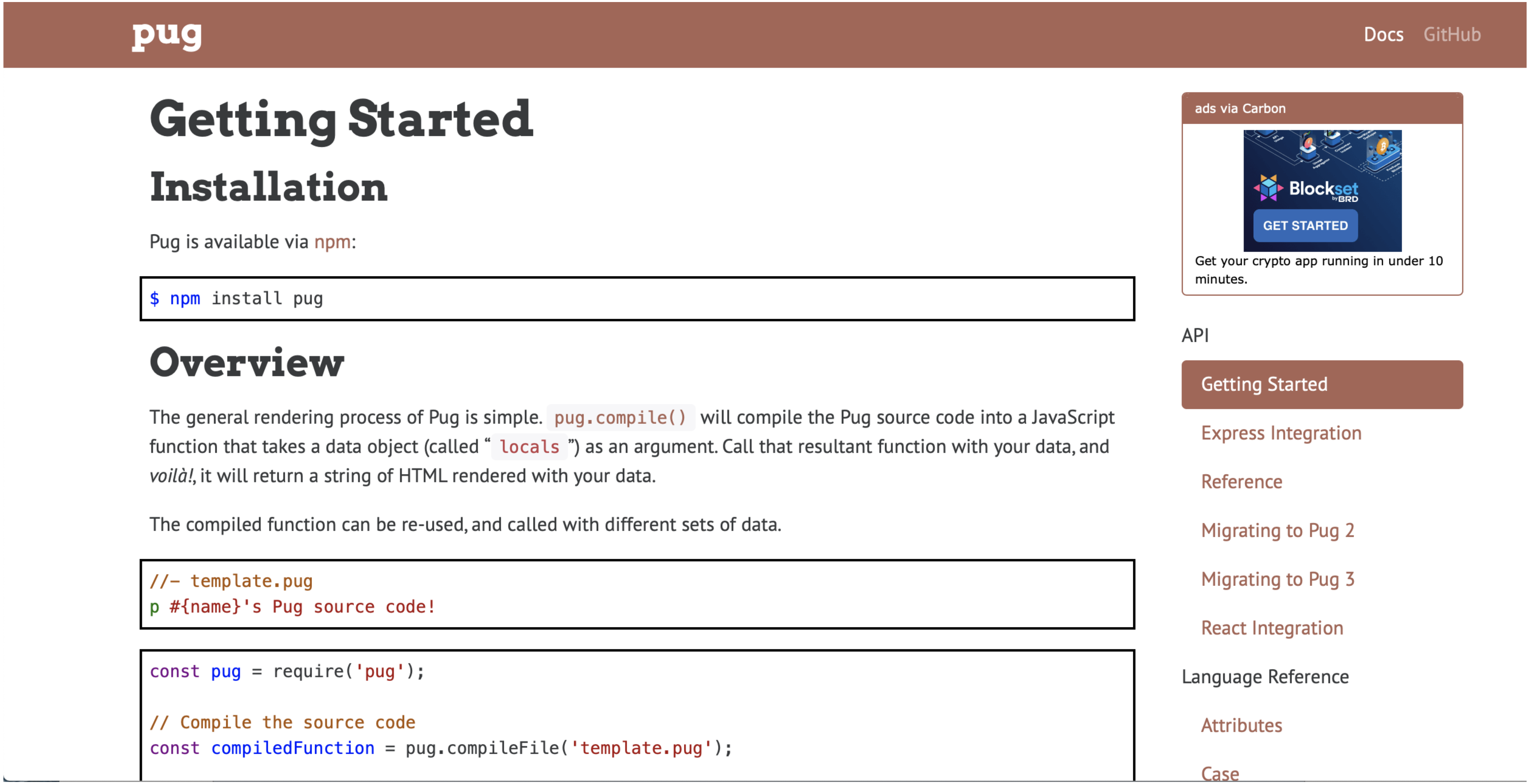
Distribueret Programmering

Lektion 14: Templating

- Templating fjerner HTML tekst fra JavaScript koden
- HTML genereres i stedet ud fra templates og data
- »Pug« er en template engine, men der findes et væld af andre som også kan bruges

https://pugjs.org/api/getting-started.html



Getting Started

Installation

Pug is available via npm:

```
$ npm install pug
```

Overview

The general rendering process of Pug is simple. `pug.compile()` will compile the Pug source code into a JavaScript function that takes a data object (called “ `locals` ”) as an argument. Call that resultant function with your data, and *voilà!*, it will return a string of HTML rendered with your data.

