

# 07.01.01: Static Product Assignment

## Description

During this iteration you will build a static website, meaning that the content, images etc. does not change and there's no integration with a database. Your assignment in the coming days will then be to continue working on it, fetching all the products from the API, making it dynamic (i.e. fetching data from a database).

So in short: **Design and build a site listing the products for the Product Shop**

You can see some of the data in [this overview](#)

In this iteration you shouldn't display all products, but choose 3-4, and use those as you design. Keep in mind that the rest of the products will be added later. Make room for these additional products, and keep in mind that you don't know exactly how many you'll have.

Remember to take into account (and present visually) that the products can

- be sold out
- have a discount price

Create your design so that all of these differences can be shown.

Hint: Make a class for each possibility that you add to the product.

## UI requirements:

Use CSS grid and optionally flexbox to layout the site.

It should be possible to click a product and read more, you can do that in any way you see fit. It could be another HTML page, it could be a "modal". When you implement this, just do it for a single product, as we will turn this into a feature that can display ANY product.

## Categories

The list of products should be sorted into categories. You don't have to add all the products in all the categories, as long as you have the basic skeleton for categories and products.

Possible choices for categories include:

- Gender
- Category
- Subcategory
- Articletype
- Season
- Usagetype
- Brandname

## Product details

How you show the details is up to you, but you should as a minimum show the following:

- Name
- Image
- Price
- Sold out status
- Discount price

It's up to you to decide how to do this.

## Hints and good ideas

Remember to make your design ready for the dynamic part, make the products as similar as possible and make a mental note of the differences.

Use as many relevant classes as possible, a soldOut class for products that are... sold out, a discounted class for products that are on a discount etc.

Add comments for yourself, for instance if there's an element that should be removed in some circumstances.

## Delivery

This iteration is not a hand-in, but rather something you will continue working on after the hand-in.

Have it ready by Thursday morning.

## Ressources

- [The sample data](#)
- [The API](#)

## Feedback

We'll see a few, randomly chosen, in class Thursday morning