

NEW YEAR
NEW SKILLS 50% OFFSale ends in
4d 19h 26m 00s X

TUTORIALS ▾

category ▾

[Home](#) > [Tutorials](#) > [Artificial Intelligence](#)

How to Use DeepSeek Janus-Pro Locally

Learn how to set up the DeepSeek Janus-Pro project, build your own Docker image, and run the Janus web application locally on your laptop.

 Contents

Jan 29, 2025 · 13 min read

**Abid Ali Awan**

Certified data scientist, passionate about building ML apps, blogging on data science, and editing.

TOPICS

Artificial Intelligence

Large Language Models

DeepSeek, a Chinese AI startup, has disrupted the global AI industry, wiping \$1 trillion from U.S. stock exchanges and shaking tech giants like Nvidia and OpenAI. It has quickly emerged as a leader in text generation, reasoning, vision models, and image generation. DeepSeek recently launched state-of-the-art models, including **Janus**, a multimodal model capable of understanding visual data and generating images from text.

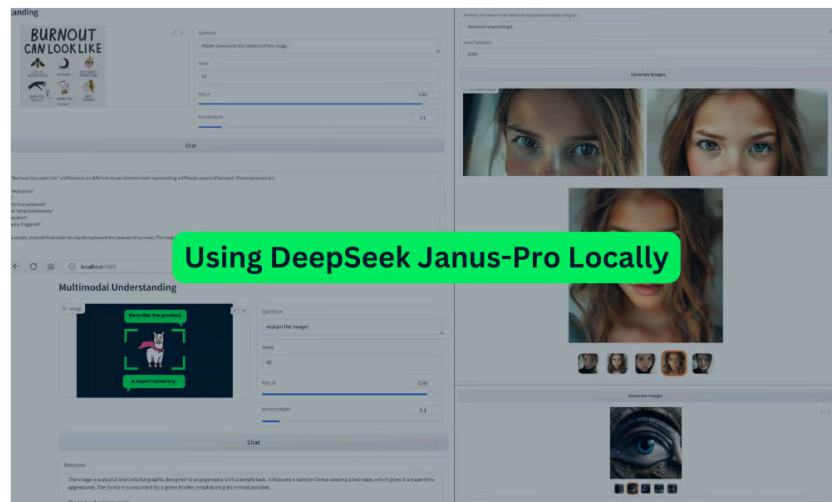


Image by Author

In this tutorial, we will learn about the Janus Series, set up the Janus project, build a Docker container to run the model locally, and test its capabilities with various image and text prompts.

Learn about DeepSeek models that are disrupting the global AI industry:

- [DeepSeek-V3: A Guide With Demo Project](#)
- [DeepSeek-R1: Features, a1 Comparison, Distilled Models & More](#)

What is the DeepSeek Janus-Series?

The DeepSeek Janus-Series is a new line of multimodal models designed to unify visual understanding and generation tasks using innovative frameworks. The series includes **Janus**, **JanusFlow**, and the advanced **Janus-Pro**, each building upon the previous iteration with notable improvements in efficiency, performance, and multimodal capabilities.

1. Janus

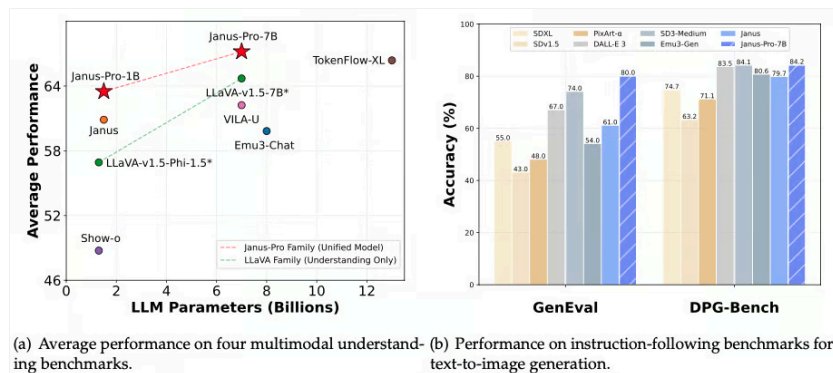
Janus is an innovative autoregressive framework that separates visual encoding into distinct pathways for understanding and generation tasks, while using a unified transformer architecture. This approach resolves conflicts between these roles, enhancing flexibility and effectiveness. Janus matches or surpasses task-specific models, making it a strong candidate for next-generation multimodal systems.

