

strategy | consulting | digital | **technology** | operations

JADE



AGENDA

- Introduction
- Using Jade as a templating Engine
- JADE Tags, Attributes, Doc type and Nesting
- Applying Styles
- Using forms in Jade
- Mixins
- Using Variables, Conditionals, Arrays and Iteration
- Reuse Jade code



Jade Introduction

- Jade is a terse language for writing HTML templates.
- Produces HTML
- Supports dynamic code
- Supports reusability (DRY)

What is JADE?

Jade is templating language for rapidly producing HTML

It achieves HTML Code Reusability

It increases the speed of producing HTML code

Pre-requisite is basic understanding of HTML and CSS

– Note that, Jade is renamed as Pug recently!

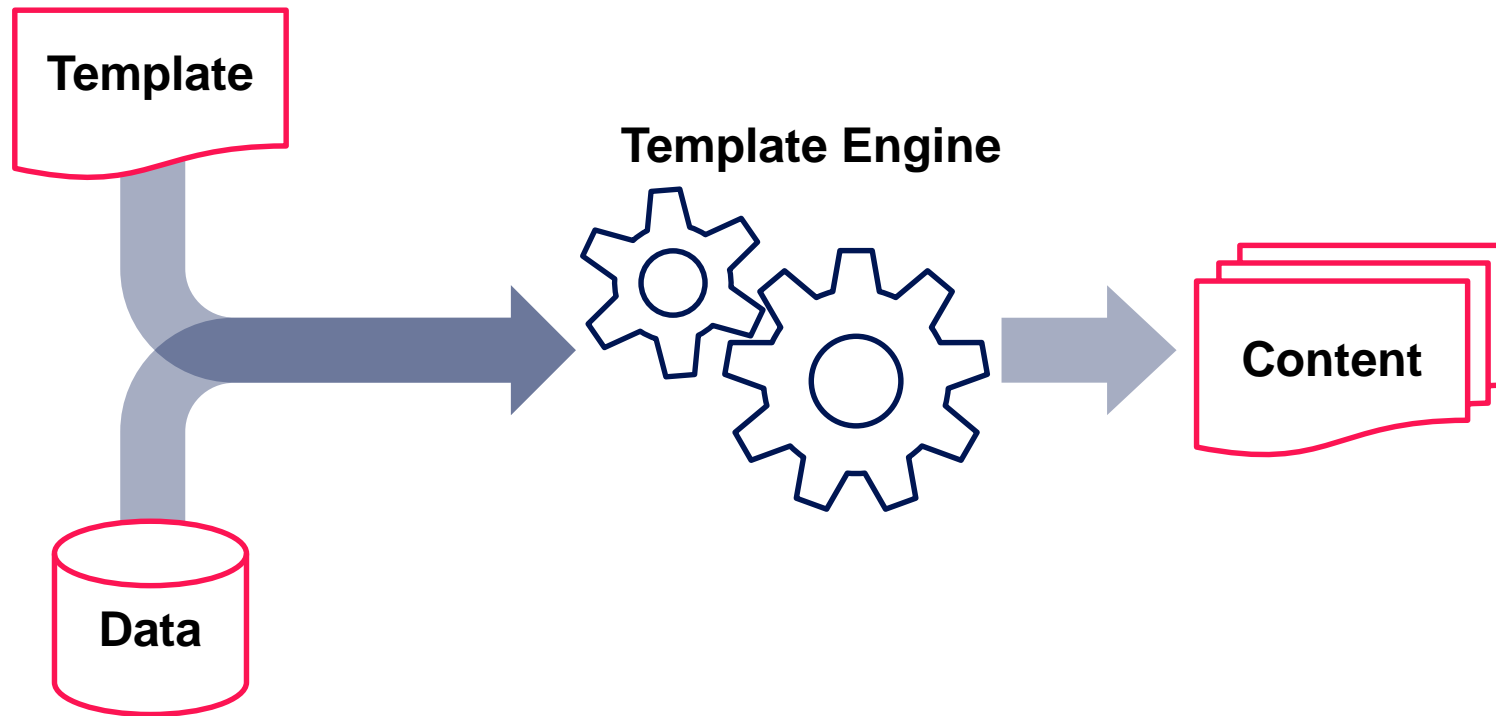
<https://github.com/pugjs/pug/issues/2184>



pug

Template Engines

- A template engine merges content in a controlled way
- Typically it will combine boilerplate text with dynamic data



