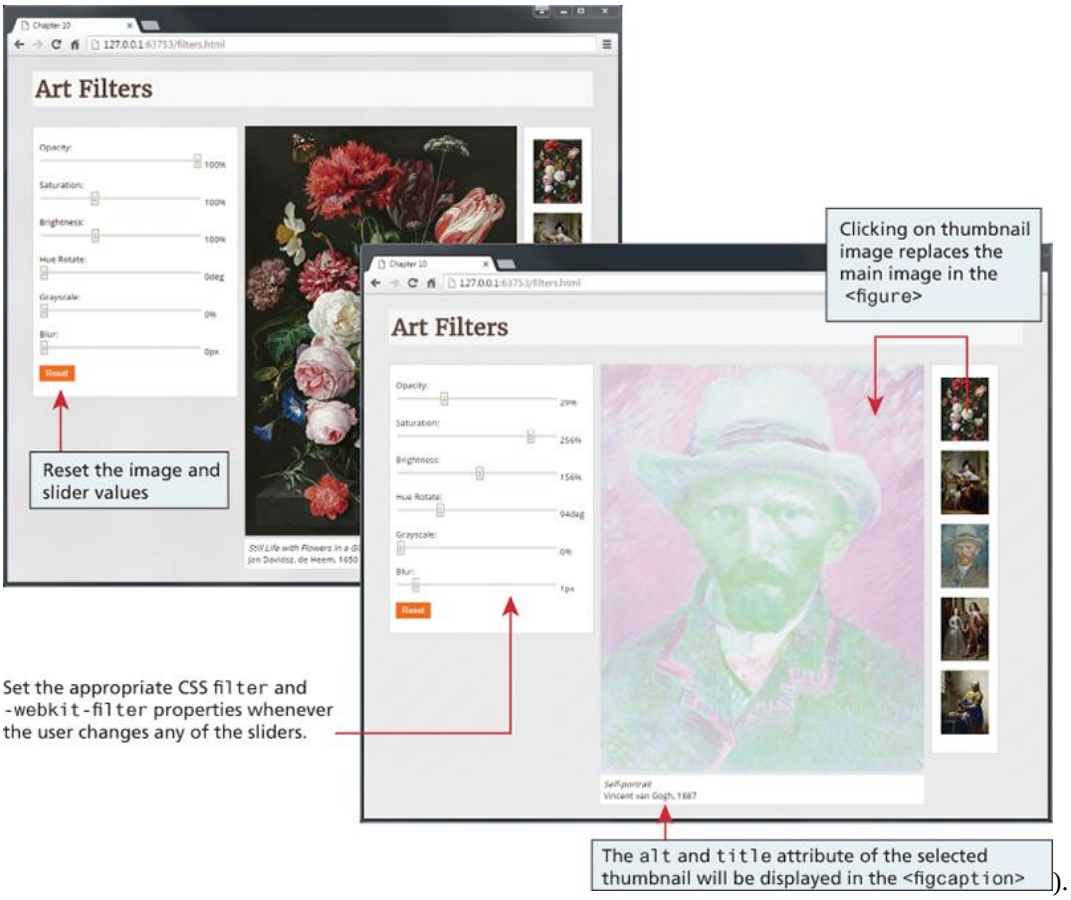
Enhancing Web Interfaces Using Filters

In this exercise, you will learn about how to use filters in CSS. Filters enable you to extend the graphical effects of HTML elements such as blueness, brightness, contrast, color shifting, among others. Please visit this link for more details on the filter functions in CSS.

There are two important limitations that you need to take into account when working with filters in CSS. The CSS filters are currently made available through the WebKit project. The WebKit project was initially a browser engine developed by Apple which enabled features such as user clicks, back-forward navigation, managing history, among others. More features were added to support advanced graphical effects and media features of HTML5. Nearly all modern browsers support the filter property. However, some browsers (e.g. Internet Explorer, Edge 12 and Safari 5.1 or earlier) do not support the filter property or may not support all of the features of the webkit-filter. In the sample HTML file of this exercise, three is a reference to both filter and webkit-filter to account for these variations. Therefore, when testing graphical effects filters when developing web application interfaces, it is strongly recommended that you as a developer test using multiple browsers to ensure consistency.

An Art Store

Use jQuery to respond to events and to programmatically modify HTML and CSS as shown in Figure 1.



).

Instructions:

1. Examine **exercise2.html** in the browser and then editor. You have been supplied with the necessary CSS, images and HTML on Canvas (Exercises → Exercise 2 → **exercise2-files.zip**).
2. Use jQuery to implement this exercise. To this extent, import jQuery in the **<head>** of the page.
3. Use jQuery to respond to click events on the painting thumbnails. Replace the **src** attribute of the **<img>** element in the **<figure>** so that it is displaying the clicked painting. Hint: get the **src** attribute of the clicked element and then replace the **small** folder name with **medium** folder name.
4. As well, change the **<figcaption>** so that it displays the newly clicked painting's title and artist information. This information is contained within the **alt** and **title** attributes of each thumbnail.
5. Set up event listeners for the **input** event of each of the range sliders. The code is going to set the **filter** and the **-webkit-filter** properties on the image in the **<figure>**. Recall from Module 3 that if you are setting multiple filters, they have to be included together separated by spaces.
6. Add a listener for the click event of the reset button. This will simply remove the filters from the image.
7. The image sizes vary in terms of width. Hence, ensure that all images are displayed consistently with a fixed size. You may use a size of 450x640 for displaying all medium-sized images for consistency.

Deliverable (upload to Canvas → exercises → exercise 2) the following files:

☐	Compress all of the files for the exercise 2 into zip format. Call this file **userid_exercise2.zip**. The compressed file should include the same structure as that of the one you have downloaded with the additions you have applied throughout completing the exercise.

Grading (10 marks)

[1 mark] page loads correct with very similar layout and design as in the mockup (including image orientation, slider control orientation, page background)
[1 mark] page loads with a default slider configuration automatically and this configuration is applied to default loaded image correctly
[6 marks] slider image controls are correctly applied for all required filters
[1 mark] reset button resets to default configuration (no need to reset image)
[1 mark] sliders' text values are updated correctly as slider is modified or changed

Additional Helpful Resources:

- CSS filter Property: https://www.w3schools.com/cssref/css3_pr_filter.asp
- CSS Filter Online Example: https://www.w3schools.com/cssref/tryit.php?filename=trycss3_filter_all