

# JavaScript Validation

## Bag

```
16  /* jshint esversion: 6 */
17
18  $('update-link').click(function(e) {
19      var form = $(this).prev('update-form');
20      form.submit();
21  });
22
23  // Remove item and reload on click
24  $('remove-item').click(function(e) {
25      var csrfToken = "{{ csrf_token }}";
26      var itemId = $(this).attr('id').split('remove_')[1];
27      var url = `/bag/delete/${itemId}/`;
28      var data = {'csrfmiddlewaretoken': csrfToken};
29
30      $.post(url, data)
31      .done(function() {
32          location.reload();
33      });
34  });
```

One undefined variable

18	\$
19	\$
24	\$
26	\$
30	\$

# JavaScript Validation

## Handles checkboxes (checkout.html)

```

17 // js: the version is 0,
18 document.addEventListener('DOMContentLoaded', function () {
19     // Function to check if UserProfileForm is filled out
20     function isUserProfileFormFilled() {
21         var paymentOption = document.querySelector('input[name="{{ order_f
22         var invoiceRef = document.getElementById('id_invoice_ref');
23
24         if (!paymentOption) {
25             return false; // Disable the button by default
26         }
27
28         if (paymentOption.value === 'invoice') {
29             var requiredFields = [
30                 "{{ user_profile.company_name|default_if_none:'' }}",
31                 "{{ user_profile.org_num|default_if_none:'' }}",
32                 "{{ user_profile.street_address1|default_if_none:'' }}",
33                 "{{ user_profile.postcode|default_if_none:'' }}",
34                 "{{ user_profile.city|default_if_none:'' }}",
35                 "{{ user_profile.country|default_if_none:'' }}",
36                 "{{ user_profile.invoice_email|default_if_none:'' }}",
37             ];
38
39             return requiredFields.every(field => field.trim() !== 'None' &
40         }
41
42         // Enable the button for card payment or other options
43         return true;
44     }
45
46     // Function to update button state based on UserProfileForm
47     function updateButtonState() {
48         var completeOrderButton = document.getElementById('complete-order-1
49         completeOrderButton.disabled = !isUserProfileFormFilled();
50     }
51
52     // Add event listeners for input events for each input field
53     var fields = [
54         'company_name',
55         'org_num',
56         'street_address1',
57         'postcode',
58         'city',
59         'country',

```

# JavaScript Validation

## Stripe\_elements.js

```
16  /* jshint esversion: 6 */
17
18      var stripePublicKey = JSON.parse(document.getElementById('id_stripe_public_key')
19  var clientSecret = JSON.parse(document.getElementById('id_client_secret').textConte
20  var stripe = Stripe(stripePublicKey);
21  var elements = stripe.elements();
22  var style = {
23      base: {
24          color: '#000',
25          fontFamily: "'Ubuntu', sans-serif",
26          fontSmoothing: 'antialiased',
27          fontSize: '16px',
28          '::placeholder': {
29              color: '#aab7c4'
30          }
31      },
32      invalid: {
33          color: '#dc3545',
34          iconColor: '#dc3545'
35      }
36  };
37  var card = elements.create('card', {style: style});
38  card.mount('#card-element');
39
40  // Handle realtime validation errors on the card element
41  card.addEventListener('change', function (event) {
42      var errorDiv = document.getElementById('card-errors');
43      if (event.error) {
44          var html = `
45              <span class="icon" role="alert">
46                  <i class="fas fa-times"></i>
47              </span>
48              <span>${event.error.message}</span>
49          `;
50          $(errorDiv).html(html);
51      } else {
52          errorDiv.textContent = '';
53      }
54  });
55
56  // Handle form submit
57  var form = document.getElementById('payment-form');
```

Two undefined variables

20 Stripe

50 \$

63 \$

77 \$

79 \$

# JavaScript Validation

Add product image (add\_product.html)

```
15
16  /* jshint esversion: 6 */
17
18  $('#new-image').change(function() {
19      var file = $('#new-image')[0].files[0];
20      $('#filename').text(`Image will be set to: ${file.name}`);
21  });
```

One undefined variable

18	\$
19	\$
20	\$

# JavaScript Validation

Filter category (products.html)

```
15
16  /* jshint esversion: 6 */
17
18  $(document).ready(function () {
19      $('.category-checkbox').change(function () {
20          filterProducts();
21      });
22
23      // Handle the click event on category labels
24      $('.category-item label').click(function () {
25          var checkbox = $(this).find('.category-checkbox');
26          checkbox.prop('checked', !checkbox.prop('checked'));
27          filterProducts();
28      });
29
30      function filterProducts() {
31          // Get the values of the selected category names
32          var selectedCategories = $('.category-checkbox:checked').map(function () {
33              return $(this).val();
34          }).get();
35
36          // Show/hide products based on the selected categories
37          $('.product-item').each(function () {
38              var productCategories = $(this).data('categories').split(',');
39
40              // Show the product if any selected category matches
41              if (selectedCategories.length === 0 || selectedCategories.some(category => productCategories.includes(category))) {
42                  $(this).show();
43              } else {
44                  $(this).hide();
45              }
46          });
47      }
48  });
```

One undefined variable

18 \$

19 \$

24 \$

25 \$

32 \$

33 \$

37 \$

38 \$

42 \$

44 \$

# JavaScript Validation

Increment quantity (quantity\_input\_script.html)

```
16  /* jshint esversion: 6 */
17
18  // Disable +/- buttons outside 1-99 range
19  function handleEnableDisable(itemId) {
20      var currentValue = parseInt($('#id_qty_' + itemId).val());
21      var minusDisabled = currentValue < 2;
22      var plusDisabled = currentValue > 98;
23      $('#decrement-qty_' + itemId).prop('disabled', minusDisabled);
24      $('#increment-qty_' + itemId).prop('disabled', plusDisabled);
25  }
26
27  // Ensure proper enabling/disabling of all inputs on page load
28  var allQtyInputs = $('.qty_input');
29  for(var i = 0; i < allQtyInputs.length; i++){
30      var itemId = $(allQtyInputs[i]).data('item_id');
31      handleEnableDisable(itemId);
32  }
33
34  // Check enable/disable every time the input is changed
35  $('.qty_input').change(function() {
36      var itemId = $(this).data('item_id');
37      handleEnableDisable(itemId);
38  });
39
40  // Increment quantity
41  $('#increment-qty').click(function(e) {
42      e.preventDefault();
43      var closestInput = $(this).closest('input-group').find('qty_input')[0];
44      var currentValue = parseInt($(closestInput).val());
45      $(closestInput).val(currentValue + 1);
46      var itemId = $(this).data('item_id');
47      handleEnableDisable(itemId);
48  });
49
50  // Decrement quantity
51  $('#decrement-qty').click(function(e) {
52      e.preventDefault();
53      var closestInput = $(this).closest('input-group').find('qty_input')[0];
```

One undefined variable

20	\$
23	\$
24	\$
28	\$
30	\$
35	\$
36	\$
41	\$
43	\$
44	\$
45	\$
46	\$
51	\$
53	\$
54	\$
55	\$
56	\$



# JavaScript Validation

Toasts (base.html)

```
15
16  /* jshint esversion: 6 */
17
18  $(document).ready(function() {
19      $('.toast').toast('show');
20
21      // Hide the toast after 7 seconds
22      setTimeout(function() {
23          $('.toast').toast('hide');
24      }, 7000);
25  });
```

One undefined variable

18	\$
19	\$
23	\$