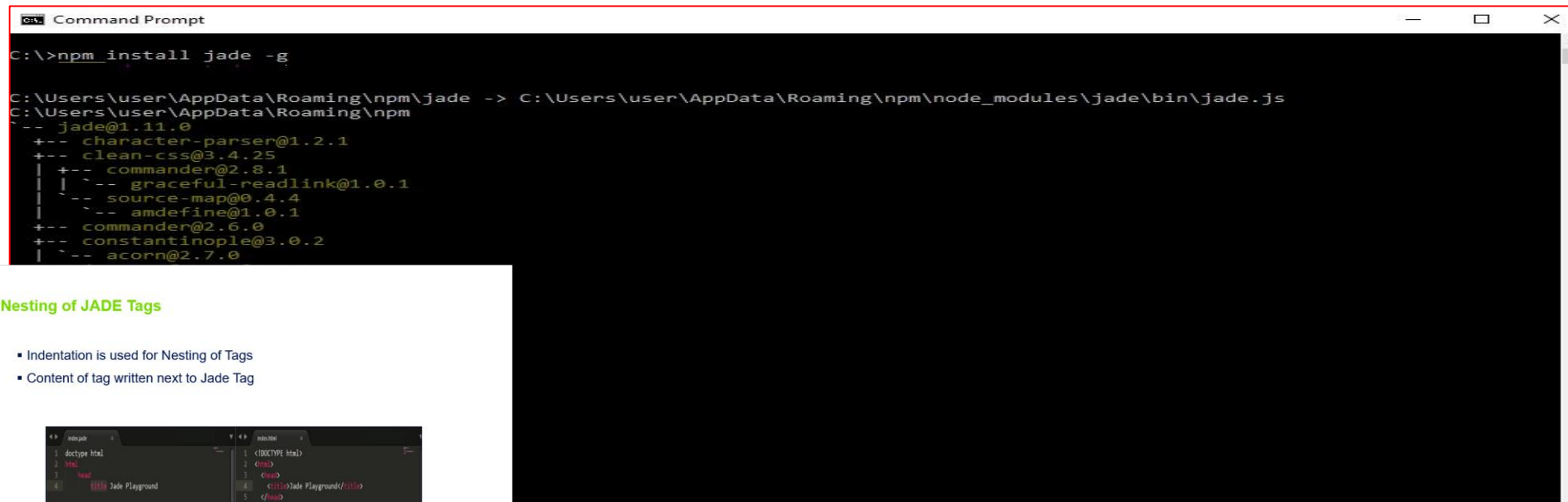
Using Jade

- HTML brackets are not required.
- Nesting is important.
- Indents do matter.
- Be consistent with your indentation key (like tab).
- Benefit: you can create markup much quicker.



Installation of JADE

- Check whether Node JS is already installed
- Install JADE Globally
- *npm install jade -g*

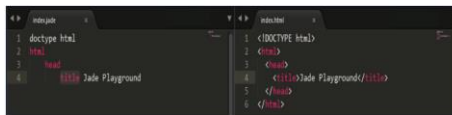


```
C:\>npm install jade -g

C:\Users\user\AppData\Roaming\npm\jade -> C:\Users\user\AppData\Roaming\npm\node_modules\jade\bin\jade.js
C:\Users\user\AppData\Roaming\npm
-- jade@1.11.0
+-- character-parser@1.2.1
+-- clean-css@3.4.25
+-- commander@2.8.1
|   |-- graceful-readlink@1.0.1
|   |-- source-map@0.4.4
|   |-- amdefine@1.0.1
+-- commander@2.6.0
+-- constantinople@3.0.2
|   |-- acorn@2.7.0
```

Nesting of JADE Tags

- Indentation is used for Nesting of Tags
- Content of tag written next to Jade Tag



