

# Exercise02: Javascript

---

## 1 Objectives

To learn to use Javascript and play with basics

- simple variables (local and global), conditions, loops, methods, and arrays.
- User input validations.
- Event Handling
- Canvas

**Work with your group (or by yourself). Each group should only upload one submission.**

## 1. Warm Up: Try Some Examples









1. First, open blackboard, go to Course Contents, and then download exercise02.zip file into your workspace (U:\workspace or something like that!). Then, unzip.
2. Play with each of the given examples (in the examples directory). Open them using a text editor of your choice and modify parts of the html or js files to learn how the different instructions work. If you want to use eclipse instead of notepad or vim or emacs etc., create a new static web project and create new html file and open it with a browser.
3. Note: w3schools.com is a good site to learn about web technologies.

Note that the assignment assumes you have understood these examples.



## 2 Form Validation

### 2.1 Create a form in HTML and validate entries of the form using javascript.

1. Create two files [validation1.html](#)
2. Change the TITLE of the validation1.html page to "Validation Form".
3. Create a HTML form in validation.html containing the fields as in the table below and also a continue button. Make it look reasonably good. Validation rules will be explained in next step.







Field Label	Field Type	Validation rule	Result
First Name	TextField	<b>*Required.</b> Must contain only alphabetical or numeric characters.	 / 
Last Name	TextField	<b>*Required.</b> Must contain only alphabetical or numeric characters.	 / 
Gender	Dropdown(male,female)	<b>*Required.</b>	 / 
State	Dropdown(California,Florida, New York,Texas,Hawaii,Washingt on,Colorado,Virginia, Iowa,Arizona)	<b>*Required,</b> select from given list and save the selected one to local storage	 / 

**\*Required field = Cannot be Empty.**



4. See [http://www.w3schools.com/js/js\\_validation.asp](http://www.w3schools.com/js/js_validation.asp). Create a file named **validation1.js**. Remember to include **validation1.js** in the head element of **validation.html**
5. Write Javascript code in **validation1.js** so that when user clicks **continue button** it does the following:
  - a) It validates the entries and displays  image (for each entry) if the validation was successful, else it displays  image. These images are included in the lab's zip file as **correct.png** and **wrong.png**.
  - b) Once the validation is successful, it goes to the next page (i.e. **validation2.html**)

## 2.2 Write another html file to enter Contact information Form

1. Create two files **validation2.html** and **validation2.js**
2. Change the TITLE of the **validation2.html** page to "Contact information".
3. Create a HTML form in **validation2.html** with the following Fields and a submit button. Make it look reasonably good.

Field Label	Field Type	Validation rule	Result
Email	TextField	<b>*Required</b> . Must be in the form <u>xxx@xxx.xxx</u> x should be alphanumeric (e.g. no special symbols).	 / 
Phone	TextField	Must be in the form xxx-xxx-xxxx or xxxxxxxxxx x should be numeric	 / 
address	TextField	<b>*Required</b> . Must have city, &state.  example: Ames, IA	 / 

**\*Required field = Cannot be Empty.**

4. See [http://www.w3schools.com/tags/att\\_input\\_pattern.asp](http://www.w3schools.com/tags/att_input_pattern.asp). Make sure to look at validation example in ExamplesJS folder. Write Javascript code in validation2.js to validate the above form when user clicks Submit button. Your code should display  image if the validation was successful, or if there was an error, display  image. Remember to include validation2.js in the head section of validation2.html .
5. When user clicks on submit:
  - a) if all field are valid, use HTML local storage [http://www.w3schools.com/html/html5\\_webstorage.asp](http://www.w3schools.com/html/html5_webstorage.asp) (click on link to find out more) to store the address only. Make sure to use text/string arguments when using localStorage.setItem.
  - b) go to a new page.
6. On the new page, retrieve local storage to show it on google charts by using geomap package. The following link <https://developers.google.com/chart/interactive/docs/gallery/geomap?csw=1> (click on link to find out more) has information on how to use Google map API to display a map with a location indicated by a circular marker. Go to markers section on the page. It has complete html + js code to use the google API.

### 3 Event Handling

Write a Javascript and HTML code to implement the functionality shown in

'Problem2Output.mp4' included in the .zip file.

Note:

- The line you create can go over any previous paths.
- The line should stop if it touches any boundary

Hints:

- Use HTML5 Canvas (see [http://www.w3schools.com/graphics/canvas\\_intro.asp](http://www.w3schools.com/graphics/canvas_intro.asp))
- **Make sure to use a timer** (see example below) to update the canvas (so that the snake keeps moving). Timer has two main functionality that can be used in the project
  - The `setInterval(function, delay)` schedules the "code" after every "delay" microseconds.
  - The `clearInterval` removes the timer

The following example will give you an alert every second until you press stop.

```
<html>
<body>

<h1><p id="header">COUNTDOWN IN SECONDS</p>
<p id="here"></p> </h1>
<input type="button" onclick="clearInterval(timer)" value="STOP COUNTDOWN"
">

<script>
  var i = 100;
  var timer = setInterval(function() {
    document.getElementById("here").innerHTML = i--;
  }, 1000)
</script>

</body>
</html>
```

#### 4 Submission:

Make sure your solutions work on Chrome (which is what TAs will use to grade the assignment). Zip your html, js files, and participation file (i.e. who worked on which part or if you worked together). Then, submit this zip file on black board. Remember there is only one submission per group. Make sure to include all the files that are needed in order to run your program.

Participation file is a simple txt file, which clarifies the specific participation of two members.