

Updated on 08/24/2026

Sign Up

WebGPU Training

2 days (14 hours)

Overview

WebGPU is a modern API that harnesses the power of graphics cards directly from the browser to create high-performance 3D rendering and parallel computations. With WGSL, rendering pipelines, and compute shaders, you can build smoother and more ambitious web graphics experiences.

Our WebGPU training course equips you with the essential skills to develop interactive graphics applications based on modern GPU architecture. You'll learn how to initialize a WebGPU context, select an adapter, create a logical device, and organize the resources needed for rendering.

You'll master the creation of buffers, textures, bind groups, and pipelines, as well as the basics of the WGSL language. The course covers vertex and fragment shader programming to generate 2D and 3D scenes displayed on a canvas.

You'll also learn how to use compute shaders to parallelize processing, manage GPU commands, and structure an efficient rendering loop. The exercises emphasize architectural choices, code readability, and resource reuse.

By the end of the course, you will be able to design a reliable WebGPU prototype, identify hardware limitations, diagnose errors, and optimize data transfers between the CPU and the GPU. You will have the necessary foundation to integrate advanced graphics features into your web applications.

Like all our training courses, this one will introduce you to **the latest stable version** of the technology and its new features.

Objectives

- Understand the WebGPU architecture and programming model
- Initialize a GPU device and configure a rendering canvas
- Create and use buffers, textures, and bind groups
- Develop rendering shaders with WGSL
- Implement a compute pipeline using compute shaders
- Troubleshooting Errors and Optimizing Resources in a WebGPU Application

Target Audience

- Front-end developers
- JavaScript developers
- 3D developer
- Software engineer
- Front-end architect

Prerequisites

- Proficiency in the fundamentals of modern JavaScript
- Knowledge of HTML tags and CSS styling
- Experience with asynchronous functions and promises in JavaScript
- Basic understanding of 2D or 3D geometry, vectors, and matrices
- Experience developing with a browser and debugging tools

Technical Requirements

- Computer running Windows 11, macOS, or Linux with up-to-date graphics drivers
- Graphics card compatible with hardware acceleration; at least 8 GB of RAM (16 GB recommended)
- A recent WebGPU-compatible browser, such as Google Chrome or Microsoft Edge
- Visual Studio Code or an equivalent code editor
- Stable internet connection and authorized access to the browser's development tools

WebGPU Training Course Outline

[Day 1 - Morning]

WebGPU Fundamentals and Architecture

- WebGPU's role in the web graphics ecosystem and differences from WebGL
- Understanding the GPU model: adapter, logical device, command queue, and resources
- Exploring the core objects: GPUAdapter, GPUDevice, GPUQueue, and GPUCanvasContext
- Initializing WebGPU in a JavaScript application and handling incompatibility
- Configuring the canvas, rendering format, and rendering preparation cycle

- Hands-on workshop: Initializing a WebGPU project, detecting available capabilities, and displaying a configured canvas

[Day 1 - Afternoon]

First rendering pipeline and WGSL shaders

- Understanding the stages of a rendering pipeline: vertices, primitives, rasterization, and fragments
- Get familiar with WGSL syntax: types, functions, attributes, structures, and vectors
- Create a shader module and define the vertex and fragment entry points
- Create a GPURenderPipeline and implement a rendering pass
- Using vertex buffers to pass positions and colors to the graphics card
- Hands-on workshop: Program and display a colored triangle using a WGSL vertex shader and fragment shader

[Day 2 - Morning]

GPU resources, transformations, and textures

- Create and populate GPUBuffers according to their purpose: vertex, index, uniform, and storage
- Understanding memory alignment, data structures, and data exchange between JavaScript and WGSL
- Organizing resources with bind groups and bind group layouts
- Apply 2D and 3D transformations using vectors, matrices, and uniform buffers
- Load and utilize textures, samplers, and UV coordinates
- Hands-on workshop: rendering transformed and textured geometry from buffers and a bind group

[Day 2 - Afternoon]

GPU Computing, Diagnostics, and Optimization

- Identify use cases for compute shaders: massively parallel processing, simulation, and data preparation
- Building a GPUComputePipeline, defining workgroups, and triggering a dispatch
- Read the results of a GPU computation and synchronize buffers with the application
- Manage limits, optional features, error scopes, and device loss
- Optimizing performance: reducing CPU-GPU transfers, sharing resources, and measuring workloads
- Hands-on workshop: perform a parallel computation on the GPU, retrieve the result, and integrate it into the Processing in a demo application

Target Audience

This training is intended for both individuals and companies—large and small—

that wish to train their teams in new, advanced IT technology or to acquire specific business knowledge or modern methods.

Assessment at the Start of Training

The pre-training assessment complies with Qualiopi quality standards. Upon final registration, the learner receives a self-assessment questionnaire that allows us to evaluate their estimated proficiency in various types of technologies, as well as their expectations and personal goals for the upcoming training, within the limits imposed by the selected format. This questionnaire also allows us to anticipate certain connection or internal security issues within the company (intra-company or virtual classroom) that could pose challenges for monitoring and ensuring the smooth running of the training session.

Teaching Methods

Hands-On Training: 60% Practical, 40% Theoretical. Training materials distributed in digital format to all participants.

Organization

The course alternates between theoretical input from the instructor—supported by examples and reflection sessions—and group work.

Assessment

At the end of the session, a multiple-choice quiz is administered to verify that participants have successfully acquired the skills.

Certification

A certificate will be issued to each trainee who has completed the entire training program.