Develop first JADE Project

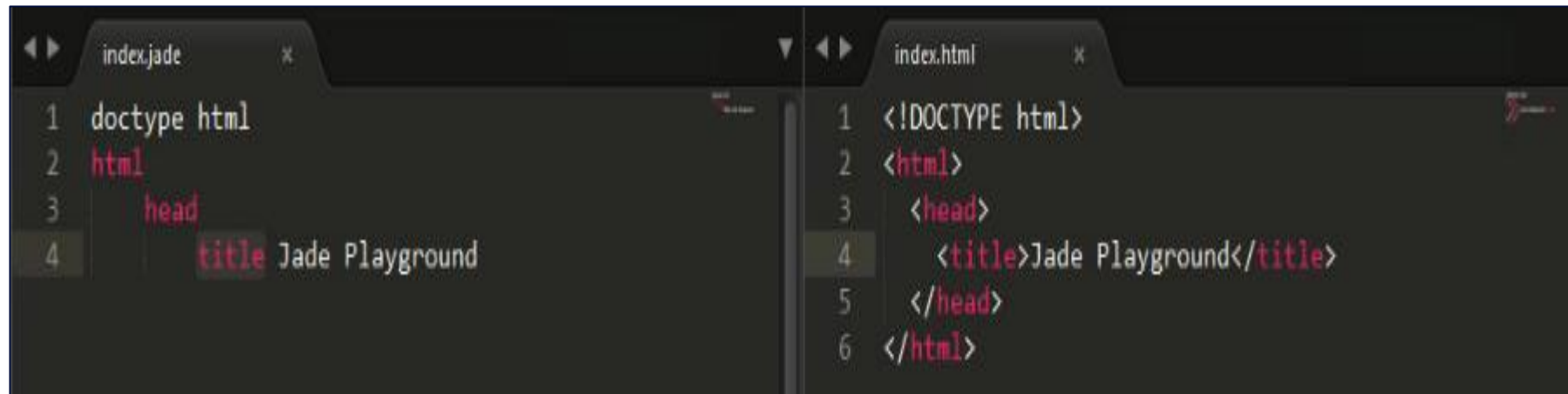
- Put below code in index.jade file

```
doctype
html
  head
  body
```

- Follow indentation properly
- To generate the html run
jade index.jade (to beautify use the command **jade --out . index.jade --pretty**)
- Observe the contents of generated html file 'index.html'

Nesting of JADE Tags

- Indentation is used for Nesting of Tags
- Content of tag written next to Jade Tag

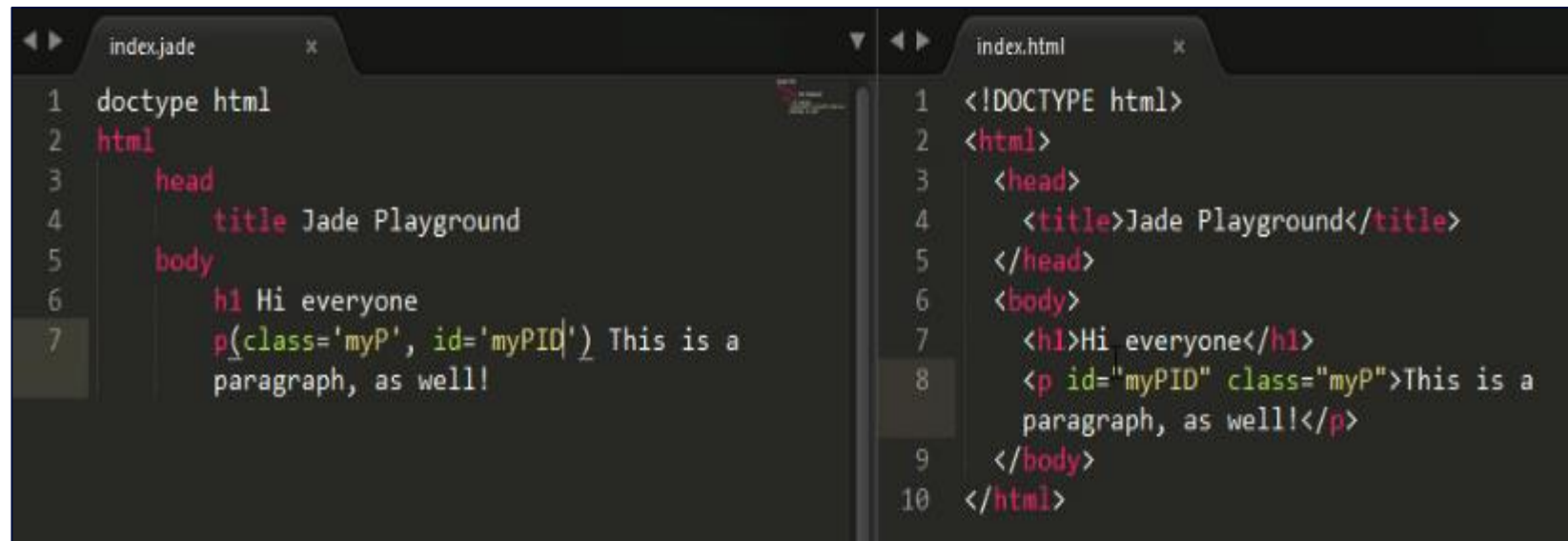


```
index.jade
1 doctype html
2 html
3   head
4     title Jade Playground

index.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Jade Playground</title>
5   </head>
6 </html>
```

JADE Tag Attributes

- One or more attribute for JADE Tags
- Class attribute and id attribute can be added for paragraph tag as below



