CUDA Cheat Sheet

Cuda Bascics	cE_t **cudaMalloc** (void** devPtr, size_t size)	Allocates memory space on device
	cE_t **cudaFree** (void* devPtr)	Frees memory on device
	cE_t **cudaMemcpy** (void* dst, const void* src, size_t count, cudaMemcpyKind kind)	Copies data between host and device, like memcpy. kind = cudaMemcpy{HostToDevice|DeviceToHost|DeviceToDevice}
	__host__, __global__ void , __device__	Function annotation for host, kernel and device function
	cE_t **cudaDeviceSynchronize** (void)	Blocks Host until current device has finished
	Kernel<<<gridSize, blockSize>>>(*dev_ptrs*)	Call to kernel from host
	threadIdx.{x|y|z}, blockIdx.{x|y|z}, blockDim.{x|y|z}, gridDim.{x|y|z}	Available structures inside kernel
	int idx = index_x + index_y * gridDim.x * blockDim.x	2D indexing inside kernel
Pinned Memory	cE_t **cudaMallocHost** (void** ptr, size_t size)	Allocates page-locked (pinned) memory on the host
	cE_t **cudaFreeHost** (void* ptr)	Frees page-locked memory.
	cE_t **cudaHostRegister** (void* ptr, size_t size, unsigned int flags)	Registers an existing host memory range for use by CUDA
	cE_t **cudaHostUnregister** (void* ptr)	
	cE_t **cudaHostAlloc** (void** pHost, size_t size, unsigned int flags)	Allocates page-locked memory on the host with flags (for mapped memory)
	cE_t **cudaHostGetDevicePointer** (void** pDevice, void* pHost, unsigned int flags)	Passes back device pointer of mapped and pinned host memory
Streams	cE_t **cudaStreamCreate** (cudaStream_t* pStream)	Creates an asynchronous stream
	cE_t **cudaStreamDestroy** (cudaStream_t stream)	Destroys and cleans up an asynchronous stream
	cE_t **cudaMemcpyAsync** (void* dst, const void* src, size_t count, cudaMemcpyKind kind, cudaStream_t stream = 0)	Copies data between host and device asynchronously. Memory must be page-locked!
	Kernel<<<girdSize, blockSize, dS, stream>>>	Launch Kernel in stream
	cE_t **cudaStreamSynchronize** (cudaStream_t stream)	Wait for specific stream
Events	cE_t **cudaEventCreate** (cudaEvent_t* event)	Creates an event. Use, cudaEventcreateWithFlags to disable timing
	cE_t **cudaEventDestroy** (cudaEvent_t event)	Destroys an event object
	cE_t **cudaEventRecord** (cudaEvent_t event, cudaStream_t stream = 0)	Set event state to « not occurred »
	cE_t **cudaEventSynchronize** (cudaEvent_t event)	Waits until an event has occurred
Managed	cE_t **cudaMallocManaged** (void** devPtr, size_t size, unsigned int flags = cudaMemAttachGlobal)	Allocates memory that will be automatically managed by the Unified Memory system.
	cE_t **cudaMemPrefetchAsync** (const void* devPtr, size_t count, int dstDevice, cudaStream_t stream = 0)	Prefetches memory to the specified destination device. Use cudaCpuDeviceId to migrate from Device to Host. Use cudaGetDevice(int *device) to get device ID.
Graphs	cE_t **cudaStreamBeginCapture** (cudaStream_t stream	Begins graph capture on a stream.
	cE_t **cudaStreamEndCapture** (cudaStream_t stream, cudaGraph_t* pGraph)	Ends capture on a stream, returning the captured graph.
	cE_t **cudaGraphInstantiate** (cudaGraphExec_t* e, cudaGraph_t graph, …)	Creates an executable graph from a graph. (Not all parameters shown.) Launch with (instance, graph, NULL, NULL, 0)
	cE_t **cudaGraphLaunch** (cudaGraphExec_t graphExec, cudaStream_t stream)	Launches an executable graph in a stream

This cheat sheet is meant to accompany the CUDA programming course. Cuda 10.0
cE_t means cudaError_t