The compiled function can be re-used, and called with different sets of data.

```
//- template.pug
p #{name}'s Pug source code!
```

```
const pug = require('pug');

// Compile the source code
const compiledFunction = pug.compileFile('template.pug');
```

API

Getting Started

Express Integration

Reference

Migrating to Pug 2

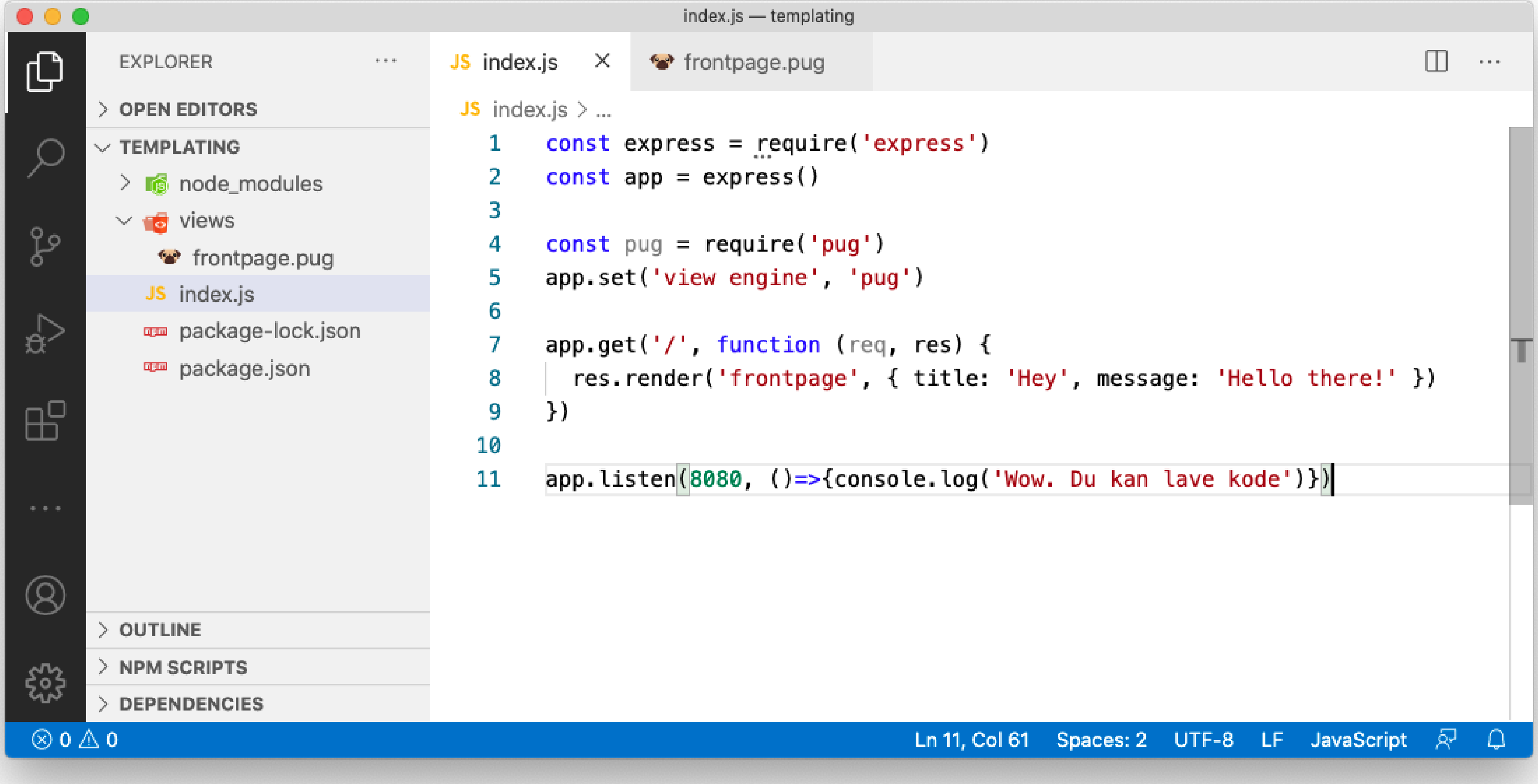
Migrating to Pug 3

React Integration

Language Reference

Attributes

Case



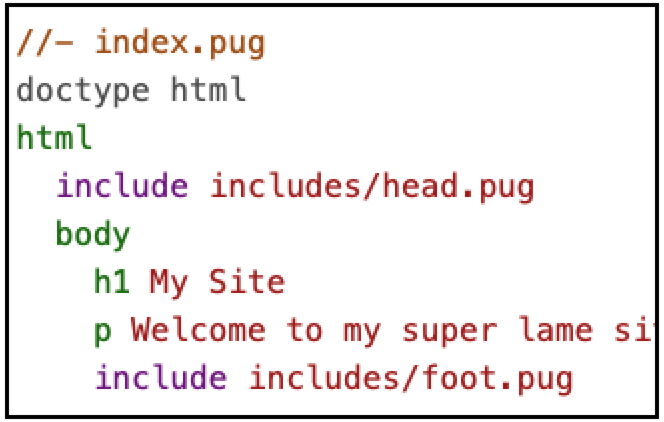
```javascript
const express = require('express')
const app = express()

const pug = require('pug')
app.set('view engine', 'pug')

app.get('/', function (req, res) {
  res.render('frontpage', { title: 'Hey', message: 'Hello there!' })
})

app.listen(8080, ()=>{console.log('Wow. Du kan lave kode')})
```

'Forenklet' HTML markup

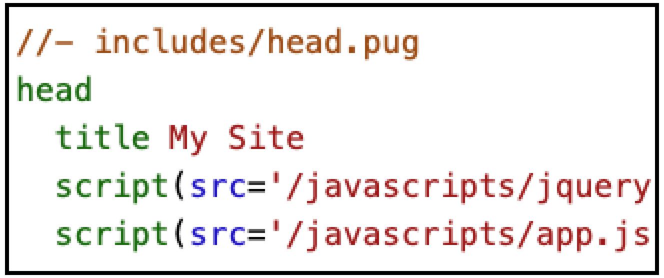
```
html
  head
    title= title
  body
    h1= message
```

```html
<html>
<head>
    <title>Hey</title>
</head>
<body>
    <h1>Hello there!</h1>
</body>
</html>
```

