

Basket.py

```
1 // Update quantity on click
2 $('.update-link').click(function (e) {
3     var form = $(this).prev('.update-form');
4     form.submit();
5 });
6
7 // Remove item and reload on click
8 $('.remove-item').click(function (e) {
9     var csrfToken = "{{ csrf_token }}";
10    var itemId = $(this).attr('id').split('remove_')[1];
11    var url = 'remove/' + itemId;
12    var data = {
13        'csrfmiddlewaretoken': csrfToken,
14    };
15
16    $.post(url, data)
17        .done(function () {
18            location.reload();
19        });
20 });
21
22
```

CONFIGURE

Metrics

There are 3 functions in this file.

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

Largest function has 5 statements in it, while the median is 2.

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

Company_faq.html

```
1 function openPage(pageName, elmnt, color) {
2     // Hide all elements with class="tabcontent" by default */
3     var i, tabcontent, tablinks;
4     tabcontent = document.getElementsByClassName("tabcontent");
5     for (i = 0; i < tabcontent.length; i++) {
6         tabcontent[i].style.display = "none";
7     }
8
9     // Remove the background color of all tablinks/buttons
10    tablinks = document.getElementsByClassName("tablink");
11    for (i = 0; i < tablinks.length; i++) {
12        tablinks[i].style.backgroundColor = "";
13    }
14
15    // Show the specific tab content
16    document.getElementById(pageName).style.display = "block";
17 }
18
19 // Get the element with id="defaultOpen" and click on it
20 document.getElementById("defaultOpen").click();
```

CONFIGURE

Metrics

There is only one function in this file.

It takes 3 arguments.

This function contains 8 statements.

Cyclomatic complexity number for this function is 3.

One unused variable

1 openPage

openPage variable is referenced further up on the page:

```
<button class="tablink" onclick="openPage('Returns', this)" id="defaultOpen">Returns</button>
<button class="tablink" onclick="openPage('Order', this)">Order</button>
<button class="tablink" onclick="openPage('Delivery', this)">Delivery</button>
<button class="tablink" onclick="openPage('Products', this)">Products</button>
```

quantity_btn_script:

```
1 // Increment the quantity of a product in the shopping basket
2 $('.increment-qty').click(function (e) {
3     e.preventDefault();
4     var closestInput = $(this).closest('.input-group').find('input')[0];
5     var currentValue = parseInt(closestInput.val());
6     closestInput.val(currentValue + 1);
7 });
8
9
10 // Decrement the quantity of a product in the shopping basket
11 $('.decrement-qty').click(function (e) {
12     e.preventDefault();
13     var closestInput = $(this).closest('.input-group').find('input')[0];
14     var currentValue = parseInt(closestInput.val());
15     closestInput.val(currentValue - 1);
16 });
17
```

CONFIGURE

Metrics

There are 2 functions in this file.

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

Largest function has 4 statements in it, while the median is 4.

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

back-to-top-button.js

```
1 let mybutton = document.getElementById("myBtn");
2
3 window.onscroll = function () {
4   scrollFunction();
5 };
6
7 function scrollFunction() {
8   if (document.body.scrollTop > 20 || document.documentElement.scrollTop > 20) {
9     mybutton.style.display = "block";
10   } else {
11     mybutton.style.display = "none";
12   }
13 }
14
15 function topFunction() {
16   document.body.scrollTop = 0;
17   document.documentElement.scrollTop = 0;
18 }
```

CONFIGURE

Metrics

There are 3 functions in this file.

Function with the largest signature take 0 arguments, while the median is 0.

Largest function has 3 statements in it, while the median is 2.

The most complex function has a cyclomatic complexity value of 3 while the median is 1.

One unused variable

15 topFunction

topFunction variable is referenced on base.html:

```
<!-- Back to top button -->
<button onclick="topFunction()" id="myBtn" title="Go to top" aria-label="Return to top button">Top</button>
```

stripe_elements.js

```
1 /*
2   Core logic/payment flow for this comes from here:
3   https://stripe.com/docs/payments/accept-a-payment
4
5   CSS from here:
6   https://stripe.com/docs/stripe-js
7 */
8
9 var stripePublicKey = $('#id_stripe_public_key').text().slice(1, -1);
10 var clientSecret = $('#id_client_secret').text().slice(1, -1);
11 var stripe = Stripe(stripePublicKey);
12 var elements = stripe.elements();
13 var style = {
14   base: {
15     color: '#000',
16     fontFamily: '"Helvetica Neue", Helvetica, sans-serif',
17     fontSmoothing: 'antialiased',
18     fontSize: '16px',
19     '::placeholder': {
20       color: '#aab7c4'
21     }
22   },
23   invalid: {
24     color: '#dc3545',
25     iconColor: '#dc3545'
26   }
27 }
```

CONFIGURE

Metrics

There are 5 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 5.

The most complex function has a cyclomatic complexity value of 3 while the median is 1.

One undefined variable

11 Stripe

stripe variable connected to Stripe payments functionality .

update-qty.js

```
1 // Disable +/- buttons outside 1-99 range
2 function handleEnableDisable(itemId) {
3   var currentValue = parseInt($('#id_qty_' + itemId).val());
4   var minusDisabled = currentValue < 2;
5   var plusDisabled = currentValue > 98;
6   $('#decrement-qty_' + itemId).prop('disabled', minusDisabled);
7   $('#increment-qty_' + itemId).prop('disabled', plusDisabled);
8 }
9
10 // Ensure proper enabling/disabling of all inputs on page load
11 var allQtyInputs = $('.qty_input');
12 for(var i = 0; i < allQtyInputs.length; i++){
13   var itemId = $(allQtyInputs[i]).data('item_id');
14   handleEnableDisable(itemId);
15 }
16
17 // Update quantity on click
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 = '/basket/remove/' + itemId + '/';
28   var data = {'csrfmiddlewaretoken': csrfToken,};
29
30   $.post(url, data)
31   .done(function() {
32     location.reload();
33   });
34 });
```

CONFIGURE

Metrics

There are 4 functions in this file.

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

Largest function has 5 statements in it, while the median is 3.5.

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