

Templating

Creating UI elements with Handlebars

handlebars



SoftUni Team
Technical Trainers



Software
University



SoftUni
Foundation



Software University

<http://softuni.bg>

Table of Contents

1. Templating Concepts
2. Simple Templating
3. Templating Engines
4. Handlebars Overview
5. Examples



Have a Question?

sli.do

#JSCORE

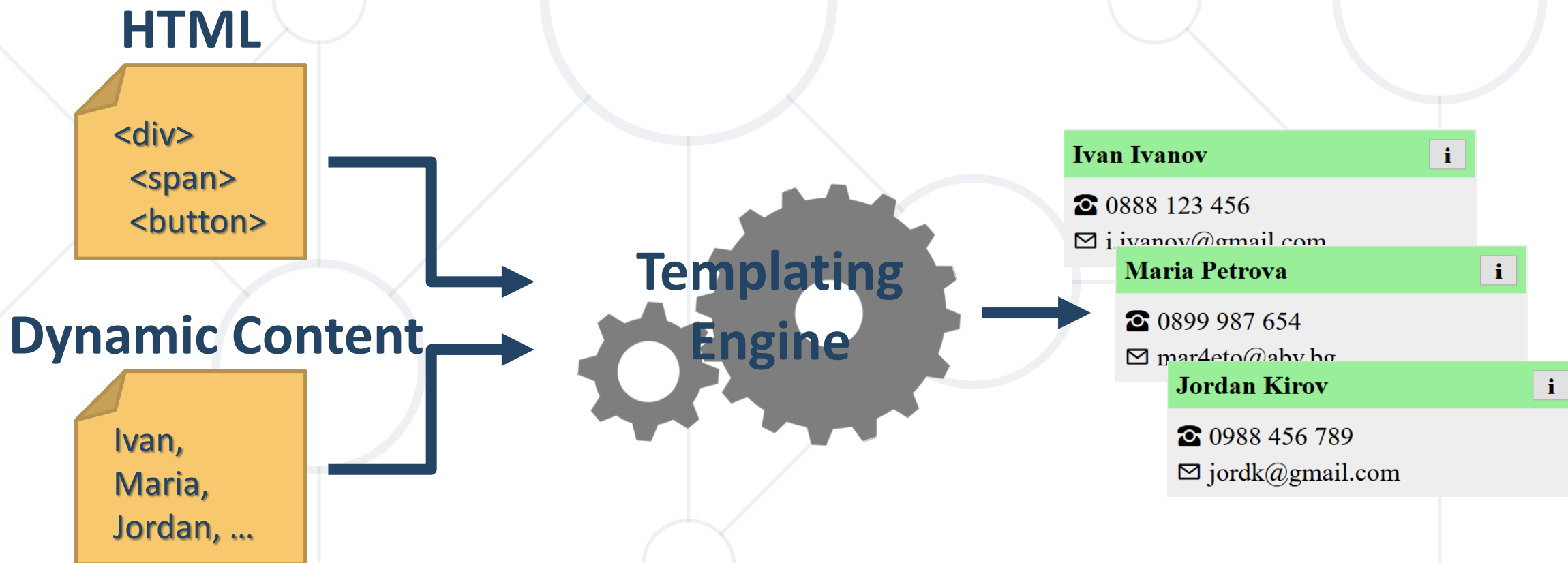


Templating Concepts

Definition and Uses

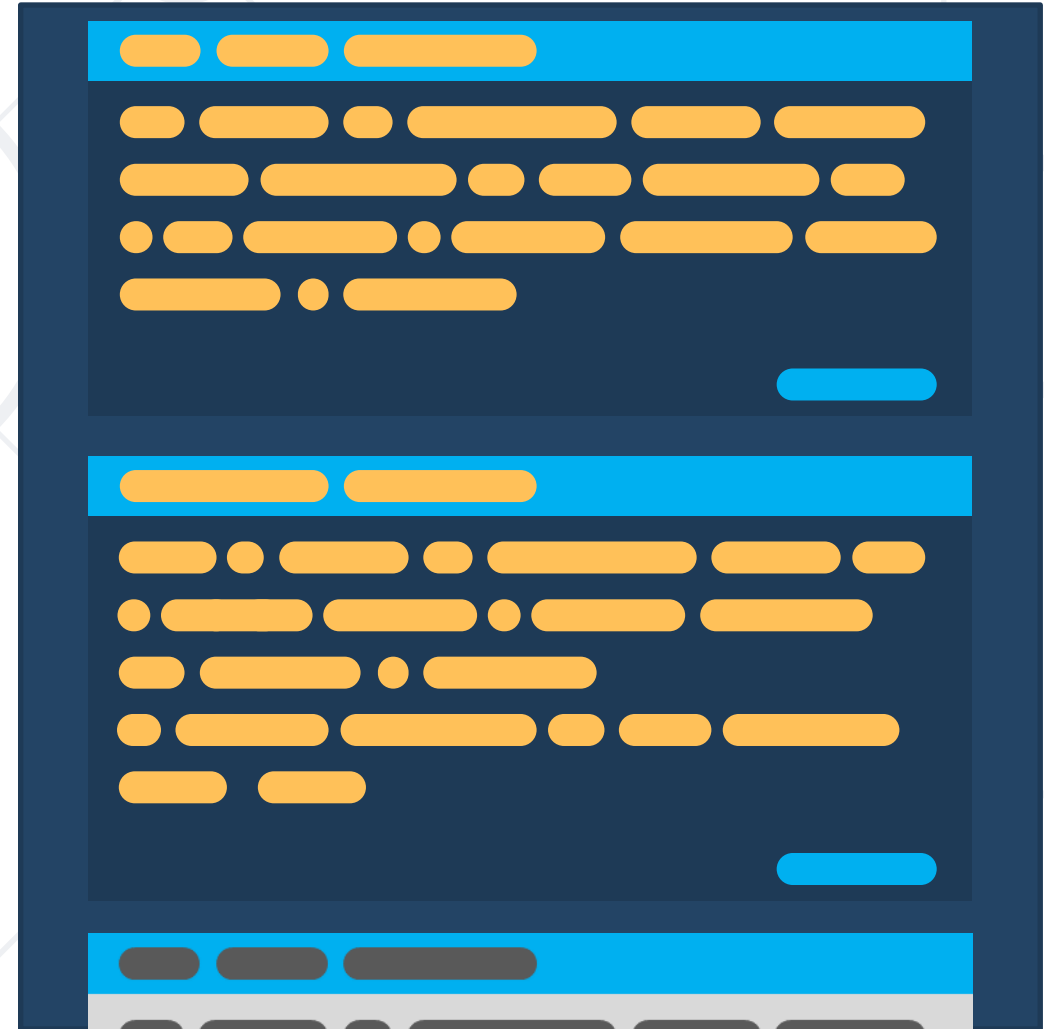
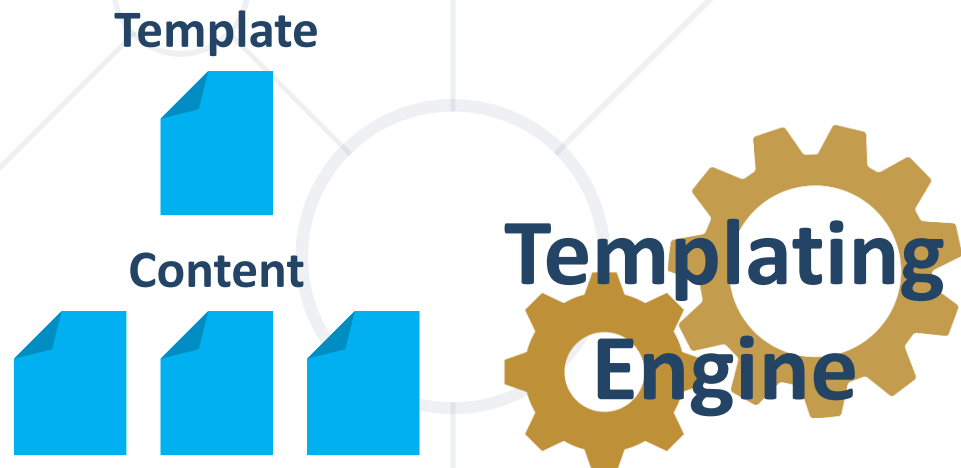
What is Templating?

