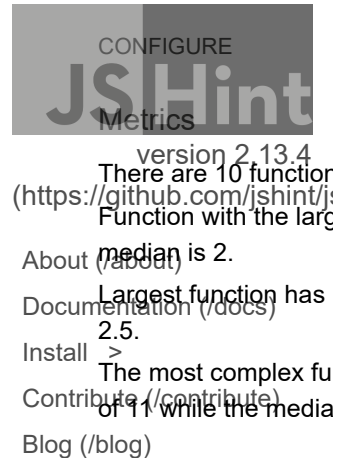


```

1  /* jshint esversion: 8 */
2  // helper functions
3
4  /**
5   * Function to add class
6   * @param {*} className
7   * @param {*} targetNode
8   */
9  function addClass(className, targetNode) {
10     targetNode.classList.add(className);
11 }
12
13 /**
14 * Function to remove class
15 * @param {*} className
16 * @param {*} targetNode
17 */
18
19 function removeClass(className, targetNode) {
20     targetNode.classList.remove(className);
21 }
22
23 /**
24 * Function to display Error after validation has been failed
25 * makes div with help message visible and in red,
26 * input's border is red and red icon
27 * with exclamation mark is displayed in input field
28 * @param {*} targetNodeInput
29 * @param {*} targetNodeHelp
30 */
31 function displayErrorValidation(targetNodeInput, targetNodeHelp) {
32     // input nodes
33     addClass("is-invalid", targetNodeInput);
34     removeClass("is-valid", targetNodeInput);
35     // help nodes
36     removeClass("d-none", targetNodeHelp);
37     addClass("invalid-feedback", targetNodeHelp);
38     removeClass("valid-feedback", targetNodeHelp);
39 }
40
41
42 /**
43 * Function to display Error after validation has been failed
44 * makes div with help message visible and in red, input's border is red
45 * and red icon with exclamation mark is displayed in input field
46 * @param {*} targetNodeInput
47 * @param {*} targetNodeHelp
48 */
49 function displayPassedValidation(targetNodeInput, targetNodeHelp) {
50     // input nodes
51     addClass("is-valid", targetNodeInput);
52     removeClass("is-invalid", targetNodeInput);
53     // help nodes
54     removeClass("d-none", targetNodeHelp);
55     addClass("valid-feedback", targetNodeHelp);
56     removeClass("invalid-feedback", targetNodeHelp);
57 }
58
59 /**
60 * function to validate if input is NOT empty
61 * @returns true or false
62 */
63
64 function isEmpty(inputNode, helpNode) {
65     if (inputNode.value === "") {
66         helpNode.innerHTML = "This field is required";
67         displayErrorValidation(inputNode, helpNode);
68         return (false);
69     } else {
70


```



```

71     helpNode.innerHTML = "All ok";
72     displayPassedValidation(inputNode, helpNode);
73     return (true);
74 }
75 }
76
77 /**
78  * function to validate if both arguments are met: regex and not empty
79  */
80 function validateInput(inputNode, helpNode, regex, errorMessage) {
81
82     if (inputNode.value === "") {
83         helpNode.innerHTML = "This field is required";
84         displayErrorValidation(inputNode, helpNode);
85         return false;
86     } else if (regex.test(inputNode.value) === false) {
87         helpNode.innerHTML = errorMessage;
88         displayErrorValidation(inputNode, helpNode);
89         return (false);
90     } else {
91         helpNode.innerHTML = "All ok";
92         displayPassedValidation(inputNode, helpNode);
93         return true;
94     }
95 }
96
97 // variables fetching the elements from the form
98
99 const contactForm = document.getElementsByTagName("FORM")[0];
100
101 // id strings for input
102 const stringIdContactName = "contact-name";
103 const stringIdEmail = "email";
104 const stringIdTelephone = "telephone";
105 const stringIdCategory = "input-category";
106 const stringIdProductName = "product-name";
107 const stringIdDescription = "description";
108 const stringIdShopName = "shop-name";
109 const stringIdPrice = "price";
110 const stringIdWebsite = "website";
111 const stringIdImage = "image";
112
113 // input elements:
114 const inputContactName = document.getElementById(stringIdContactName);
115 const inputEmail = document.getElementById(stringIdEmail);
116 const inputTelephone = document.getElementById(stringIdTelephone);
117 const inputCategory = document.getElementById(stringIdCategory);
118 const inputProductName = document.getElementById(stringIdProductName);
119 const inputDescription = document.getElementById(stringIdDescription);
120 const inputShopName = document.getElementById(stringIdShopName);
121 const inputPrice = document.getElementById(stringIdPrice);
122 const inputWebsite = document.getElementById(stringIdWebsite);
123 const inputImage = document.getElementById(stringIdImage);
124
125 // help divs id strings
126 const stringIdContactNameHelp = stringIdContactName + "-help";
127 const stringIdEmailHelp = stringIdEmail + "-help";
128 const stringIdTelephoneHelp = stringIdTelephone + "-help";
129 const stringIdCategoryHelp = stringIdCategory + "-help";
130 const stringIdProductNameHelp = stringIdProductName + "-help";
131 const stringIdDescriptionHelp = stringIdDescription + "-help";
132 const stringIdShopNameHelp = stringIdShopName + "-help";
133 const stringIdPriceHelp = stringIdPrice + "-help";
134 const stringIdWebsiteHelp = stringIdWebsite + "-help";
135 const stringIdImageHelp = stringIdImage + "-help";
136
137
138 // help divs elements:
139 const helpContactName = document.getElementById(stringIdContactNameHelp);
140 const helpEmail = document.getElementById(stringIdEmailHelp);
141 const helpTelephone = document.getElementById(stringIdTelephoneHelp);
142 const helpCategory = document.getElementById(stringIdCategoryHelp);

```



CONFIGURE
JSHint
Metrics

version 2.13.4
There are 10 function
(<https://github.com/jshint/jshint/>)
Function with the large
median is 2.
Largest function has
Documentation (/docs)
2.5.
Install >
The most complex fu
Contribute (/contribute)
of 11 while the media
Blog (/blog)

CONFIGURE

JSHint

Metrics

version 2.13.4

There are 10 function

(https://github.com/jshint/j

Function with the larg

median is 2.

Largest function has

Documentation (docs)

2.5.

The most complex fu

Contribute ((contribute)

Blog (/blog)

```
215     case stringIdTelephone:
216         validateInput(inputTelephone, helpTelephone, regexTelephone, errorMessageTelephone);
217         break;
218     case stringIdCategory:
219         isEmpty(inputCategory, helpCategory);
220         break;
221     case stringIdProductName:
222         isEmpty(inputProductName, helpProductName);
223         break;
224     case stringIdDescription:
225         isEmpty(inputDescription, helpDescription);
226         break;
227     case stringIdShopName:
228         isEmpty(inputShopName, helpShopName);
229         break;
230     case stringIdPrice:
231         validateInput(inputPrice, helpPrice, regexPrice, errorMessagePrice);
232         break;
233     case stringIdWebsite:
234         validateInput(inputWebsite, helpWebsite, regexURL, errorMessageURL);
235         break;
236     case stringIdImage:
237         validateInput(inputImage, helpImage, regexURL, errorMessageURL);
238         break;
239     }
240 }));
```

CONFIGURE

JSHint

Metrics

version 2.13.4

There are 10 function

(https://github.com/jshint/j

Function with the larg

median is 2.

About (/about)

Largest function has

Documentation (/docs)

2.5.

Install >

The most complex fu

Contribute (/contribute)

of 11 while the media

Blog (/blog)