# Fundamentals of Accelerated Computing with CUDA Python

You'll learn the fundamental tools and techniques for running GPU-accelerated Python applications in this workshop using CUDA and the NUMBA compiler GPUs.  You'll use a live, cloud-based GPU-enabled development environment to work though dozens of hands-on coding exercises. Learn how to:

- Write code for a GPU accelerator
- Configure code parallelization using the CUDA thread hierarchy
- Manage and optimize memory migration between the CPU and GPU accelerator
- Generate random numbers on the GPU
- Intermediate GPU memory management techniques.

finish by implementing your new workflow to accelerate a fully functional linear algebra program originally designed for CPUs to observe impressive performance gains.

After the workshop ends, you'll have additional resources enabling you to create new GPU-accelerated applications on your own.

| | |
|---|---|
| **Duration** | 8 hours |
| **Price** | $10,000 for up to 20-person groups. Each student gains dedicated access to a fully-configured GPU accelerated workstation |
| **Assessment type** | Code-based |
| **Certification** | Upon successful completion of the workshop, participants will receive NVIDIA DLI Certification to recognize subject matter competency and support professional career growth. |
| **Prerequisites** | Basic Python competency including familiarity with variable types, loops, conditional statements, functions, and array manipulations. NumPy competency including the use of ndarrays and ufuncs.  No previous knowledge of CUDA programming is required. |
| **Languages** | English |
| **Tools, Libraries, and Frameworks** | Numba, NumPy |

## Learning Objectives

Gain understanding on how to use fundamental tools and techniques for GPU-accelerate Python applications with CUDA and Numba, including:

- GPU-accelerate NumPy ufuncs with a few lines of code
- Write custom CUDA device kernels for maximum performance and flexibility
- Use memory coalescing and on-device shared memory to increase CUDA kernel bandwidth

## Why Deep Learning Institute Hands-on Training?

- Learn how build deep learning and accelerated computing applications across a wide range of industry segments such as autonomous vehicles, digital content creation, finance, game development, healthcare, and more.

# Fundamentals of Accelerated Computing with CUDA Python

- Learn in aguided, hands-on experience using the most widely-used, industry-standard software, tools, and frameworks.
- Gain real world expertise through content designed in collaboration with industry leaders such as the Children's Hospital of Los Angeles, Mayo Clinic, and PwC.
- Earn NVIDIA DLI certification to prove your subject matter competency and support professional career growth.
- Access content anywhere, anytime with a fully configured GPU-accelerated workstation in the cloud.

## Workshop Outline

| | **Components** | **Description** |
|---|---|---|
| **Introduction**<br>(15 mins) | ● Getting started | The first 15 minutes introduces  how to set-up your training environment. |
| **Introduction to CUDA Python with Numba**<br>(120 mins) | ● Optimize CPU code with the Numba compiler<br>● GPU-accelerate NumPy ufuncs<br>● Optimize host-to-device and device-to-host memory transfers | Begin to working with the Numba compiler and CUDA programming in Python. Learn to use Numba decorators to accelerate numerical Python functions. Complete an assessment to accelerate a neural network layer. |
| Break (60 mins) | | |
| **Custom CUDA Kernels in Python with Numba**<br>(120 mins) | ● Learn CUDA's parallel thread hierarchy<br>● Launch massively parallel custom CUDA kernels on the GPU<br>● Utilize atomic operations to avoid race conditions during parallel execution. | Learn how to extend parallel program possibilities,. including the ability to design and write flexible and powerful CUDA kernels. You'll grasp ho to easily handle race conditions with CUDA atomic operations and parallel thread synchronization. You'll also complete an assessment to accelerate a Mandelbrot set calculator and visualizer. |
| Break (15 mins) | | |
| **RNG, Multidimensional Grids, and Shared Memory for CUDA Python with Numba**<br>(90 mins) | ● Use xoroshiro128+ RNG to support GPU-accelerated monte carlo methods<br>● Learn multidimensional grid creation and how to work in parallel on 2D matrices<br>● Leverage on-device shared memory to communicate between threads | Generate a random number state for thousands of parallel threads in this intermediate-level module. Use shared memory for on-device caching and promoting memory coalescing while reshaping 2D matrices. |
| Break (15 mins) | | |
| **Assessment**<br>(30 mins) | ● Accelerate a CPU-only linear algebra subprogram | This module teaches you to leverage your learning to accelerate a CPU-only linear algebra subroutine for massive performance gains. |
| **Next Steps**<br>(15 mins) | ● Complete workshop survey<br>● Set up your own GPU enabled environment | Finally, learn how to set up your CUDA and GPU-enabled environment to begin work on your own projects. |

# Fundamentals of Accelerated Computing with CUDA Python

This content is also available as a self-paced online course at https://courses.nvidia.com/