

|  |
| --- |
| Test Automation Mentoring Program  JAVASCRIPT: ENTRY TASK |
|  |

CONTENT

[TYpes. functions. objects 3](#_Toc494727085)

[Task 1 3](#_Toc494727086)

[Task 2 3](#_Toc494727087)

[Task 3 3](#_Toc494727088)

[NODE JS. AUTOMATION 4](#_Toc494727089)

[Task 4 4](#_Toc494727090)

# TYpes. functions. objects

The tasks below are around basic JS knowledge, related to data types, functions and JS objects. Use any ES notation to create appropriate scripts. Design is up to you. Use Google Chrome console or any other way to debug/run the scripts. The final result should be formed in several independent scripts.

## Task 1

Create a function(s) that fulfills array with 10 random integer values and returns a multiply of 3 greatest values.

*Materials:*

[*https://www.w3schools.com/js/js\_functions.asp*](https://www.w3schools.com/js/js_functions.asp)

[*https://www.w3schools.com/js/js\_arrays.asp*](https://www.w3schools.com/js/js_arrays.asp)

[*https://www.w3schools.com/js/js\_random.asp*](https://www.w3schools.com/js/js_random.asp)

## Task 2

Create a function(s) that returns total amount of seconds starting from the beginning of today and till now.

*Materials:*

[*https://www.w3schools.com/js/js\_functions.asp*](https://www.w3schools.com/js/js_functions.asp)

[*https://www.w3schools.com/js/js\_dates.asp*](https://www.w3schools.com/js/js_dates.asp)

## Task 3

Define an object **happiness** within 5 properties. Each property should have default integer value, representing its priority. Then create a function that returns a list of property names sorted by priority (highest is on top).

# NODE JS. AUTOMATION

The task below is about simple automation that could be created based on JS.

The TAF structure and tests’ implementation is up to you. The goal is to have a workable and stable solution to perform testing according to provided steps.

## Task 4

Create a simple TAF to perform e2e testing according to steps below. Result should be sent as archive of TAF folder.

Pre. Manually create 2 email accounts on *tut*, *gmail* or any other mail service;

1. Login into account 1;
2. Send email to account 2. Email body is up to you;
3. Check presence of e-mail in Sent folder;
4. Login into account 2;
5. Check presence of email sent from account 1.

Use Node.JS + Jasmine + WebDriverIO or Protractor.

*Tips: use randomize to form email Subject in order to make it unique in terms of multi-time execution.*

*Materials:*

<https://nodejs.org/en/>

<https://docs.angularjs.org/guide/e2e-testing>

<https://jasmine.github.io/>

<http://www.protractortest.org/#/>

<http://webdriver.io/>

<http://webdriver.io/guide/testrunner/pageobjects.html>

<https://www.joecolantonio.com/2016/06/14/top-8-essential-javascript-automation-frameworks/>