Putting it all together (PyTorch)

Install the Transformers, Datasets, and Evaluate libraries to run this notebook.

```
[ ] !pip install datasets evaluate transformers[sentencepiece]
```

```
[ ] from transformers import AutoTokenizer

    checkpoint = "distilbert-base-uncased-finetuned-sst-2-english"
    tokenizer = AutoTokenizer.from_pretrained(checkpoint)

    sequence = "I've been waiting for a HuggingFace course my whole life."

    model_inputs = tokenizer(sequence)
```

```
[ ] sequence = "I've been waiting for a HuggingFace course my whole life."

    model_inputs = tokenizer(sequence)
```

```
[ ] sequences = ["I've been waiting for a HuggingFace course my whole life.", "So have I!"]

    model_inputs = tokenizer(sequences)
```

```
[ ] # Will pad the sequences up to the maximum sequence length
    model_inputs = tokenizer(sequences, padding="longest")
```

```
# Will pad the sequences up to the model max length
# (512 for BERT or DistilBERT)
model_inputs = tokenizer(sequences, padding="max_length")

# Will pad the sequences up to the specified max length
model_inputs = tokenizer(sequences, padding="max_length", max_length=8)
```

```
[ ] sequences = ["I've been waiting for a HuggingFace course my whole life.", "So have I!"]

    # Will truncate the sequences that are longer than the model max length
    # (512 for BERT or DistilBERT)
    model_inputs = tokenizer(sequences, truncation=True)

    # Will truncate the sequences that are longer than the specified max length
    model_inputs = tokenizer(sequences, max_length=8, truncation=True)
```

```
[ ] sequences = ["I've been waiting for a HuggingFace course my whole life.", "So have I!"]

    # Returns PyTorch tensors
    model_inputs = tokenizer(sequences, padding=True, return_tensors="pt")

    # Returns TensorFlow tensors
    model_inputs = tokenizer(sequences, padding=True, return_tensors="tf")

    # Returns NumPy arrays
    model_inputs = tokenizer(sequences, padding=True, return_tensors="np")
```

```
[ ] sequence = "I've been waiting for a HuggingFace course my whole life."

    model_inputs = tokenizer(sequence)
    print(model_inputs["input_ids"])

    tokens = tokenizer.tokenize(sequence)
    ids = tokenizer.convert_tokens_to_ids(tokens)
    print(ids)
```

```
[101, 1045, 1005, 2310, 2042, 3403, 2005, 1037, 17662, 12172, 2607, 2026, 2878, 2166, 1012, 102]
```

```
[1045, 1005, 2310, 2042, 3403, 2005, 1037, 17662, 12172, 2607, 2026, 2878, 2166, 1012]
```

```
[ ]  print(tokenizer.decode(model_inputs["input_ids"]))
     print(tokenizer.decode(ids))
```

```
"[CLS] i've been waiting for a huggingface course my whole life. [SEP]"
"i've been waiting for a huggingface course my whole life."
```

```
[ ]  import torch
     from transformers import AutoTokenizer, AutoModelForSequenceClassification

     checkpoint = "distilbert-base-uncased-finetuned-sst-2-english"
     tokenizer = AutoTokenizer.from_pretrained(checkpoint)
     model = AutoModelForSequenceClassification.from_pretrained(checkpoint)
     sequences = ["I've been waiting for a HuggingFace course my whole life.", "So have I!"]

     tokens = tokenizer(sequences, padding=True, truncation=True, return_tensors="pt")
     output = model(**tokens)
```