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



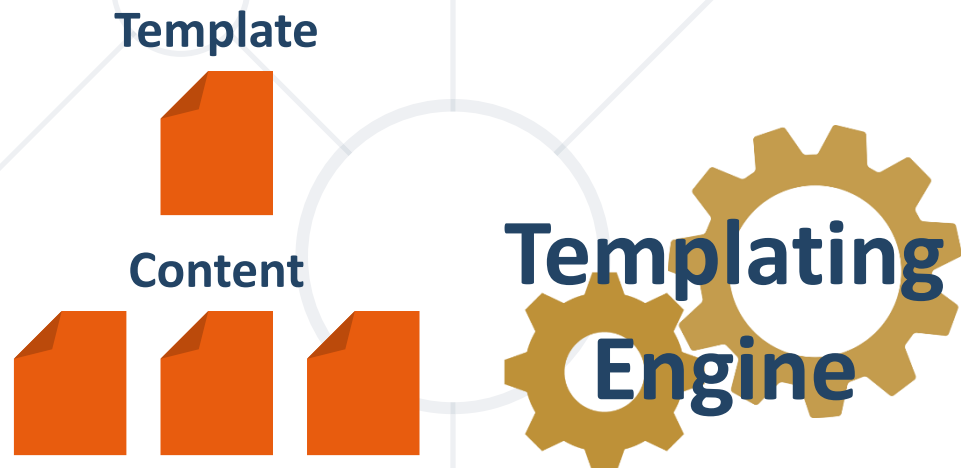
- The **static parts** of a webpage are stored as **templates**
- The **dynamic content** is kept separately (e.g. in a **database**)
- A **templating engine** combines the two
- Benefits:
 - **Productivity** – avoid writing the same markup over and over
 - **Save bandwidth** – send the HTML once, fill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