The image shows a side-by-side comparison of JADE and HTML code in a code editor. The left pane, titled 'index.jade', contains the following code:

```
1 doctype html
2 html
3   head
4     title Jade Playground
5   body
6     h1 Hi everyone
7     p(class='myP', id='myPID') This is a
      paragraph, as well!
```

The right pane, titled 'index.html', shows the equivalent HTML output:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Jade Playground</title>
5   </head>
6   <body>
7     <h1>Hi everyone</h1>
8     <p id="myPID" class="myP">This is a
      paragraph, as well!</p>
9   </body>
10 </html>
```

JADE Tag Attributes

- Putting multiple class names and id in Div Tag

Jade Code

```
div.myDiv.my2nddiv#mydivID
```

HTML Code

```
<div id="mydivID" class="myDiv my2nddiv"><  
/div>
```

Styles in JADE

- Apply inline CSS to JADE
 - Styles for Div Tag

Jade Code

```
doctype html
html
  head
    title Jade Playground
  body
    h1 Hi everyone
    p(class='myP', id='myPID') This is a
      paragraph, as well!
    div(style={color: 'gray', background:
      'black'})
```

HTML Code

```
<!DOCTYPE html>
<html>
  <head>
    <title>Jade Playground</title>
  </head>
  <body>
    <h1>Hi everyone</h1>
    <p id="myPID" class="myP">This is a
      paragraph, as well!</p>
    <div style="color:gray;background:black"><
      /div>
  </body>
</html>
```

JADE Forms

- Displaying Forms with different components

Jade Code

```
doctype html
html
  head
    title Jade Playground
  body
    h1 Hi everyone
    p(class='myP', id='myPID') This is a
      paragraph, as well!

    div(style={color: 'gray', background:
      'black'})

    form(action='', method='POST')
      label(for='firstName') First Name:
      input(type='text', name='firstName
        ', id='firstName')
```

HTML Code

```
<!DOCTYPE html>
<html>
  <head>
    <title>Jade Playground</title>
  </head>
  <body>
    <h1>Hi everyone</h1>
    <p id="myPID" class="myP">This is a
      paragraph, as well!</p>
    <div style="color:gray;background:black"><
      /div>
    <form action="" method="POST">
      <label for="firstName">First Name:</
        label>
      <input type="text" name="firstName" id="
        firstName">
    </form>
  </body>
</html>
```

JADE Forms

- Lists in Form

Jade Code

```
form(action='', method='POST')
  label(for='firstName') First Name:
  input(type='text', name='firstName', id='firstName')

  label(for='lastName') Last Name:
  input(type='text', name='lastName', id='lastName')

  label(for='myList') Occupation:
  select(id='myList', name='myList')
    option(value='Designer') Designer
    option(value='Coder') Coder
```

HTML Code

```
<form action="" method="POST">
  <label for="firstName">First Name:</label>
  <input type="text" name="firstName" id="firstName">
  <label for="lastName">Last Name:</label>
  <input type="text" name="lastName" id="lastName">
  <label for="myList">Occupation:</label>
  <select id="myList" name="myList">
    <option value="Designer">Designer
    </option>
    <option value="Coder">Coder</option>
  </select>
</form>
```


JADE Forms

- Buttons on Form

Jade Code

```
form(action='', method='POST')
  label(for='firstName') First Name:
  input(type='text', name='firstName', id='firstName')

  label(for='lastName') Last Name:
  input(type='text', name='lastName', id='lastName')

  label(for='myList') Occupation:
  select(id='myList', name='myList')
    option(value='Designer') Designer
    option(value='Coder') Coder

  input(type='button', value='Submit')
```

HTML Code

```
<form action="" method="POST">
  <label for="firstName">First Name:</label>
  <input type="text" name="firstName" id="firstName">
  <label for="lastName">Last Name:</label>
  <input type="text" name="lastName" id="lastName">
  <label for="myList">Occupation:</label>
  <select id="myList" name="myList">
    <option value="Designer">Designer</option>
    <option value="Coder">Coder</option>
  </select>
  <input type="button" value="Submit">
</form>
```

JADE Forms

- Output of Index.html

Hi everyone

This is a paragraph, as well!

First Name: Last Name: Occupation:

Designer
Designer
Coder

JADE Mixins

- Mixin is used to remove redundant code
- Paragraph Mixin is replaced by definition

Jade Code

```
mixin paragraph
  p Hello

+paragraph
```

HTML Code

```
<p>Hello</p>
```

JADE Mixins

- Parameters to Mixin

Jade Code

```
mixin paragraph2(myContent)  
  p= myContent  
  
+paragraph2('Hey, how is it  
going?')
```

HTML Code

```
<p>Hey, how is it going?</p>
```

JADE Mixins

- Sending Attribute Value to Mixin

Jade Code

