

Every page that results from implementing the exercises must comply with HTML5 and CSS3 standards. The validation of the pages can be made using the services provided by <http://validator.w3.org> (HTML5) and <http://www.css-validator.org> (CSS3).

JQUERY

Please Note: There is a video for each exercise that illustrates the expected behavior of the web pages when the exercise is completed. It is recommended that you watch each video before you start the implementation of the associated exercise.

Before implementing the exercises, download the file "jQuery_Base.zip" from Moodle and expand its content to a folder. You can use the provided code and folder structure to solve the exercises.

All exercises require the use of the latest available version of jQuery library – use a CDN address (change X.XX.X for the correct version):

```
<script src="http://code.jquery.com/jquery-X.XX.X.min.js"></script>
```

This section (jQuery) uses only the jQuery library without any plugins.

- 1) [jq-alerbox.html, js/alerbox.js] Create a web page that asks the user's name and writes it in a message box. Use prompt and alert JavaScript functions and the `$(document).ready(...)` jQuery callback (or equivalent).
- 2) [jq-showhide.html, js/show-hide.js] Create a web page that shows and hides a section of a page when a user clicks or passes over a certain HTML element. Do not change the html file - except to add `<script>` elements (if required).
- 3) [jq-synchronize.html, js/synchronize.js] Create a web page that synchronizes the values of the fields of a form with specific HTML elements of the page. Do not change the html file - except to add `<script>` elements (if required).

Note: when viewing the corresponding video, pay attention to the differences in the behavior that are generated by the fields "Name" and "Grade". Also, try to use a jQuery selector to obtain the description for the selected evaluation period.

- 4) [jq-slideshow.html, js/slideshow.js] Create a web page that displays an image slide show with previous and next links for navigation. The images to display are placed on a specific folder ("slide_img") and have the following naming format: "img-**X**.jpeg", where **X** is an

index from 1 to a specified value. Create and use a function that receives the folder name and the last index to display: *function init(folderName, lastIndex)*. The slideshow should automatically change images every 2 seconds.

- 5) [jq-imdb.html, js/imdb.js] Create a simple web application that fetches IMDB records from the web service <http://www.omdbapi.com/?s=<user+words>> and lists the results. The page should display the title, year and thumbnail for each match. Clicking on the thumbnail should display the movie IMDB page.

JQUERY PLUGINS

This section (jQuery Plugins) uses jQuery plugins or widgets.

- 6) [jq-datepicker.html, js/datepicker.js] Create a web page that displays four fields associated with datepickers, as demonstrated on the video. Two of those fields display a Portuguese version of the calendar and the other two display the version from the United States. For each language there is a variant of the datepicker without any configuration and a version that applies some configuration rules.

Note: Use the DatePicker widget of the jQuery UI, and apply the "redmond" theme. Search for CDN downloads. For the localization files check the API documentation.

- 7) [jq-validation.html, js/validation.js] Validate the presented form which contains the fields "name", "grade" and "evaluation period" using the following rules:

- The field name is mandatory;
- The field grade must be an integer value between 0 and 20;
- A valid evaluation period must be selected;

The validation should be made in two moments:

- Whenever the value changes from a valid value to an invalid one;
- Before submitting the form data.

The data can only be submitted if the values of both fields are valid. For the validation process, use the jQuery plugin: jquery-validation (<http://jqueryvalidation.org/>). When the form is submitted with no errors an alert should be presented with the text "Form submitted". Make sure the error messages match the location and content shown on the video.

EXTRA CLASS

- 8) [jq-todolist.html, js/todo-list.js] Implement a to-do list application.
- a) When the checked image is clicked, the associated task text should be strikethrough. If the image is clicked again, the strikethrough should be removed. **Hint:** Inspect the HTML code for the markup to be used on each case.
 - b) Add tasks to the list by inserting its description on the "Add Task" field and clicking in the "add" button.
 - c) Each task can be removed by clicking the associated "remove" button
- 9) [jq-slideshow-2.html, js/slideshow-2.js] Create a web page that displays an image slide show based on the images of exercise 4. The slideshow must use the jQuery plugin Slick and employ the following settings (<http://kenwheeler.github.io/slick/>):
- show dots;
 - autoplay (2s between each transition). Pauses autoplay when the mouse is hover a dot;
 - shows 5 slides and scrolls one slide on each transition;
 - support the following responsive breakpoints:
 - $\leq 980\text{px}$ – shows 4 slides;
 - $\leq 768\text{px}$ – shows 3 slides;
 - $\leq 568\text{px}$ – shows only one slide and don't show dots;
- 10) [jq-sum-game.html, js/sum-game.js] Implement the SUM GAME using JavaScript and jQuery. The game consists of a board with a 4x3 random grid of numbers ranging from 1 to 12 and a target sum for each row. The goal of the game is to sort the grid so that the required sum for each row is correct. To assist the player during the game an extra column is filled for each row with the difference between the current sum and the target sum. To change the number position the player must drag the desired number to its target position. All the remaining numbers are pushed to the right. To achieve this behavior use the interaction "Sortable" from jQuery UI.

The player has a timeout (starts with 120s) to reorder the numbers so that the sum for each row matches the target sum. When the timeout expires the game ends. If the player completes the game before the timeout expires, the score is incremented by 78, the level increases by one, a new board is created, the timeout decrements by 10 and the game continues.