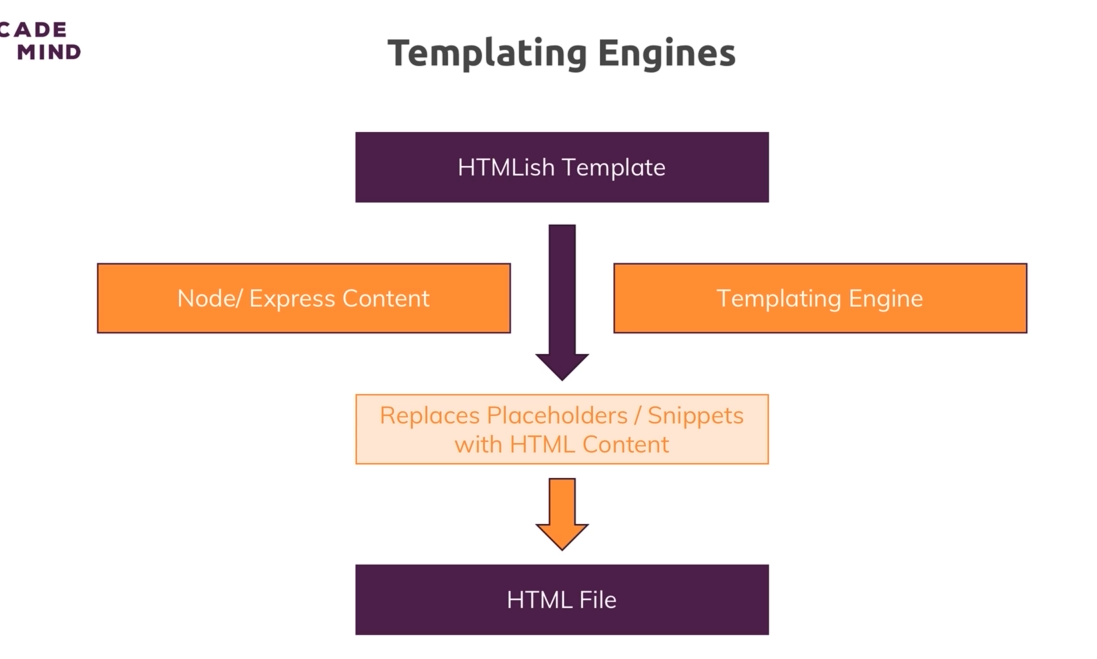
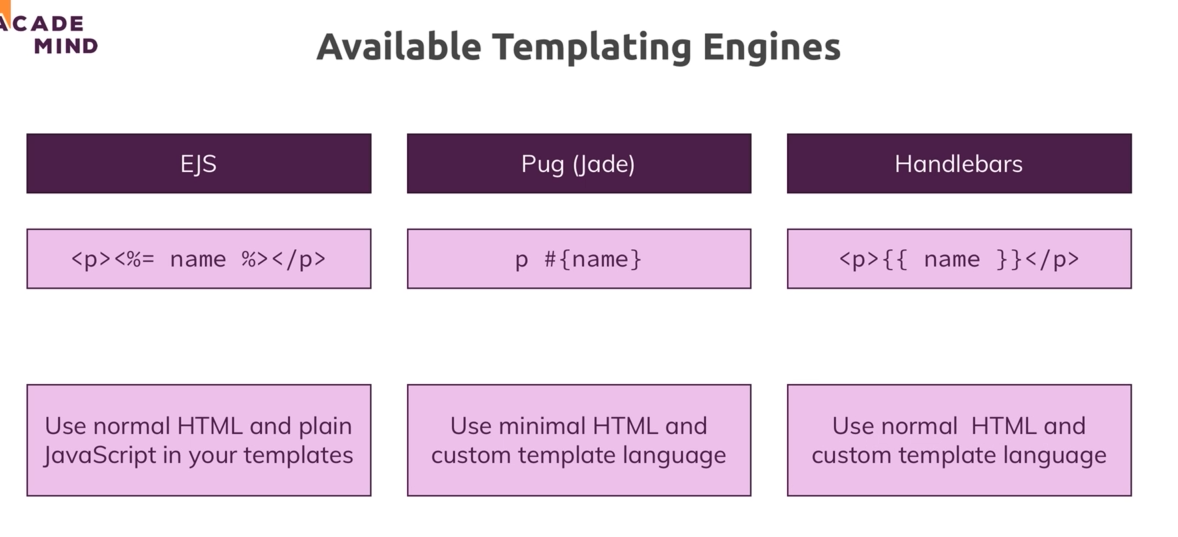
* We can server dynamic pages to users with template engine. 



* **Pug/jade** probably the best scripting template since it has minimal html also have layout system, with javascript logic in the template. I won’t use it for now since don’t want to learn new html.
* **Handlebars** use normal html language, but can’t use javascript logic in the template, only data. Probably to avoid confusion, well I won’t use it either.
* **Ejs** is the one I’m going to use, it uses html syntax, supports JavaScript as well, only drawback is it don’t have layout. But we can overcome this by using partial include

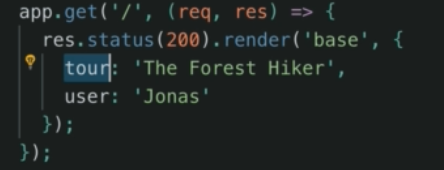
.

**Pug**

* In pug we specify a html tag by writing its name and space after it
* The child parent relationship is done by indentation, each child is one tab indented to the right of the parent.
* For attributes, write the name first, open parenthesis and then right attributes with single quote.
* We could still write regular html as well in pug, one more important thing, for href link paths, the root folder is public (or whichever folder we specify in static) by default.

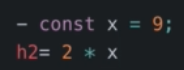


* We can pass data into pug template, from the render method. And use them in template using = syntax. We can even use JavaScript there.



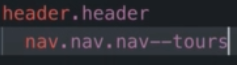
* The above is called buffered code, we also have unbuffered code. It won’t be shown to the users. Here 18 as a h2 will be output. We can even make comments unbuffered using – syntax.



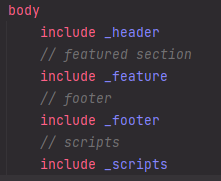
* We can also have variables, kind of similar to template string in JavaScript. But with the # syntax instead of $.



* To write class name, we need to write element then . then class name.

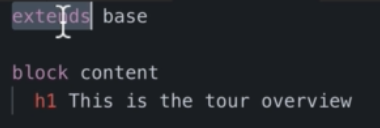


* We can separate parts of pug files into smaller files and include them

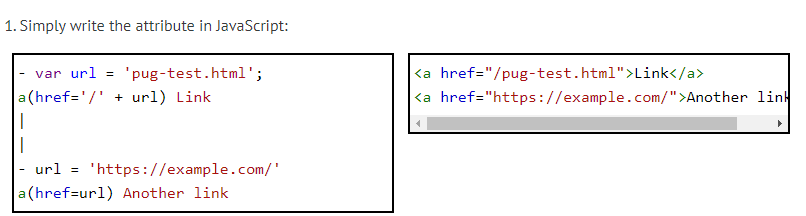


* We can extend a base template with block/extend. We can define a block in a template and another template can extend that and redefine it.





* Mixin is another interesting concept, it lets you create reusable blocks, kind of similar to functions. Check pug documentation for more details.
* For attributes we can simply assign variables to replace them.



* Or we can use ES6 template strings



* To make space between two inline elements we can use |, it will create a space.