Tensor Indexing API

Indexing a tensor in the PyTorch C++ API works very similar to the Python API. All index types such as `None` / `...` / integer / boolean / slice / tensor are available in the C++ API, making translation from Python indexing code to C++ very simple. The main difference is that, instead of using the `[]`-operator similar to the Python API syntax, in the C++ API the indexing methods are:

- `torch::Tensor::index` (link)
- **System Message: WARNING/2 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\cpp\source\notes\(pytorch-master)(docs)(cpp)(source)(notes)tensor_indexing.rst`, line 2); *backlink***

  Duplicate explicit target name: "link".

  `torch::Tensor::index_put_` (link)

It's also important to note that index types such as `None` / `Ellipsis` / `Slice` live in the `torch::indexing` namespace, and it's recommended to put `using namespace torch::indexing` before any indexing code for convenient use of those index types.

Here are some examples of translating Python indexing code to C++:

Getter

Python	C++ (assuming `using namespace torch::indexing`)
`tensor[None]`	`tensor.index({None})`
`tensor[Ellipsis, ...]`	`tensor.index({Ellipsis, "..."})`
`tensor[1, 2]`	`tensor.index({1, 2})`
`tensor[True, False]`	`tensor.index({true, false})`
`tensor[1::2]`	`tensor.index({Slice(1, None, 2)})`
`tensor[torch.tensor([1, 2])]`	`tensor.index({torch::tensor({1, 2})})`
`tensor[..., 0, True, 1::2, torch.tensor([1, 2])]`	`tensor.index({"...", 0, true, Slice(1, None, 2), torch::tensor({1, 2})})`

Setter

Python	C++ (assuming `using namespace torch::indexing`)
`tensor[None] = 1`	`tensor.index_put_({None}, 1)`
`tensor[Ellipsis, ...] = 1`	`tensor.index_put_({Ellipsis, "..."}, 1)`
`tensor[1, 2] = 1`	`tensor.index_put_({1, 2}, 1)`
`tensor[True, False] = 1`	`tensor.index_put_({true, false}, 1)`
`tensor[1::2] = 1`	`tensor.index_put_({Slice(1, None, 2)}, 1)`
`tensor[torch.tensor([1, 2])] = 1`	`tensor.index_put_({torch::tensor({1, 2})}, 1)`
`tensor[..., 0, True, 1::2, torch.tensor([1, 2])] = 1`	`tensor.index_put_({"...", 0, true, Slice(1, None, 2), torch::tensor({1, 2})}, 1)`

Translating between Python/C++ index types

The one-to-one translation between Python and C++ index types is as follows:

Python	C++ (assuming `using namespace torch::indexing`)
`None`	`None`
`Ellipsis`	`Ellipsis`
`...`	`"..."`
`123`	`123`
`True`	`true`
`False`	`false`
`:` or `::`	`Slice()` or `Slice(None, None)` or `Slice(None, None, None)`
`1:` or `1::`	`Slice(1, None)` or `Slice(1, None, None)`
`:3` or `:3:`	`Slice(None, 3)` or `Slice(None, 3, None)`
`::2`	`Slice(None, None, 2)`
`1:3`	`Slice(1, 3)`
`1::2`	`Slice(1, None, 2)`
`:3:2`	`Slice(None, 3, 2)`
`1:3:2`	`Slice(1, 3, 2)`

Python	C++ (assuming `using namespace torch::indexing`)
`torch.tensor([1, 2])`	`torch::tensor({1, 2})`