- Display articles in a blog



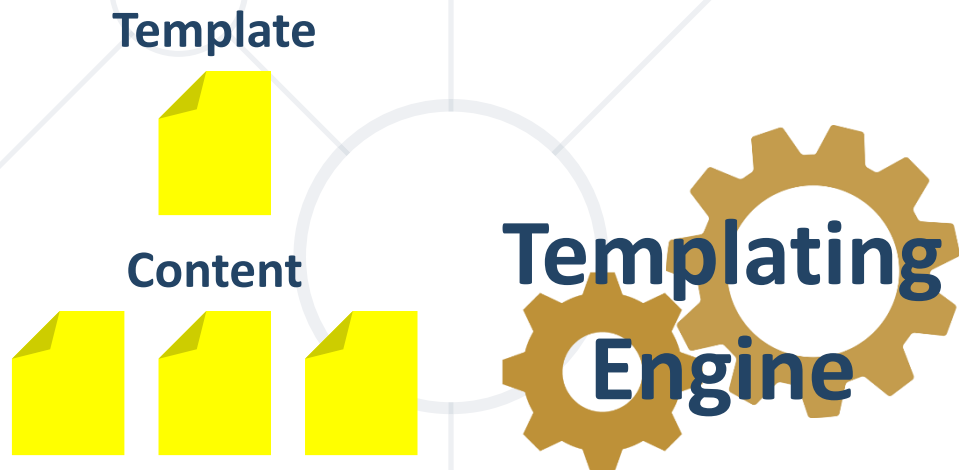
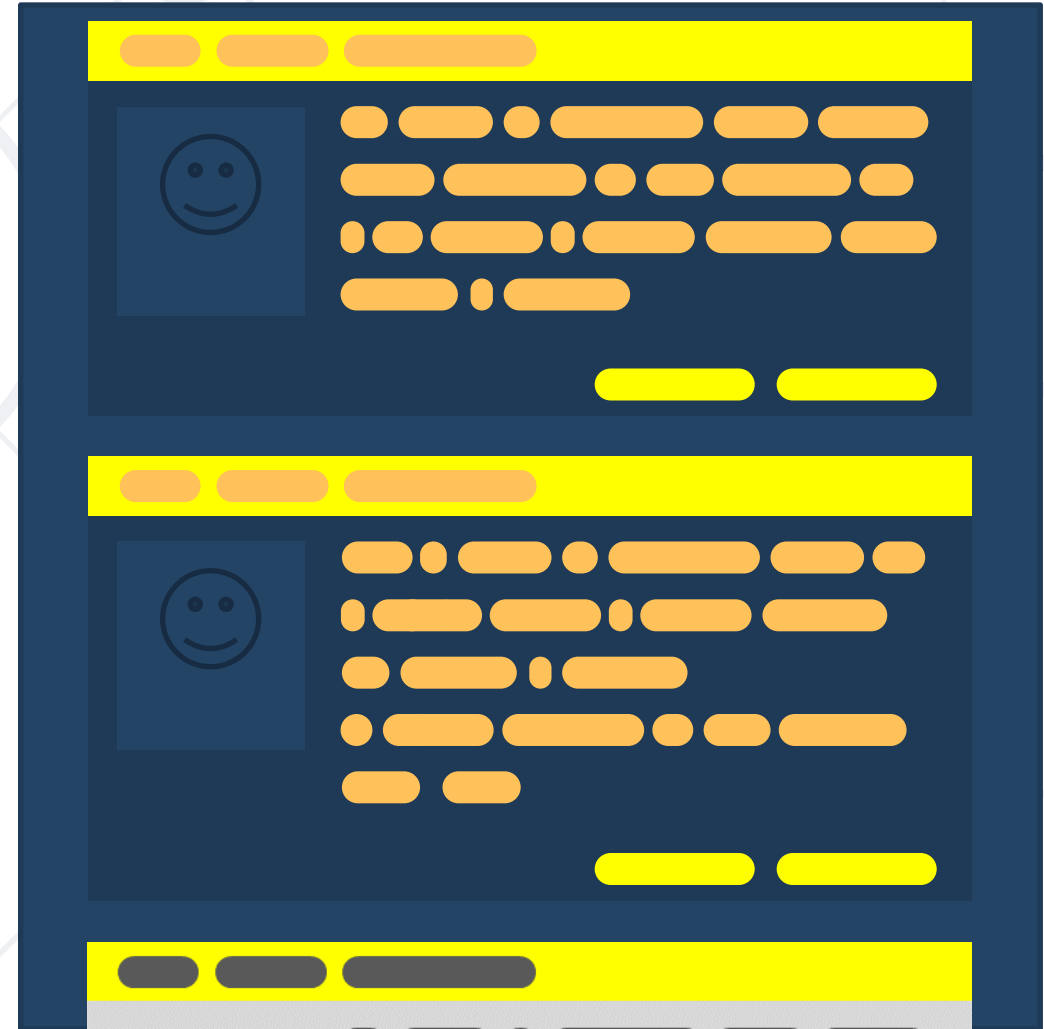
Examples

