

User could buy two digital photographs but it would only let you download one, once. This would obviously create unhappy customers. This could only be done by editing your basket to then give two of the same digital photo.

The code responsible for handling the condensing of the basket for repeat items is below. It looks and is correct:

```
def condensing_basket(request):
    basket = request.session.get('basket', {})
    if basket != {}:
        list_of_items_to_check = []
        duplicates_found = []
        for item in basket['items']:
            appending_item = [item['basket_item_id'], int(item['quantity']), [item['item_id'], item['digital_download'],
                                                                           item['linked_products']]]
            list_of_items_to_check.append(appending_item)

        for i in range(0, len(list_of_items_to_check)):
            for j in range(i+1, len(list_of_items_to_check)):
                if list_of_items_to_check[i][2] == list_of_items_to_check[j][2]:
                    print("duplicate found")
                    print(list_of_items_to_check[j])
                    print(list_of_items_to_check[i])
                    duplicates_found.append([list_of_items_to_check[i], list_of_items_to_check[j]])
                    print(duplicates_found)

        for k in range(len(duplicates_found)):
            for item in basket['items']:
                if item['digital_download'] is None:
                    if item['basket_item_id'] == duplicates_found[k][0][0]:
                        item['quantity'] = item['quantity'] + duplicates_found[k][1][1]
                        request.session['basket'] = basket
                        print("new item quantity")
                        print(item['quantity'])

        for k in range(len(duplicates_found)):
            print(len(duplicates_found))
            item_number = -1
            for item in basket['items']:
                item_number = item_number + 1
                print(item_number)
                if item['basket_item_id'] == duplicates_found[k][1][0]:
                    del basket['items'][item_number]
                    request.session['basket'] = basket
            return redirect(checkout)
```

However, the digital photos would not condense, leaving two of the same type on the checkout screen. So the error must be in identifying whether or not it was digital (the highlighted line). I worked out that for some reason, the "digital_download" must have a different value when added to a basket and when edited in the basket.

```
# Updating the digital download if necessary
if product.digital_download is True:
    if updated_delivery == "True":
        item['digital_download'] = 'on'
        item['quantity'] = 1
        messages.warning(request, "Digital purchases are restricted to a quantity of 1.")
    else:
        item['digital_download'] = None
```

Figure 1. This image shows that item['digital_download'] has a value of True when edited in the basket.

```
[28/Jan/2021 22:30:21] "GET /products
Printing digital download on
<class 'str'>
on
```

Figure 2. This image shows that item['digital_download'] has a value of 'on' when initially adding to the basket.

I changed the value in the edit_basket function to match the value when a product is initially added to the basket and it now condenses all images that are duplicates.