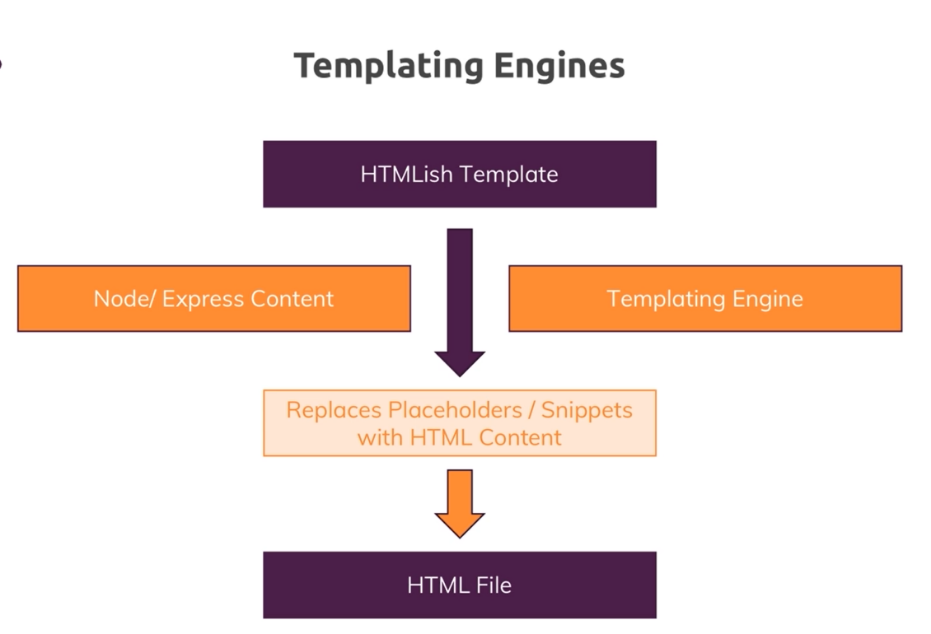
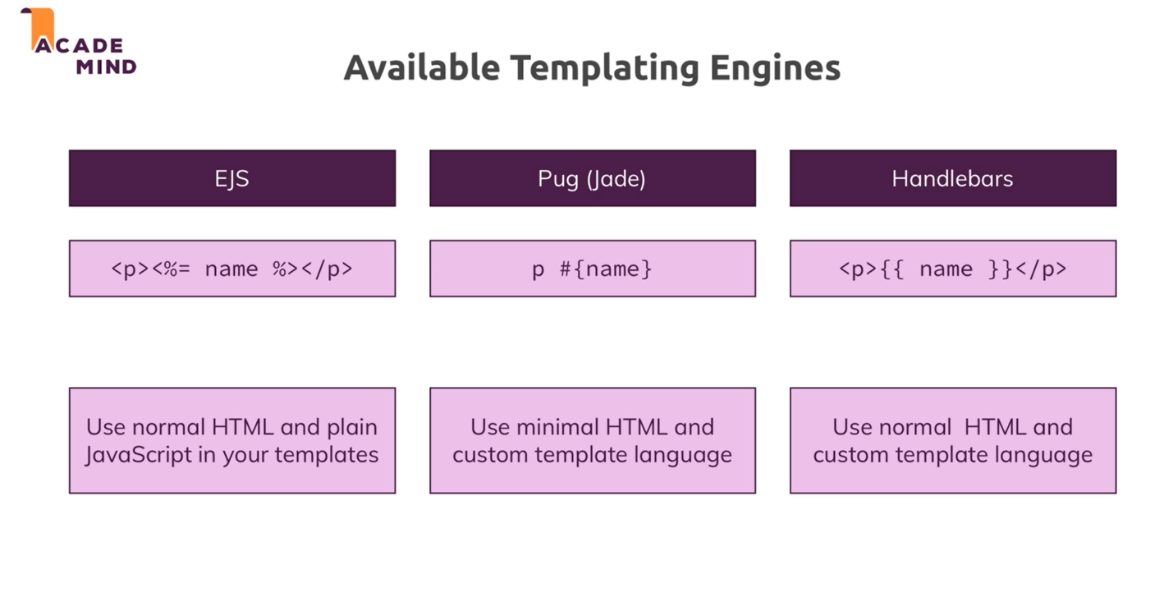
**Dynamic Content & Templates**

**Table of content**

* Managing Data (without a Database)
* Render Dynamic Content in our view
* Understanding Templating Engine

1. **Sharing Data across requests & User**a. in this, we create a global variable and we export that.  
   b. problem in this is irrespective of requests the data remain same among all users.
2. **Templating Engines**  
     
   available Templating engines:  
   a. EJS  
   b. Pug (Jade)  
   c. Handlebars
3. **Installing & using Pug**a. install pug, npm install --save pug  
   b. to use pug, go to app.js file, to node we have a templating engine pls use it for render dynamic. We do that by using set() which set a value globally and we can get that value using get()  
   c. app.set(name, value) & app.get(value)  
   d. app.set(‘view engine’, ‘pug’) => this we auto-register pug as a view engine.  
   e. app.set(‘views’ , ‘views’) =>this register our location of views folder  
   f. in above 2 steps, we are telling express we want to compile dynamic template with the pug engine and where it can find this templates(views folder)  
   g. we will render pug file, res.render(shop) were shop is file name
4. **Outputting Dynamic content in PUG**a. to send data to the pug file, we use render() as it as 2 parameters: file-name and parameter(options of type object)  
    res.render('shop', { prods: products, docTitle: 'Shop' });  
   b. use that parameters in, we use same key we added in object   
   title #{docTitle}  
   if prods.length > 0  
   each product in prods
5. **Converting large view file into small files in PUG**a. we can do this using extends and block  
   b. first we have to create a pug file with common view elements and uncommon as block (declare block)  
   c. we will extend another file and we will define block
6. **Working with HandleBar**a. first we have import express-handlebar:   
   const expressHbs = require(‘express-handlebar’);  
   b. now we have told express this engine exist:   
   app.engine(‘handlebars’, expressHbs);  
   OR   
   app.engine(‘hbs’, expressHbs);  
   c. now we have to register engine:  
   app.set(‘view engine’, ‘handlerbars’);  
   OR  
   app.set(‘view engine’, ‘hbs’);  
   d. now register location of view file,  
   app.set(‘views’ , ‘views’)  
   e. file extension should handlebars or hbs or any other you had used in above function  
   f. to display dynamic content we pass object in res.render() and use {{ key-name }} => to display it  
   g. {{#if }}, {{/if}}, {{#each }}  
   h. {{{}}}=> place-holder
7. **Working with EJS**a. for this we don’t need to import any package and we have to register engine same as pub.  
   app.set(‘view engine’, ‘ejs);  
   app.set(‘views’ , ‘views’);  
   b. to get dynamic data(value), we use syntax: <%= key-name%>  
   c. using <% %> we can use js code in html file.  
   eg.   
   <%if(10>9){%>  
    <h1> Samar </h1>  
   <%}%>  
   d. <%= include(‘file-path’)%> => will display text of the file  
   e. <%- include(‘file-path’)%> => will render html file