# JavaScript

## Introduction, Basics

See WEB Training I-1 Practice

## Functions, Objects

See WEB Training I-3 Practice.

Implement the following Underscore.js functions: bind, bindAll, partial, memoize, delay, defer, throttle, debounce, once, after, before, wrap, negate, compose.

See <http://underscorejs.org/#functions> for more details and examples.

# HTML, CSS

## Layout 1

Consider the following layout:

**Block A**

**Block B1**

**Block C**

**Block D1**

**Block D2 D@D2**

**Block B2**

Please create a HTML markup for it using:

1. CSS float property
2. CSS display:inline-block property

https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Block\_formatting\_context

1. Twitter Bootstrap 3 grid classes

http://getbootstrap.com/css/#grid

1. CSS Flexbox

https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Flexible\_boxes

Additional requirements are:

1. For the screen width greater than 1024px the layout should look like on the picture
2. For the screen width from 600 to 1024 px the block B1 should be placed below block B2
3. For the screen width less than 600px the block D2 should be placed below block D1, and block B1 should be below block B2.

Block properties:

* Block A. Width: 90%, height: 75px
* Block B1. Width: 35%, height: (screen height – Block A height)
* Block B2. Width: 65%, height: same as for B1
* Block C. Width: 90%, height: 75%
* Block D1. Width: 90%, height: 25%
* Block D2. Width 90%, height: 25%
* All blocks: padding 25px

## Layout 2

Place all four layouts from Layout 1 on a single page and implement using CSS (no JavaScript or JQuery) switch control: hovering a mouse on blocks 1-4 changes a layout in block L. *Or prove it’s not possible* ☺

1

2

3

4

L

# Browser JavaScript

## JQuery

Implement the following JQuery functions:

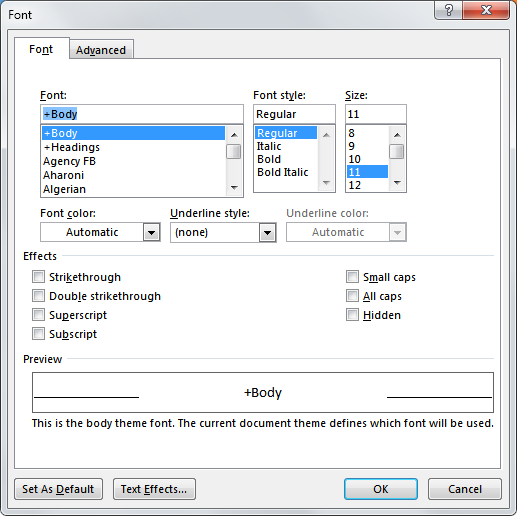
$(<selector>), width(), height(), hasClass(), addClass(), ,toggleClass(), removeClass(), wrap(), append(), prepend()

Use raw DOM API for this task. Notes:

* $() should return an array of matched elements
* Methods $(), addClass, toggleClass, removeClass, wrap, append should provide chaining

## Font Dialog

Consider the following Font selection dialog:



Implement it in browser using JQuery and Twitter Bootstrap 3. Implementation notes:

* Use Bootstrap controls: buttons, comboboxes, lists
* No need to implement “Advanced” tab (however “Font” tab should be present)
* Dialog should be a Bootstrap dialog and should be opened when a special “Font settings…” button is pressed
* “OK” button should close the dialog and display CSS rules required to apply the selected settings in a special box
* List of fonts should be populated with fonts from the Google Fonts directory