



KINGSLAND  
UNIVERSITY

Templating

handlebars



Creating UI Elements



# Table of Contents

- ✔ Templating Concepts
- ✔ Simple Templating
- ✔ Templating Engines
- ✔ Handlebars Overview



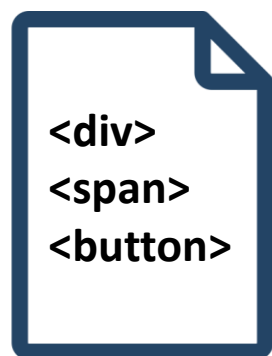
# Definition and Uses

**Templating Concepts**

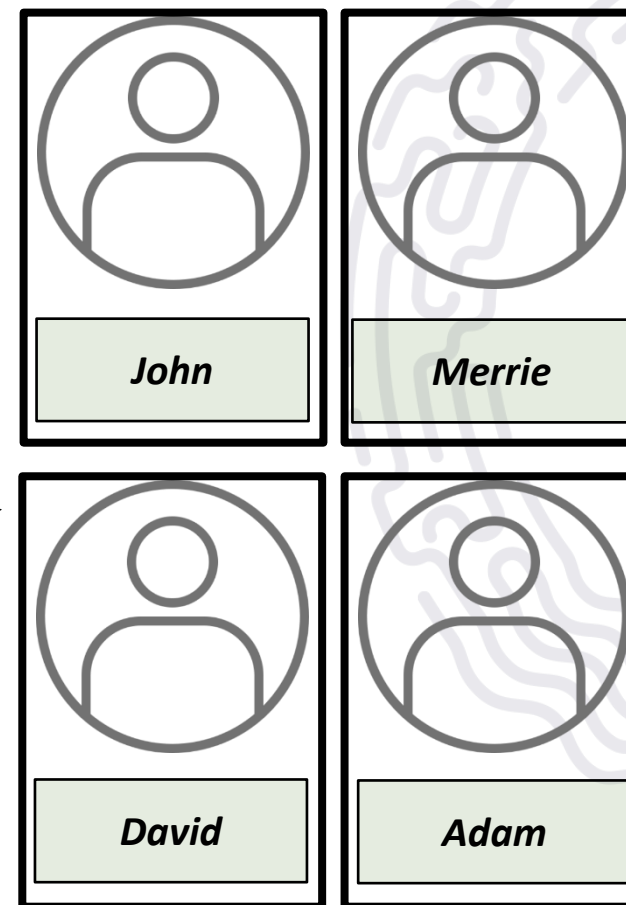
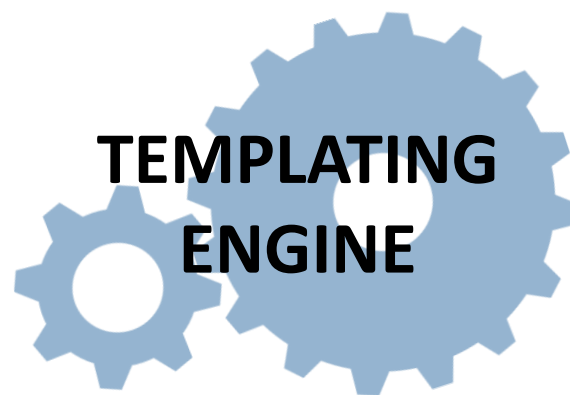
# What is Templating?

- ✓ Templates allow similar content to be **replicated** in a web page, **without repeating** the corresponding markup everywhere

## HTML



## Dynamic Content





# Templating Concepts

- ✓ **Templates** are a method of **separating HTML** structure from the **content** contained within
- ✓ Templating systems generally introduce some **new syntax** but are usually very simple to work with
- ✓ Typically token replacement is used to indicate part, which must be replaced - (**{{ ... }}**, **<%...%>** etc.)



# Templating Concepts

- ✓ Templates should be as **simple** as possible
  - ✓ Do not write business logic in the templates
- ✓ **Follow the principles of functional programming**
  - ✓ Templates are basically **pure functions**



# Templating Concepts

- ✓ The **static parts** of a webpage are stored as **templates**
- ✓ The **dynamic content** is kept separately (e.g. in a **database**)
- ✓ A **templating engine** combines the two
- ✓ Benefits:
  - ✓ **Productivity** - avoid writing the same markup over and over
  - ✓ **Save bandwidth** - send the HTML once, fill in any content
  - ✓ **Composability** - a single element can be used on multiple pages