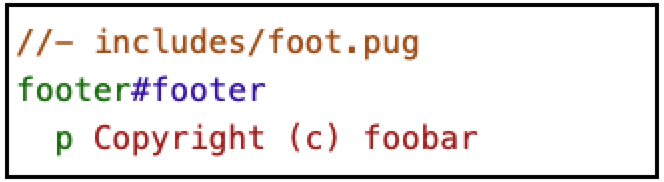
Features i pug »include«



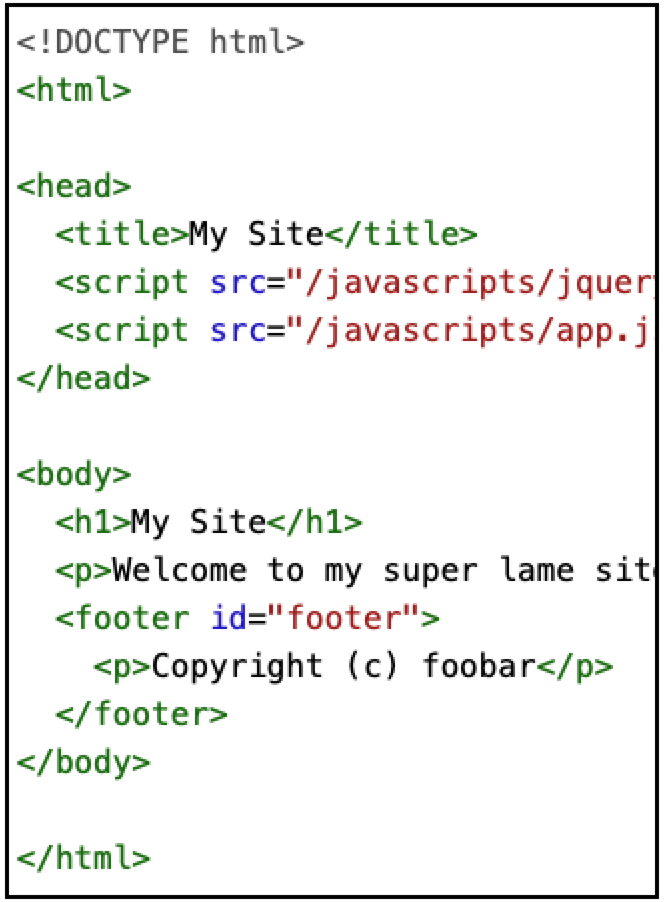
```
//- index.pug
doctype html
html
  include includes/head.pug
  body
    h1 My Site
    p Welcome to my super lame si
    include includes/foot.pug
```



```
//- includes/head.pug
head
  title My Site
  script(src='/javascripts/jquery
  script(src='/javascripts/app.js
```



```
//- includes/foot.pug
footer#footer
  p Copyright (c) foobar
```



```
<!DOCTYPE html>
<html>

<head>
  <title>My Site</title>
  <script src="/javascripts/jquer
  <script src="/javascripts/app.j
</head>

<body>
  <h1>My Site</h1>
  <p>Welcome to my super lame sit
  <footer id="footer">
    <p>Copyright (c) foobar</p>
  </footer>
</body>

</html>
```

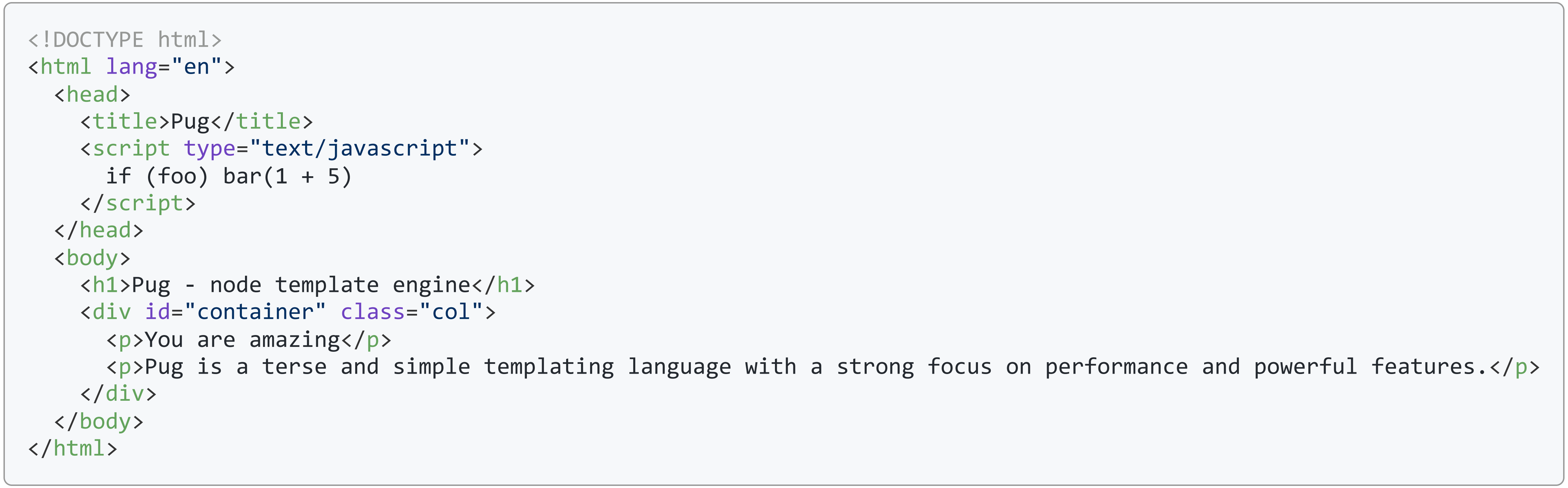
Vis pug-side fra js-fil

```
app.set('view engine', 'pug');
app.set('views', path.join(__dirname, '/views'));
```

```
let valuesForTemplate= {navn:"Ole", alder:10, venner:[{navn:"Ib", alder:12}, {navn:"hans", alder:11}]};
console.log(pug.renderFile(path.join(__dirname, '/views/PugExample.pug'),valuesForTemplate));

app.get('/', (req, res) => res.render('PugExample',valuesForTemplate))
```