- Display articles in a blog
- Display a gallery of photos



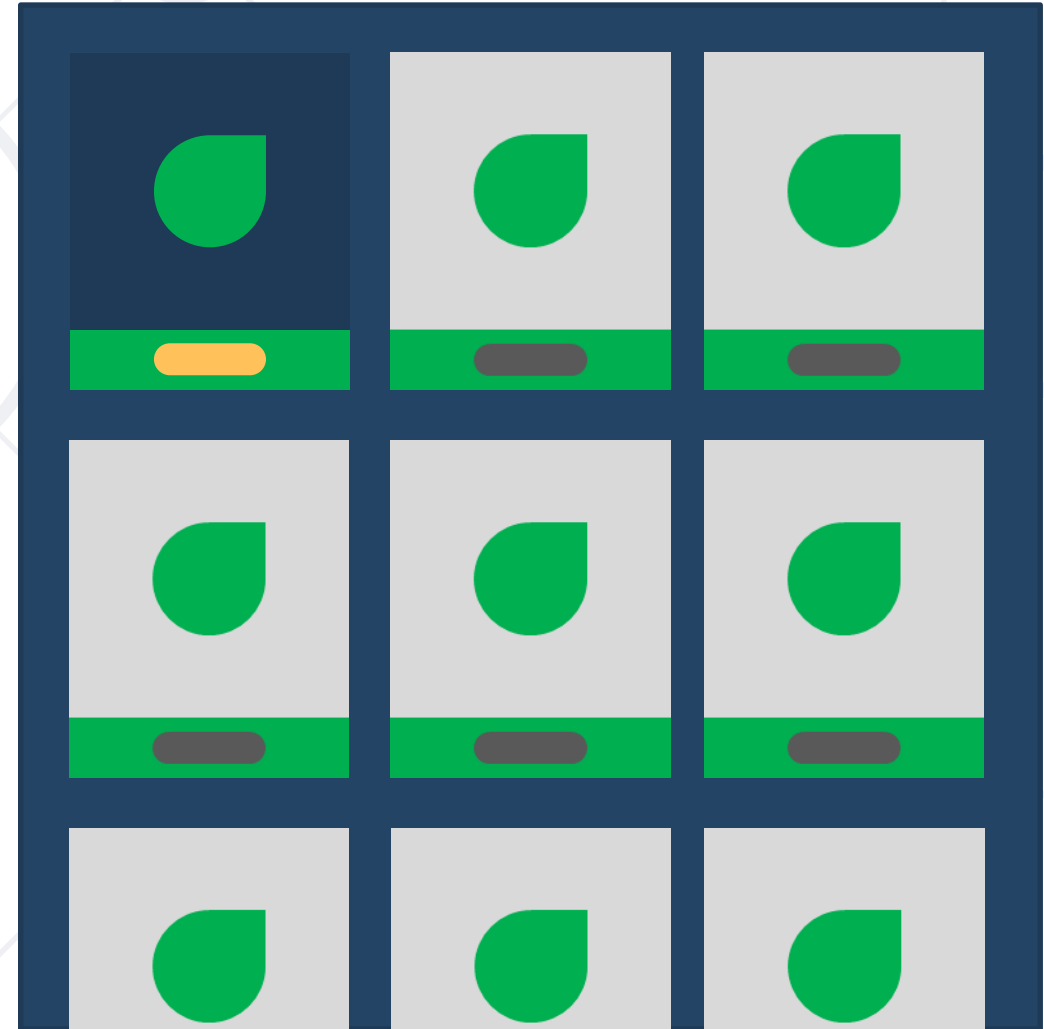
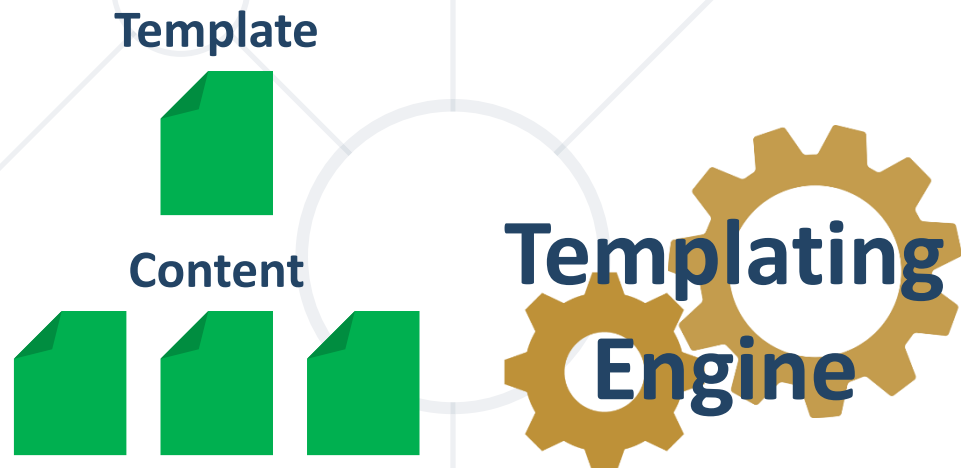
Examples

- Display articles in a blog
- Display a gallery of photos
- Visualize user profiles