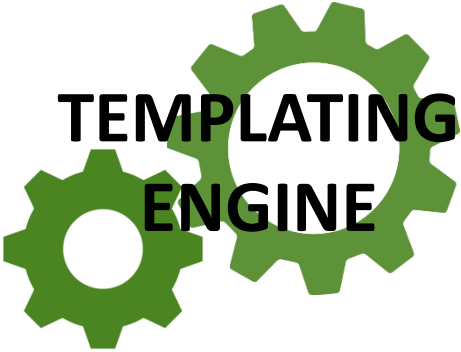
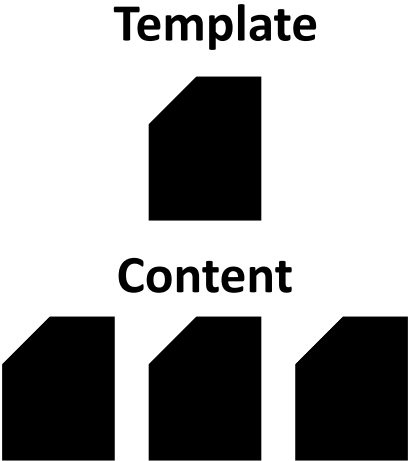


# When Should We Use JS Templating?

- ✓ As soon as we find ourselves including **HTML inside JavaScript strings**
- ✓ **Separation of concerns** is of utmost importance when building a maintainable codebase



