Dokumentasjon Prosjekt 1

# CSS Flexbox

In this project CSS Flexbox has been used to make a sensible and responsive layout on the website. Elements that are to be placed next to each other horizontally, are placed in the same container, which is then manipulated with “display: flex” and “flex-direction: row”, so that the layout wanted was achieved. The artwork and the titles of the art are placed in one container each, and as seen on the website, they appear alongside each other.

The rest of the layout is decided by the basic box model from CSS. Since the other elements only needed to be placed on top of each other, vertically, there was no need to use CSS Flexbox. When basing the rest of the layout on the basic CSS box model, the containers of the art and titles are behaving like the element to be placed vertically, not the elements within the container.

# HTML Canvas and SVG in HTML

The shapes and figures in SVG were computed in HTML by specifying the coordinates, heights, widths and so on. Some of them have styling in CSS as well. In SVG in HTML, each figure is treated like an object. They have their own ID and their own attributes. This made it easy to interact with the figures, because you can make a happening to be associated with an object.

With HTML Canvas, on the other hand, the Canvas is only a container where you can draw elements by specifying coordinates. The elements in the artwork each have their own draw function in JavaScript, but they have no ID or attributes. To add functionality to a single element, one would have to manually match the coordinates of the pointer with the coordinates of the drawn figure. This made it difficult to make i.e. a click function to only interact with a specified element. Therefore all the interaction with the artwork made in HTML Canvas happens to the entire container, and are not limited to the figures drawn.

# jQuery

jQuery has been used to add functionality and interactivity to the two pieces of art. With the figures in SVG, a function is associated with that element only using selectors. They are triggered by different events like hover, click or double click, and they cause different actions like color change, hide, show etc.

With the Canvas art, the selector of the functions is always the entire canvas container. Different events still trigger different actions, but the events are not limited to happen in a specified area or on a specified element in the canvas. Within the jQuery functions, basic JavaScript is used to make it seem like it is only one element that is triggered, by drawing a new figure on top of the other in another color.

# Cross-browser testing

For this project, the cross-browser testing was executed on Google Chrome and Safari. There was minimal noticeable difference, and all the interaction and functionality behaves like it should in both browsers.

# Tutorials and sources

Tutorials:

<https://scrimba.com/g/ghtmlcss>

<https://www.w3schools.com/jquery/default.asp>

<https://www.w3schools.com/html/html5_canvas.asp>

<https://www.w3schools.com/html/html5_svg.asp>

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/?fbclid=IwAR2UK2iaJfFeCTH6FuyxOfmvB03ibAAxdGqGdnTP-XT3UtuTK1d1BUTFajA>

Sources:

Ellipse in Canvas: <http://www.williammalone.com/briefs/how-to-draw-ellipse-html5-canvas/>

Rounded rectangle in Canvas: <https://stackoverflow.com/questions/1255512/how-to-draw-a-rounded-rectangle-on-html-canvas>