

```
1 <script>
2 const validator = (function () {
3     let is_valid = true;
4
5     return {
6         isNumeric: function (text) {
7             switch (typeof text) {
8                 case "number":
9                     break;
10                case "string":
11                    this.is_valid = !isNaN(text);
12                    break;
13                default:
14                    this.is_valid = false;
15                    break;
16            }
17            return this.is_valid;
18        },
19        isInteger: function (text) {
20            if (validator.isNumeric(text)) {
21                if (typeof text === "string") {
22                    text = Number(text);
23                }
24                if (!Number.isInteger(text)) {
25                    this.is_valid = false;
26                    return false;
27                }
28            }
29            else {
30                this.is_valid = false;
31                return false;
32            }
33            return this.is_valid;
34        },
35        isNegativeInteger: function (text) {
36            if (validator.isInteger(text)) {
37                text = Number(text);
38                if (text < 0) {
39                    return true;
40                }
41                else {
42                    this.is_valid = false;
43                    return false;
44                }
45            }
46            else {
47                this.is_valid = false;
48                return false;
49            }
50        },
51        isPositiveInteger: function (text) {
52            if (validator.isInteger(text)) {
53                text = Number(text);
54                if (text > 0) {
55                    return true;
56                }
57                else {
58                    this.is_valid = false;
59                    return false;
60                }
61            }
62        }
63    };
64 }
```

```

62         else {
63             this.is_valid = false;
64             return false;
65         }
66     },
67     isNonNegativeInteger: function (text) {
68         if (validator.isInteger(text)) {
69             text = Number(text);
70             if (text >= 0) {
71                 return true;
72             }
73             else {
74                 this.is_valid = false;
75                 return false;
76             }
77         }
78         else {
79             this.is_valid = false;
80             return false;
81         }
82     },
83     isInRange: function (text, m, n) {
84         if (!validator.isNumeric(text)) {
85             this.is_valid = false;
86             return false;
87         }
88         if (typeof m !== "undefined") {
89             if (!validator.isNumeric(m)) {
90                 this.is_valid = false;
91                 return false;
92             }
93             if (Number(text) < Number(m)) {
94                 this.is_valid = false;
95                 return false;
96             }
97         }
98         if (typeof n !== "undefined") {
99             if (!validator.isNumeric(n)) {
100                 this.is_valid = false;
101                 return false;
102             }
103             if (Number(text) > Number(n)) {
104                 this.is_valid = false;
105                 return false;
106             }
107         }
108         return true;
109     },
110     isValidEmail: function (text) {
111         let re = /[a-z0-9!#$%&'*/+=?^_`{|}~-]+(?:\.[a-z0-9!#$%&'*/+=?^_`{|}~-]+)*@(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?/g
112         if (re.test(text)) {
113             return true;
114         }
115         else {
116             this.is_valid = false;
117             return false;
118         }
119     },
120     isEmpty: function (obj) {
121         if (typeof obj === "boolean") {

```

```
122         return true;
123     }
124     if (obj === 0) {
125         return true;
126     }
127     for(var key in obj) {
128         if(obj.hasOwnProperty(key))
129             return true;
130     }
131     this.is_valid = false;
132     return false;
133 },
134 matchesRegex: function (text, regex) {
135     let re = new RegExp(regex);
136     if (re.test(text)) {
137         return true;
138     }
139     else {
140         this.is_valid = false;
141         return false;
142     }
143 },
144 lengthIsInRange: function (text, m, n) {
145     if (typeof text !== "string") {
146         this.is_valid = false;
147         return false;
148     }
149     validator.isInRange(text.length, m, n);
150 },
151 isValid: function () {
152     return this.is_valid;
153 },
154 reset: function () { this.is_valid = true; },
155 logStatus: function () {
156     if (validator.isValid()) {
157         console.log('All is well');
158     } else {
159         console.log('Something failed validation'); }
160 },
161 logNegativeStatus: function () {
162     if (validator.isValid()) {
163         console.log('Something failed validation');
164     } else {
165         console.log('All is well'); }
166 }
167 };
168 }());
169
170
171 validator.reset();
172 validator.isNumeric(5);
173 validator.isNumeric('5');
174 validator.isInteger(5);
175 validator.isInteger('50');
176 validator.logStatus();
177
178 validator.isInteger("one");
179 validator.logNegativeStatus();
180 validator.reset();
181 validator.isPositiveInteger(5);
182 validator.isNegativeInteger(-5);
```

```
183     validator.logStatus();
184
185     validator.isNegativeInteger(-5.6);
186     validator.logNegativeStatus();
187     validator.reset();
188
189     validator.isNonNegativeInteger(0);
190     validator.isInRange(50, "4", "1000");
191     validator.isInRange("50", 4, "1000");
192     validator.isValidEmail("kylebharrison16@gmail.com");
193     validator.isValidEmail("kyle@c.d");
194     validator.isEmpty(false);
195     validator.isEmpty(".");
196     validator.logStatus();
197
198     validator.lengthIsInRange("12345", 5, 5);
199     validator.matchesRegex("hello", "^hello[, my friend]?");
200     validator.isEmpty("89");
201     validator.logStatus();
202
203 </script>
204
205
```