Web 2D Graphics Programming

HTML Canvas, SVG, Web Animation, Web Fonts, Image Element, Conversions, Coordinates, Files

This focused course explores all aspects of 2D graphics programming on the web. There are many 2D graphics options available to the modern web app developer; and they need to know which to choose for different sets of requirements.

This course covers in detail how to render onscreen & offscreen in 2D, create visual effects, download/upload image content, capture and animate visuals and how to handle fonts. We also look at user input and how to correctly respond to events (e.g. mouse moves) and how to handle screen refresh and window resizing.

Specific technologies covered include the HTML canvas element and its contexts, Canvas 2D graphical APIs, structured graphics with SVG, fonts with WOFF2, web animations using CSS and lots more.

We also investigate 2D graphics programming as part of larger application development and how to correctly structure how the graphical code interacts with other parts of the larger application – an issue often ignored until it is too late in the development process.

Target Audience

Experienced web developers seeking to programmatically create 2D graphics.

Prerequisites

Some background of graphical programming is mandatory, as is web programming experience.

All demo and lab code will be using Angular 8 and TypeScript, so experience of both of these is needed.

Note: This course covers 2D only – it does not cover 3D graphics on the web. The W3C is working on a new 3D standard for the web called WebGPU and its shader language called WHLSL. We plan to release a new course covering these nearer to when they are released.

Contents of One-Day Training Course

W3C Standards

The W3C offer a number of specs in the area of 2D graphics

Web developers have plenty of options but need to choose carefully depending on requirements

Tour of what is available

Introduction to each programming model

HTML Canvas Context

The <canvas> html element

Where a canvas can be contained and what it itself can contain

Canvas attributes (width, height, ..)

The idea of a rendering context

`CanvasRenderingContext2D`

Canvas Primitives

Graphical primitives: draw line, point, rect

Styles and strokes

Transform, scale, rotate, translate

Composing, path, text handling

Canvas Advanced

Offscreen bitmaps

`HtmlCanvasElement.OffscreenCanvas`

Programmatically extracting canvas data using `toBlob` and `toDataURl`

`ImageBitmapRenderingContext`

HTML Image

Images and the rest of a HTML page

`HTMLImageElement`: src, & srcset

JPEG and PNG formats

ALT and accessibility

Handling images on a web page

Interacting with images via advanced CSS

SVG Intro

Scalable Vector Graphics is just that XML-based

Unlike canvas (no file format), SVG is retained graphics with stream presentation

`SVGImageElement`

SVG in standalone file / HTML embedded

Drawing primitives and coordinate system

SVG Advanced

Styling with style attributes

Fills, strokes, effects, widths

Paths and line segments

Filtering

Advanced effects

HTML Web Animation

Relationship to CSS Transitions, CSS Animations and SVG

Configuring animations

Keyframes

Timelines

Angular 8's Web Animation

Use of CSS animatable properties

Enabling Angular 8's animation package

Transitions and timing in TypeScript

Triggers

Web Open Font Format (WOFF)

Typography on the web

WOFF - W3C spec to represent fonts

https://fonts.google.com/

Project

Building a larger project that integrates all of the 2D graphics programming capabilities explored in this course