Javascript indlejret i pug-kode

```
doctype html
html(lang="en")
  head
    title= pageTitle
    script(type='text/javascript').
      if (foo) bar(1 + 5)
  body
    h1 Pug - node template engine
    #container.col
      if youAreUsingPug
        p You are amazing
      else
        p Get on it!
      p.
        Pug is a terse and simple templating language with a
        strong focus on performance and powerful features.
```



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Pug</title>
    <script type="text/javascript">
      if (foo) bar(1 + 5)
    </script>
  </head>
  <body>
    <h1>Pug - node template engine</h1>
    <div id="container" class="col">
      <p>You are amazing</p>
      <p>Pug is a terse and simple templating language with a strong focus on performance and powerful features.</p>
    </div>
  </body>
</html>
```

```
- var friends = 10
case friends
  when 0
    p you have no friends
  when 1
    p you have a friend
  default
    p you have #{friends} friends
```

```
<p>you have 10 friends</p>
```

send objekt til pug-side (kald det et view)

```
app.get('/', (req, res) => res.render('PugExample',valuesForTemplate))
```

eller send det til console

```
console.log(pug.renderFile(path.join(__dirname, '/views/PugExample.pug'),valuesForTemplate));
```

Opgave 14.1

```
- var user = {description: 'foo bar baz'}
- var authorised = false
#user
  if user.description
    h2.green Description
    p.description= user.description
  else if authorised
    h2.blue Description
    p.description.
      User has no description,
      why not add one...
  else
    h2.red Description
    p.description User has no description
```

```
<div id="user">
  <h2 class="green">Description</h2>
  <p class="description">foo bar baz</p>
</div>
```

```pug
ul
  each val in [1, 2, 3, 4, 5]
    li= val
```

```html
<ul>
  <li>1</li>
  <li>2</li>
  <li>3</li>
  <li>4</li>
  <li>5</li>
</ul>
```

```
ul
  each val, key in {1: 'one', 2: 'two', 3: 'three'}
    li= key + ': ' + val
```

```
<ul>
  <li>1: one</li>
  <li>2: two</li>
  <li>3: three</li>
</ul>
```

https://pugjs.org/language/mixins.html

```
//- Declaration
mixin list
  ul
    li foo
    li bar
    li baz
//- Use
+list
+list
```

```
<ul>
  <li>foo</li>
  <li>bar</li>
  <li>baz</li>
</ul>
<ul>
  <li>foo</li>
  <li>bar</li>
  <li>baz</li>
</ul>
```

```
mixin article(title)
  .article
    .article-wrapper
      h1= title
      if block
        block
      else
        p No content provided

+article('Hello world')

+article('Hello world')
  p This is my
  p Amazing article
```

```html
<div class="article">
  <div class="article-wrapper">
    <h1>Hello world</h1>
    <p>No content provided</p>
  </div>
</div>
<div class="article">
  <div class="article-wrapper">
    <h1>Hello world</h1>
    <p>This is my</p>
    <p>Amazing article</p>
  </div>
</div>
```