Composition with Components

The spirit of web components is to allow encapsulation so that our site is easier to maintain and easier to extend. In fact, we have the opportunity to do that right now. If you look at the main page you'll see a link for "Ship Orders" but on our main page, we already have a list of orders to ship. Aha! We have an opportunity for re-use!

1. Open a command window and use the Angular CLI to generate a new component called ListOfOrders. It should exist in the shipping folder. (Hint: use the --flat flag and specify the folder as part of the name ie. shipping/ListOfOrders)
2. Edit Dashboard.component.html and add a tag to show the list-of-orders component.
3. Run and test. If you can see your new component hosted inside the dashboard, you've just implemented composition.
4. Now edit dashboard.component.html and find the the html that creates the list of orders. Cut it from there and paste it into your new list-of-orders.component.html.
5. Run and test. You won't see any orders lists because the "orders" property is defined in DashboardComponent but is trying to be displayed in ListOfOrders. We need to pass those orders into the ListOfOrders component.

Property binding between components

6. Open ListOfOrders and create a new property called orders. Mark it with an "@Input" annotation. (Hint: In order for this to work, you're going to have to import *Input* from angular/core).
7. Now to pass it in the HTML. Open dashboard.component.html. Find where you've included the tag to list-of-orders. Add this property binding to it:
`[orders]="orders"`
8. This says to take the orders property from the host component and pass it down into the inner component as orders. (They happen to have the same name but they could be different if you prefer).
9. Run and test again. You should see the list in your dashboard again.

Cool. What we've done will make our app more modular and easier to understand. But we now have an opportunity for re-use.

Reusing a component

Notice that the AppComponent has a menu choice at the top called "Ship Orders", We really want to have that menu choice bring up a List of orders which are ready to ship and then allow the user to click on one to see the details for that one order. But we don't have a component for shipping multiple orders, only the single ship-order component. Let's create it.

10. Use Angular CLI to generate another component called OrdersToShip. Again, put it in the shipping folder.
11. Change OrdersToShip to be your startup component.
12. Add a nice header to your component and a paragraph telling the user to click on one of the orders to see its detail. The message and layout are up to you.
13. Look in the DashBoardComponent and copy the creation of this.orders to OrdersToShip.
14. Edit orders-to-ship.component.html and add a tag to display the list-of-orders component. Don't forget your property binding.

15. Run and test. Once the list of orders is available, you've successfully used and re-used a component.
16. Bonus! Look in setup/assets/html for a file called list-of-orders.css. This file has styles for your new component. Copy it and make it work.

Bonus Goal: The ship-order component. Make the lines be a new sub component. When the user checks the ready box, change its color..