**HTML Templates**

*For instructions on how to run the tests, check the following link:*[*https://github.com/TelerikAcademy/JavaScript-UI-and-DOM/blob/master/README.md#user-content-preparing-the-local-machine-for-unit-testing-with-mocha-and-chai*](https://github.com/TelerikAcademy/JavaScript-UI-and-DOM/blob/master/README.md#user-content-preparing-the-local-machine-for-unit-testing-with-mocha-and-chai)

**Task 1 - Just a table**

* Define a function that sets the template inside a provided element.
  + The template should generate the table with class .items-table following the rules:
    - A **data object** is provided and it contains two properties:
      * headers - an array of strings that should be used in the template to generate the headers of the table
      * items - an array of objects, where every object has keys col1, col2 and col3
  + *Example:*
    - Having:
    - data = {
    - headers : ['Vendor', 'Model', 'OS'],
    - items : [{
    - col1: 'Nokia',
    - col2: 'Lumia 920',
    - col3: 'Windows Phone'
    - }, {
    - col1: 'LG',
    - col2: 'Nexus 5',
    - col3: 'Android'
    - }, {
    - col1: 'Apple',
    - col2: 'iPhone 6',
    - col3: 'iOS'
    - }]
    - };
    - the template should generate:
    - <table class="items-table">
    - <thead>
    - <tr>
    - <th>#</th>
    - <th>Vendor</th>
    - <th>Model</th>
    - <th>OS</th>
    - </tr>
    - </thead>
    - <tbody>
    - <tr>
    - <td>0</td>
    - <td>Nokia</td>
    - <td>Limia 920</td>
    - <td>Windows Phone</td>
    - </tr>
    - <tr>
    - <td>1</td>
    - <td>LG</td>
    - <td>Nexus 5</td>
    - <td>Android</td>
    - </tr>
    - <tr>
    - <td>2</td>
    - <td>Apple</td>
    - <td>iPhone 6</td>
    - <td>iOS</td>
    - </tr>
    - </tbody>
    - </table>

**Task 2**

* Define a function that creates a div, containing a list with animals
  + Many of the elements have classes, see them in the example
  + The div should contain 2 elements:
    - A h1 that always has the content "Animals"
    - A UL with many LIs
  + Every LI element has a single element - an anchor (<a>)
    - The anchor has href and content
    - The href of the anchor may come from the **data object** or if it is missing, always points to an default URL
      * The default URL is "<http://cdn.playbuzz.com/cdn/3170bee8-985c-47bc-bbb5-2bcb41e85fe9/d8aa4750-deef-44ac-83a1-f2b5e6ee029a.jpg>"
    - The content of the anchor is always built from the **data object**, but is different if a href is available or not:
      * If a href is available, the content must be "See a THE\_NAME\_FROM\_THE\_DATA\_OBJECT"
      * Otherwise the content must be "No link for THE\_NAME\_FROM\_THE\_DATA\_OBJECT, here is Batman!"
  + *Example:*
    - Having:
    - var data = {
    - animals: [{
    - name: 'Lion',
    - url: 'https://susanmcmovies.files.wordpress.com/2014/12/the-lion-king-wallpaper-the-lion-king-2-simbas-pride-4685023-1024-768.jpg"
    - }, {
    - name: 'Turtle',
    - url: 'http://www.enkivillage.com/s/upload/images/a231e4349b9e3f28c740d802d4565eaf.jpg'
    - }, {
    - name: 'Dog'
    - }, {
    - name: 'Cat',
    - url: 'http://i.imgur.com/Ruuef.jpg'
    - }, {
    - name: 'Dog Again'
    - }]
    - }
    - Should produce:
    - <div class="container">
    - <h1>Animals</h1>
    - <ul class="animals-list">
    - <li>
    - <a href="https://susanmcmovies.files.wordpress.com/2014/12/the-lion-king-wallpaper-the-lion-king-2-simbas-pride-4685023-1024-768.jpg">See a Lion</a>
    - </li>
    - <li>
    - <a href="http://www.enkivillage.com/s/upload/images/a231e4349b9e3f28c740d802d4565eaf.jpg">See a Turtle</a>
    - </li>
    - <li>
    - <a href="http://cdn.playbuzz.com/cdn/3170bee8-985c-47bc-bbb5-2bcb41e85fe9/d8aa4750-deef-44ac-83a1-f2b5e6ee029a.jpg">No link for Dog, here is Batman!</a>
    - </li>
    - <li>
    - <a href="http://i.imgur.com/Ruuef.jpg">See a Cat</a>
    - </li>
    - <li>
    - <a href="http://cdn.playbuzz.com/cdn/3170bee8-985c-47bc-bbb5-2bcb41e85fe9/d8aa4750-deef-44ac-83a1-f2b5e6ee029a.jpg">No link for Dog Again, here is Batman!</a>
    - </li>
    - </ul>
    - </div>

**Task 3**

* Create a jQuery plugin for ListView
  + Apply a template for each item of a collection
  + Using the data-template attribute set the ID of the template to use for the items
  + Must work with different collections and templates
  + *Example:*
    - Having the HTML:
    - <ul id="books-list" data-template="book-item-template"></ul>
    - <script id="book-item-template" type="…">
    - <li class="book-item">
    - <a href="/#books/{{id}}">
    - <strong>{{title}}</strong>
    - </a>
    - </li>
    - </script>
    - And executing:
    - $('#books-list').listview(books);
    - Should generate:
    - <ul id="books-list" data-template="book-item-template">
    - <li class="book-item">
    - <a href="/#books/1">
    - <strong>JavaScript: The Good Parts</strong>
    - </a>
    - </li>
    - <li class="book-item">
    - <a href="/#books/2">
    - <strong>Secrets of the JavaScript Ninja</strong>
    - </a>
    - </li>
    - <li class="book-item">
    - <a href="/#books/3">
    - <strong>Core HTML5 Canvas</strong>
    - </a>
    - </li>
    - <li class="book-item">
    - <a href="/#books/4">
    - <strong>JavaScript Patterns</strong>
    - </a>
    - </li>

</ul>