```
mixin paragraph2(myContent, id)
  p(id=id)= myContent

+paragraph2('Hey, how is it
going?', 'myParaID')
```

HTML Code

```
<p id="myParaID">Hey, how is it going?</p>
```

JADE Mixins

- Block of Code in Mixin

Jade Code

```
mixin formText(textID, textValue)
  label(for= textID)= textValue
  input(type='text', name= textID, id= textID)

+formText('firstName', 'First Name:')

+formText('lastName', 'Last Name:')
```

HTML Code

```
<form action="" method="POST">
  <label for="firstName">First Name:</label>
  <input type="text" name="firstName" id="firstName">
  <label for="lastName">Last Name:</label>
  <input type="text" name="lastName" id="lastName">
```

JADE Mixins

- Mixin is used to remove redundant code
- Paragraph Mixin is replaced by definition

Jade Code

```
mixin paragraph
  p Hello

+paragraph
```

HTML Code

```
<p>Hello</p>
```

JADE Mixins

- Parameters to Mixin

Jade Code

```
mixin paragraph2(myContent)  
  p= myContent  
  
+paragraph2('Hey, how is it  
going?')
```

HTML Code

```
<p>Hey, how is it going?</p>
```

JADE Mixins

- Sending Attribute Value to Mixin

Jade Code

```
mixin paragraph2(myContent, id)
  p(id=id)= myContent

+paragraph2('Hey, how is it
going?', 'myParaID')
```

HTML Code

```
<p id="myParaID">Hey, how is it going?</p>
```

JADE Mixins

- Block of Code in Mixin

Jade Code

```
mixin formText(textID, textValue)
  label(for= textID)= textValue
  input(type='text', name= textID, id= textID)

+formText('firstName', 'First Name:')

+formText('lastName', 'Last Name:')
```

HTML Code

```
<form action="" method="POST">
  <label for="firstName">First Name:</label>
  <input type="text" name="firstName" id="firstName">
  <label for="lastName">Last Name:</label>
  <input type="text" name="lastName" id="lastName">
```


Extends in JADE

- Extending layouts
- Achieves Reusability
- We will create layout.jade file in layouts folder in the project and reuse it in current jade file

Reusing already
written jade code

Jade Code

```
extends ../layouts/layout.jade

doctype html
html
  head
    title Jade Playground
  body
    - var myTitle2 = 'Hey everyone!'
    - var mySkills = ['PHP', 'MySQL', 'SASS']
```

JADE Extends

- Create layouts folder and layout.jade file in that

layout.jade

```
doctype html
html
  head
    block headContent
  body
    block bodyContent
```

index.jade

```
extends ../layouts/layout

block headContent
  title Jade Playground

block bodyContent

  h1 My title
  p(class='myP', id='myPID') This is
    a paragraph, as well!

  form(action='', method='POST')
```

Using headContent
defined in layout.jade

Using bodyContent
defined in
layout.jade

JADE Extends

- Can you find out which code is came from
- Index.jade and which from layout.jade?

Index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Jade Playground</title>
  </head>
  <body>
    <h1>My title</h1>
    <p id="myPID" class="myP">This is a
    paragraph, as well!</p>
    <form action="" method="POST">
      <label for="firstName">First Name:</
      label>
      <input type="text" name="firstName"
      id="firstName">
      <label for="lastName">Last Name:</
      label>
      <input type="text" name="lastName"
      id="lastName">
      <label for="myList">Occupation:</label>
      <select id="myList" name="myList">
        <option value="Designer">Designer</
        option>
```

JADE Includes

- Reusing Mixing
- How it is different than Extends?

Mixins are defined in mixins.jade file and using it here

index.jade

```
extends ../layouts/layout.jade
include ../includes/mixins.jade

block headContent
  title Jade Playground
```

mixins.jade

```
mixin formText(textID, textValue)
  label(for= textID)=
    textValue
  input(type='text', name=
    textID, id= textID)
```

Lesson Summary

Here are some key points to take away from this lesson:

- Understood templating in HTML
- Install and develop first Project in JADE
- How to use different building blocks like JADE Tags, Attributes, Doc type and Nesting
- How to do Formatting in JADE with Styles
- How to enter data using JADE Forms
- How to develop JADE Mixins
- Using Variables, Conditionals, Arrays and Iteration
- How to reuse JADE code with extends and includes capability



BREAK : 15 MINUTES



LUNCH BREAK 45 MINUTES



THANK YOU

