

## End-of-chapter quiz

Let's test what you learned in this chapter!

### 1. What can you use Gradio to do?

- ☒ Create a demo for your machine learning model

**Correct!** With a few lines of python code you can generate a demo for your ML model using our library of pre-built components.

- ☒ Share your machine learning model with others

**Correct!** Using the `share=True` parameter in the launch method, you can generate a share link to send to anyone.

- ☒ Debug your model

**Correct!** One advantage of a gradio demo is being able to test your model with real data which you can change and observe the model's predictions change in real time, helping you debug your model.

- ☐ Train your model

You got all the answers!

### 2. Gradio ONLY works with PyTorch models

- ☐ True
- ☒ False

**Correct!** Gradio is model agnostic, meaning you can create a demo for any type of machine learning model.

You got all the answers!

### 3. Where can you launch a Gradio demo from?

- ☒ Standard python IDEs

**Correct!** Gradio works great with your favorite IDE.

- ☒ Google Colab notebooks

**Correct!** You can create and launch a demo within your Google colab notebook.

- ☒ Jupyter notebooks

**Correct!** Good choice - You can create and launch a demo within your Jupyter notebook.

You got all the answers!

### 4. Gradio is designed primarily for NLP models

- ☐ True
- ☒ False

**Correct!** Gradio supplies developers with a library of pre-built components for pretty much all data types.

You got all the answers!

### 5. Which of the following features are supported by Gradio?

- ☒ Multiple inputs and outputs

**Correct!** Multiple inputs and outputs is possible with gradio. All you need to do is pass in a list of inputs and outputs to their corresponding parameters

- ☒ State for data persistence

**Correct!** Gradio is capable of adding state to your interface.

- ☒ Username and passwords authentication

**Correct!** Pass in a list of username/password tuples to the launch method to add authentication.

- ☐ Automatic analytics for who uses your gradio demo
- ☒ Loading a model from Hugging Face's model hub or Hugging Face Spaces

**Correct!** Absolutely - load any Hugging Face model using the `gr.Interface.load()` method

You got all the answers!

### 6. Which of the following are valid ways of loading a Hugging Face model from Hub or Spaces?

- ☒ `gr.Interface.load('huggingface/{user}/{model_name}')`

**Correct!** This is a valid method of loading a Hugging Face model from the Hub

- ☒ `gr.Interface.load('model/{user}/{model_name}')`

**Correct!** This is a valid method of loading a Hugging Face model from the Hub

- ☐ `gr.Interface.load('demos/{user}/{model_name}')`
- ☒ `gr.Interface.load('spaces/{user}/{model_name}')`

**Correct!** This is a valid method of loading a Hugging Face model from Spaces

You got all the answers!

### 7. Select all the steps necessary for adding state to your Gradio interface

- ☒ Pass in an extra parameter into your prediction function, which represents the state of the interface.

**Correct!** An extra parameter storing history or state of your interface is necessary.

- ☒ At the end of the prediction function, return the updated value of the state as an extra return value.

**Correct!** This history or state value needs to be returned by your function.

- ☒ Add the state input and state output components when creating your Interface

**Correct!** Gradio provides a state input and output component to persist data.

You got all the answers!

### 8. Which of the following are components included in the Gradio library?

- ☒ Textbox.

**Correct!** Yes, you can create textboxes with the Textbox component.

- ☐ Graph.
- ☒ Image.

**Correct!** Yes, you can create an image upload widget with the Image component.

- ☒ Audio.

**Correct!** Yes, you can create an audio upload widget with the Audio component.

You got all the answers!

### 9. What does Gradio Blocks allow you to do?

- ☒ Combine multiple demos into one web app

**Correct!** You can use the ``with gradio.Tabs():`` to add tabs for multiple demos

- ☒ Assign event triggers such as clicked/changed/etc to `Blocks` components

**Correct!** When you assign an event, you pass in three parameters: fn: the function that should be called, inputs: the (list) of input component(s), and outputs: the (list) of output components that should be called.

- ☒ Automatically determine which `Blocks` component should be interactive vs. static

**Correct!** Based on the event triggers you define, `Blocks` automatically figures out whether a component should accept user input or not.

- ☒ Create multi-step demos; meaning allowing you to reuse the output of one component as the input to the next

**Correct!** You can use a component for the input of one event trigger but the output of another.

You got all the answers!

### 10. You can share a public link to a Blocks demo and host a Blocks demo on Hugging Face spaces.

- ☒ True

**Correct!** Just like `Interface`, all of the sharing and hosting capabilities are the same for `Blocks` demos!

- ☐ False

You got all the answers!

[Update](#) on GitHub

### End-of-chapter quiz

1. What can you use Gradio to do?

2. Gradio ONLY works with PyTorch models

3. Where can you launch a Gradio demo from?

4. Gradio is designed primarily for NLP models

5. Which of the following features are supported by Gradio?

6. Which of the following are valid ways of loading a Hugging Face model from Hub or Spaces?

7. Select all the steps necessary for adding state to your Gradio interface

8. Which of the following are components included in the Gradio library?

9. What does Gradio Blocks allow you to do?

10. You can share a public link to a Blocks demo and host a Blocks demo on Hugging Face spaces.