

```

1 $(document).ready(function() {
2   /* from Code Institute Boutique ado */
3
4   // Disable +/- buttons outside 1-99 range
5   function handleEnableDisable(itemId) {
6     let currentValue = parseInt($('#id_qty_${itemId}`).val());
7     let minusDisabled = currentValue < 2;
8     let plusDisabled = currentValue > 98;
9     $('#decrement-qty_${itemId}`).prop('disabled', minusDisabled);
10    $('#increment-qty_${itemId}`).prop('disabled', plusDisabled);
11  }
12
13  // Ensure proper enabling/disabling of all inputs on page load
14  let allQtyInputs = $('.qty_input');
15  for(let i = 0; i < allQtyInputs.length; i++){
16    let itemId = $(allQtyInputs[i]).data('item_id');
17    handleEnableDisable(itemId);
18  }
19
20  // Check enable/disable every time the input is changed
21  $('.qty_input').change(function() {
22    let itemId = $(this).data('item_id');
23    handleEnableDisable(itemId);
24  });
25
26  // Increment quantity
27  $('#increment-qty').click(function(e) {
28    e.preventDefault();
29    let closestInput = $(this).closest('.input-group').find('.qty_input')[0];
30    let currentValue = parseInt($(closestInput).val());
31    $(closestInput).val(currentValue + 1);
32    let itemId = $(this).data('item_id');
33    handleEnableDisable(itemId);
34  });
35
36  // Decrement quantity
37  $('#decrement-qty').click(function(e) {
38    e.preventDefault();
39    let closestInput = $(this).closest('.input-group').find('.qty_input')[0];
40    let currentValue = parseInt($(closestInput).val());
41    $(closestInput).val(currentValue - 1);
42    let itemId = $(this).data('item_id');
43    handleEnableDisable(itemId);
44  });
45
46  // Update quantity on click

```

## CONFIGURE

### Metrics

There are 8 functions in this file.

Function with the largest signature take 1 arguments, while the median is 1.

Largest function has 10 statements in it, while the median is 4.5.

The most complex function has a cyclomatic complexity value of 2 while the median is 1.



version 2.13.1

(<https://github.com/jshint/jshint>)

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

### 21 warnings

- 6 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 6 'template literal syntax' is only available in ES6 (use 'esversion: 6').
- 7 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 8 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 9 'template literal syntax' is only available in ES6 (use 'esversion: 6').
- 10 'template literal syntax' is only available in ES6 (use 'esversion: 6').
- 14 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 15 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 16 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

```

48     $('update-link').click(function(e) {
49         let form = $(this).parent().prev('update-form');
50         form.submit();
51     });
52
53     // Remove item and reload on click
54     $('remove-item').click(function(e) {
55         let itemId = $(this).attr('id').split('remove_')[1];
56         let url = `/cart/remove/${itemId}/`;
57         // middleware requited by Django
58         let data = {'csrfmiddlewaretoken': csrfToken};
59
60         $.post(url, data)
61             .done(function() {
62                 location.reload();
63             });
64     });
65 });
66

```

## CONFIGURE

### Metrics

There are 8 functions in this file.

Function with the largest signature take 1 arguments, while the median is 1.

Largest function has 10 statements in it, while the median is 4.5.

The most complex function has a cyclomatic complexity value of 2 while the median is 1.



version 2.13.1

(<https://github.com/jshint/jshint>)

[About \(/about\)](#)

[Documentation \(/docs\)](#)

[Install \(/install\)](#)

[Contribute \(/contribute\)](#)

[Blog \(/blog\)](#)

### 21 warnings

- 6 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 6 'template literal syntax' is only available in ES6 (use 'esversion: 6').
- 7 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 8 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 9 'template literal syntax' is only available in ES6 (use 'esversion: 6').
- 10 'template literal syntax' is only available in ES6 (use 'esversion: 6').
- 14 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 15 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 16 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).