Examples

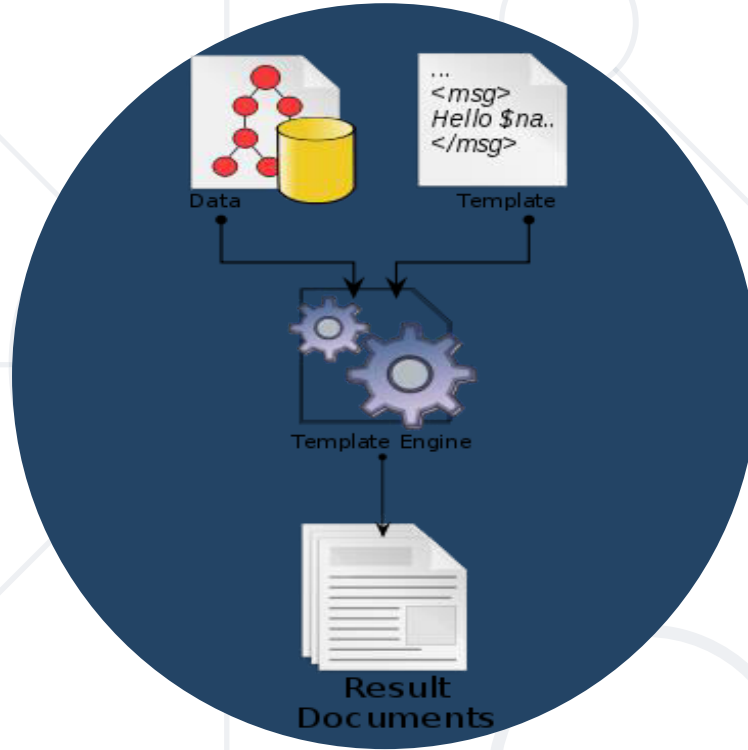
- Display articles in a blog
- Display a gallery of photos
- Visualize user profiles
- Show items in a catalog





Live Demo

Simple HTML Templating



Templating Engines

Overview of Popular JS Libraries

Templating Engines



Pug



Vue



jQuery Templates



Underscore Templates



Mustache



Handlebars



Templating with Handlebars

Syntax and Examples

- 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>
```

Example

```
<script id="entry-template" type="text/x-handlebars-template">
  <div class="entry">
    <h1>{{title}}</h1>
    <div class="body">{{body}}</div>
  </div>
</script>
```

My New Post

This is my first post!

```
let source = $("#entry-template").html();
let template = Handlebars.compile(source);
let context = {
  title: "My New Post",
  body: "This is my first post!"
};
let html = template(context);
```

Compile the template
from a string

Populate with data

- Download Handlebars using WebStorm's terminal

```
npm install --save handlebars
```

- Or download from handlebarsjs.com

- Browser builds will be located in

```
node_modules/handlebars/dist/
```

- Use handlebars from an online CDN:

```
https://cdnjs.cloudflare.com/ajax/libs/handlebars.js/4.0.10/  
handlebars.min.js
```

Note: It's best if your project has a **package.json** file

```
<!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>
```

- 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
- Set the context by executing the template with an object
- Named identifiers will be looked up in the object's properties
- Handlebars supports conditional statements and other helpers

```
<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>
```

- Once loaded, a template must be compiled

```
let source = $("#contact-template").html();  
let template = Handlebars.compile(source);
```

- Compiled templates are functions, that can be executed with what ever variables we need

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

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

```
let context = {  
  contacts: [  
    { name: 'Ivan Ivanov', email: 'i.ivanov@gmail.com'},  
    { name: 'Maria Petrova', email: 'mar4eto@abv.bg'},  
    { name: 'Jordan Kirov', email: 'jordk@gmail.com'}  
  ]};
```

```
<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 are templates that can be inserted into other templates

```
let source = $("#contact-template").html();  
Handlebars.registerPartial('contact', source);
```

Partial name

Template as **string**

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

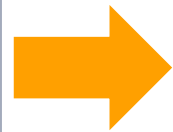
Partials are
globally accessible

- By default, any strings that are evaluated will be HTML-escaped
- To prevent this, use the "triple-stash"

```
title: "All about <p> Tags"
```

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

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



```
<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>
```



Live Demo

Templating with Handlebars



Live Exercises in Class (Lab)