# Display Articles in Blog



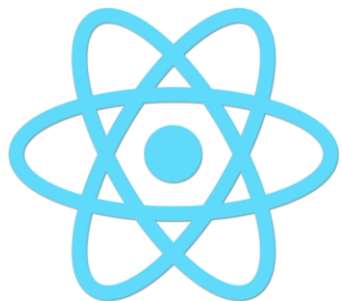


# Overview of Popular JS Libraries

Templating Engines



# Templating Engines



React



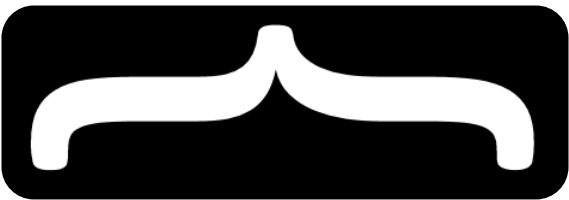
Vue



Angular



Web Components

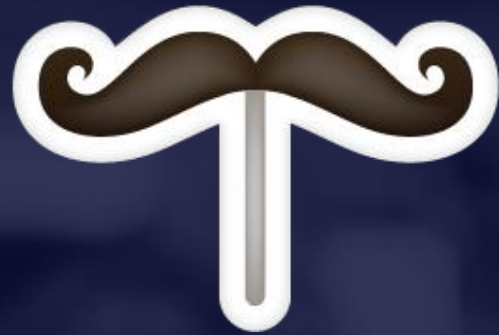


Mustache



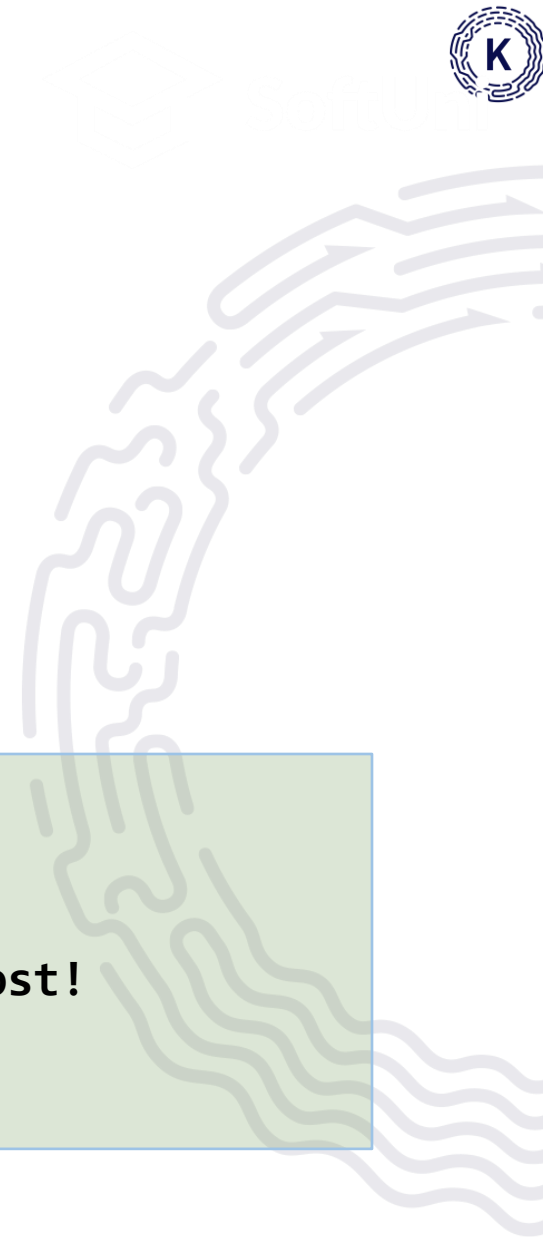
Handlebars

# handlebars



## Syntax and Examples

Templating with Handlebars



# Overview

- ✓ Based on the Mustache specification
- ✓ Adds helper functions and nested context paths
- ✓ Uses double curly brace notation

```
<div class="entry">  
  <h1>{{ title }}</h1>  
  <div class="body">  
    {{ body }}  
  </div>  
</div>
```

```
<div class="entry">  
  <h1>My New Post</h1>  
  <div class="body">  
    This is my first post!  
  </div>  
</div>
```



# Handlebars Installation and Using

- ✔ Download Handlebars using the terminal
- ✔ Or download from [handlebarsjs.com](https://handlebarsjs.com)
- ✔ Browser builds will be located in
- ✔ Use handlebars from an online CDN
- ✔ Link it with a script tag

```
<script  
  src="node_modules/handlebars/dist/handlebars.js"  
</script>
```



# Expressions

- ✓ You can place your templates in a script element
  - ✓ Use type **text/x-handlebars-template**
  - ✓ Give the element an ID for easier use
- ✓ Anything inside double curly braces **{{ }}** will be evaluated







# Compilation and Execution

☑ Once loaded, a template must be compiled

```
let src = document.getElementById("contact-template").innerHTML;  
let template = Handlebars.compile(src);
```

☑ Compiled templates are functions, that can be executed with whatever variables we need

```
let context = {  
  name: 'Ivan Ivanov',  
  phone: '0888 123 456',  
  email: 'i.ivanov@gmail.com'  
};  
let html = template(context);
```



# Hello Handlebars



```
<!DOCTYPE html>
<head>
  <meta charset="UTF-8">
  <title>Hello Handlebars</title>
  <!-- Include Handlebars distribution -->
</head>
<body>
<div id="app"></div>
<script>
  let template = Handlebars.compile('<h1>Hello {{name}}</h1>');
  let container = document.getElementById('app');
  container.innerHTML = template({ name: 'Handlebars' });
</script>
</body>
</html>
```



# Simple Identifiers

```
<script id="contact-template"
      type="text/x-handlebars-template">
  <article>
    <div class="title">
      {{name}}
      <button>&#8505;</button>
    </div>
    <div class="info">
      <span>&phone; {{phone}}</span>
      <span>&#9993; {{email}}</span>
    </div>
  </article>
</script>
```

# For-Loops

- ✓ A template can be repeated for every entry in an array

```
<ul id="contacts">
  {{#each contacts}}
    <li>{{name}}: {{email}}</li>
  {{/each}}
</ul>
```

The expression inside  
the loop uses each  
entry as context



# Conditional Statements

```
{{#if sunny}}  
  The sky is clear  
{{else}}  
  The sky is overcast  
{{/if}}
```

Variable to check  
for truthiness

```
<ul id="contacts">  
  {{#each contacts}}  
    <li>{{name}}: {{email}}</li>  
  {{else}}  
    <i>(empty)</i>  
  {{/each}}  
</ul>
```

Will be shown if  
the array is empty

# Partials

- ✓ Partials are templates that can be inserted into other templates

```
let src = document.getElementById("contact-template").innerHTML;  
Handlebars.registerPartial('contact', src);
```

Partial name

Template as string

```
<div id="contacts">  
  {{#each contacts}}  
    {{> contact}}  
  {{else}}  
    <i>(empty)<i>  
  {{/each}}  
</div>
```

Partials are  
globally accessible



# HTML Escaping

## ✔ Using the "triple-stash"

```
title: "All about <p> Tags"  
body: "<p>This is a post about &lt;p> tags</p>"
```

```
<div class="entry">  
  <h1>All About &lt;p> Tags</h1>  
  <div class="body">  
    <p>  
      This is a post about&lt;p> tags  
    </p>  
  </div>  
</div>
```

The background of the slide is a dark blue, blurred image of a classroom. In the foreground, the backs of several students' heads are visible as they sit at desks. In the background, a whiteboard is mounted on a wall, and other students are seated further back in the room.

# Live Exercise





# Summary

- **Templates** speed up and simplify the development process
- **Handlebars** offers effective **templates** and simple helper functions

```
<div class="entry">  
  <h1>{{ title }}</h1>  
  <div class="body">  
    {{ body }}  
  </div>  
</div>
```





# Questions?





# License

- ✔ This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- ✔ Unauthorized copy, reproduction or use is illegal
- ✔ © Kingsland University – <https://kingslanduniversity.com>





THANK YOU

