

Tutorials



During our [courses/trainings](#) we will teach you the best of what you can find here.]

We try to keep the following information as complete as possible, so please [contact](#) us if something is missing.

Learning OpenCL

[list1]

[Hands on OpenCL](#), by Simon McIntosh-Smith and Tom Deakin from the University of Bristol in the UK. It currently is the most up-to-date tutorial on OpenCL, including code for lab-sessions.]

- Bruno Jurkovski wrote a clear [quickstart](#).]

[AMD introduction to OpenCL](#).]

MacResearch playlist on Youtube. Code of episode [3](#) and [6](#). Zip of [PDFs](#).

[CMSoft's complete OpenCL tutorial](#).]

- The Code Project has a series on OpenCL, episodes [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#) and [8](#). By Rob Farber.]

[Dr.Dobb's](#) has a series called "CUDA, Supercomputing for the Masses". It is CUDA-oriented, but you can learn a lot about GPGPU in general and on NVIDIA specific optimisations. Login to their site and then you can access parts [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#), [15](#), [16](#), [17](#), [18](#), [19](#), [20](#) and [21](#). Registration is free.]

[AMD's university program](#). This is loads of information!]

[NVIDIA's OpenCL pages](#) provide all you need to program on NVIDIA.]

[Enjalot's adventures in OpenCL](#) giving the basics in OpenCL and pyOpenCL.]

[StreamHPC's basic concepts](#) with various tips&tricks on OpenCL.]

[KISTI Supercomputing Learning Centre](#) has a [beginners course for OpenCL](#). Material including [PDFs and code](#) is available on SF.net.]

[OpenCL cookbook](#) by Dhruva Bandopadhyay.]

- Anteru's introduction to OpenCL, part [#1](#), [#2](#) and [#3](#).]

[/list1]

OpenCL Optimisation guides

[Intel](#) Xeon and XeonPhi]

[NVidia](#) (CUDA, but same applies to OpenCL)]

[AMD](#) GPUs and CPUs]

[ARM](#) MALI T600]

[Altera](#) FPGAs]

Not available (yet):

- Imagination PowerVR
- Qualcomm Adreno
- Xilinx FPGAs

[infobox type="information"][widgets_on_pages id=Trainings][infobox]

University courses

OpenCL-based GPU-programming courses

[list1]

[Marcus Bannermann university course](#) made for the university of Erlangen, Germany.]

[Advanced Parallel Programming](#) is a course on parallel programming by professor John Cavazos of University of Delaware.]

[Programming for Performance](#) is a course on parallel programming by Jonathan Eyolfson of University of Waterloo.]

[Manchester OpenCL tutorial wiki](#). Materials from previous courses and more.]

[University of Innsbruck GPU-programming using OpenCL](#) by Juan J. Durillo PhD.]

[University of Waterloo Programming for Performance](#). Lecture notes and assignments.]

Architectures

[Berkely University Computer Architecture and Engineering](#)]

[/list1]

Videos

[list1]

[AMD's OpenCL introduction](#). Takes about an hour in total, slides are provided.]

[Harvard Lectures on GPGPU](#). One hour each.]

[/list1]

Cases/Studies

[list1]

[AMD optimisation case study](#): Diagonal Sparse Matrix Vector Multiplication .]

[AMD optimisation case study](#): Simple reductions.]

[/list1]

WebCL

WebCL is a new standard-to-be for OpenCL in the browser. Currently there are a few implementations, while Khronos is working on an official standard. WebCL is available on Firefox for [Linux32, Windows32 and Windows64](#) by Nokia. Also available for [Safari](#) on OSX by Samsung. A [Node.js-implementation](#) is made by Motorola. Examples made for another implementation will probably

not work.

Tutorials:

[list1]

[Nokia Research](#)]

[Paraplui](#)]

[/list1]

Check Khronos' [WebCL page](#) for more resources.

C/C++

Basic knowledge of C is needed to understand how to write kernels. Also many tutorials are in C++.

[list1]

[A little C primer.](#)]

[C++ for Java-programmers.](#)]

[C for Java-programmers.](#)]

<http://www.stanford.edu/class/cs101/> for if you have never programmed - don't think about GPU-programming yet.]

[/list1]

Basic OpenGL

Getting a grasp of OpenGL has advantages. Techniques for faster memory-operations in OpenGL have equivalents in OpenCL, giving reason to read on this subject.

[list1]

[Discussion](#) about OpenGL Shaders.]

[OpenGL Samples](#) from getting an empty screen till the famous teapot.]

[/list1]