Practice: Templating

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

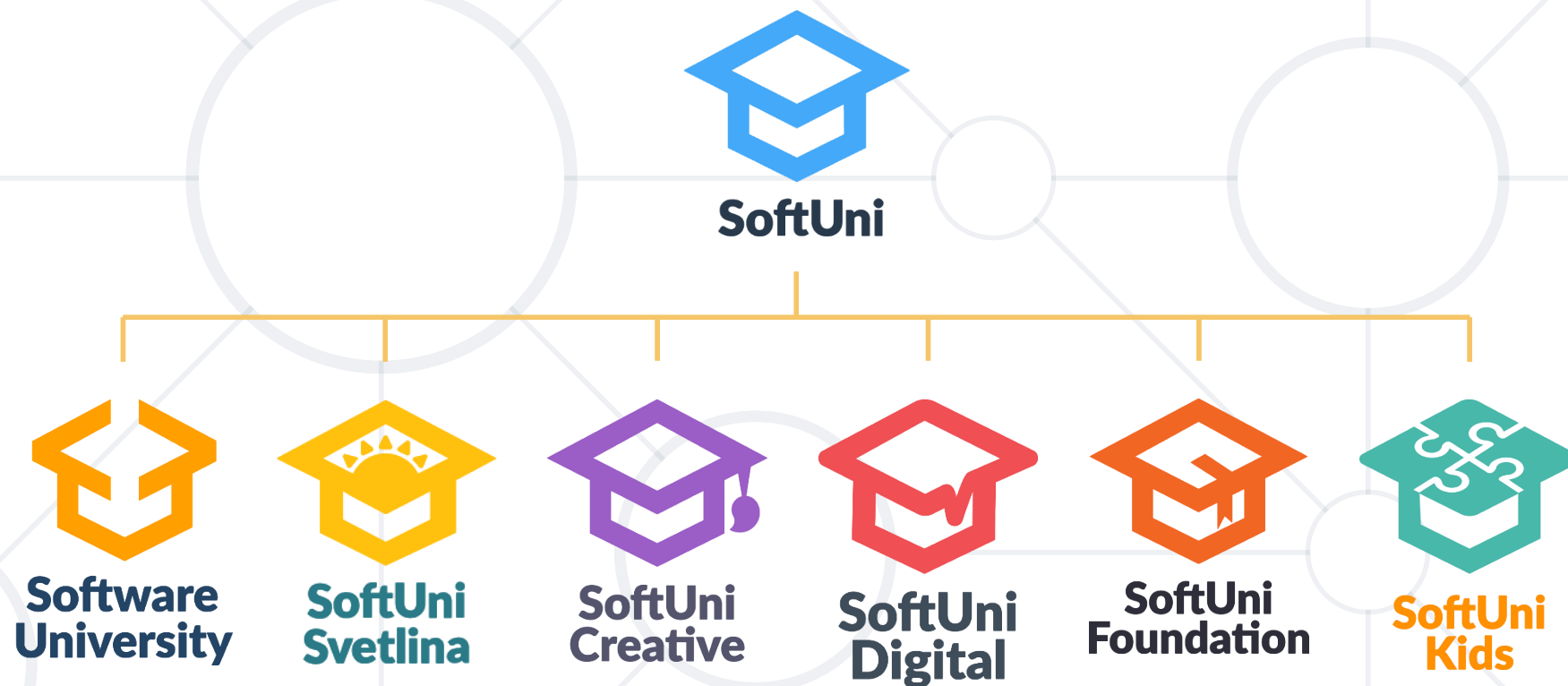
```
<div class="entry">  
  <h1>{{ title }}</h1>  
  <div class="body">  
    {{ body }}  
  </div>  
</div>
```



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



Questions?



SoftUni Diamond Partners



XSsoftware



SBTech



telenor



SoftwareGroup
doing it right

NETPEAK



SmartIT



Postbank

Решения за твоето утре

**SUPER
HOSTING
.BG**

INDEAVR

Serving the high achievers



INFRAGISTICS®

LIEBHERR

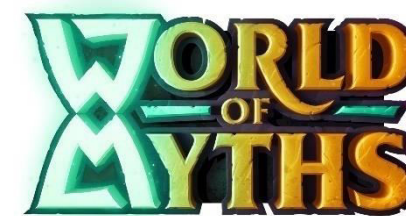


æternity



codexio

SoftUni Organizational Partners



Trainings @ Software University (SoftUni)

- Software University – High-Quality Education and Employment Opportunities
 - softuni.bg
- Software University Foundation
 - <http://softuni.foundation/>
- Software University @ Facebook
 - facebook.com/SoftwareUniversity
- Software University Forums
 - forum.softuni.bg



- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license