2. JanusFlow

JanusFlow combines autoregressive language modeling with rectified flow, a state-of-the-art generative modeling technique. Its minimalist design enables easy training within large language model frameworks without the need for complicated modifications. JanusFlow demonstrates superior performance on benchmarks when compared to both specialized and unified approaches, effectively pushing the boundaries of vision-language modeling.

3. Janus-Pro

Janus-Pro enhances the previous models by introducing optimized training strategies, expanded training datasets, and scaling to larger model sizes. These advancements significantly improve multimodal understanding and text-to-image instruction-following while ensuring more stable text-to-image generation.



Source: [deepseek-ai/Janus](https://deepseek-ai.com/janus)

Read [DeepSeek's Janus-Pro: Features, DALL-E 3 Comparison & More](#) to learn more about the Janus series, how to access it, and how it compares to OpenAI's DALL-E 3.

Setting up the Janus Project

Janus is a relatively new model, and currently, there are no quantized versions or local AI applications available for easy use on laptops or desktops.

However, the Janus repository on GitHub provides an excellent [Gradio](#) web application demo for anyone to try. The challenge, however, is that the demo often fails to work due to multiple package conflicts.

To overcome this issue, in this project, we will modify the original code, build our own Docker image, and run the container locally using the Docker Desktop application.


1. Install Docker Desktop

To get started, download and install the latest version of Docker Desktop from the [official Docker website](#).

Note for Windows users:

If you are using Windows, you will also need to install the Windows Subsystem for Linux (WSL). Open your terminal and run the following command to install WSL:

```
wsl --install
```

 Explain code


POWERED BY  datalab

2. Clone the Janus repository

Next, clone the Janus repository from GitHub and navigate to the project directory:

```
git clone https://github.com/deepseek-ai/Janus.git
cd Janus
```

 Explain code

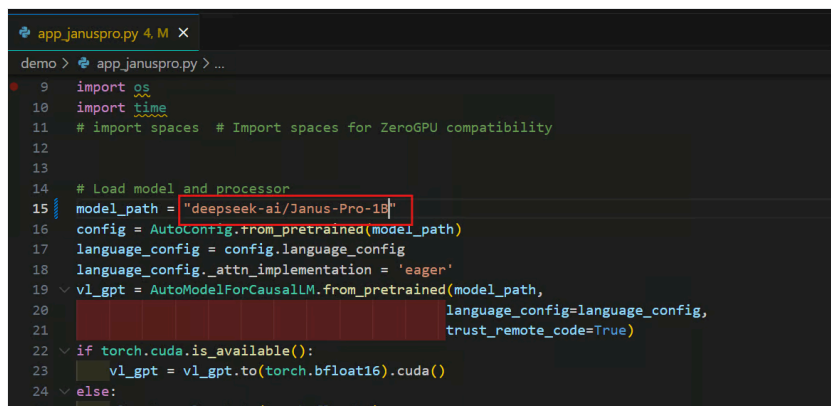
POWERED BY  datalab

3. Modify the demo code

Navigate to the demo folder and open the file `app_januspro.py` in your preferred code editor. Make the following changes:

1. Change the model name: Replace `deepseek-ai/Janus-Pro-7B` with `deepseek-ai/Janus-Pro-1B`.

This will load the lighter version of the model, which is only 4.1 GB in size, making it more suitable for local use.



```

9  import os
10 import time
11 # import spaces # Import spaces for ZeroGPU compatibility
12
13
14 # Load model and processor
15 model_path = "deepseek-ai/Janus-Pro-1B"
16 config = AutoConfig.from_pretrained(model_path)
17 language_config = config.language_config
18 language_config.attn_implementation = 'eager'
19 vl_gpt = AutoModelForCausalLM.from_pretrained(model_path,
20                                               language_config=language_config,
21                                               trust_remote_code=True)
22
23 if torch.cuda.is_available():
24     vl_gpt = vl_gpt.to(torch.bfloat16).cuda()
25 else:

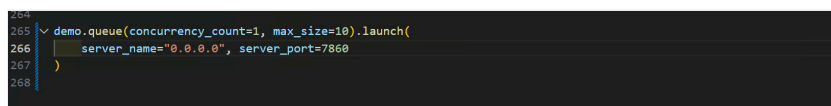
```

2. Update the last line of the script: Modify the `demo.queue` function to the following:

```
demo.queue(concurrency_count=1, max_size=10).launch(
    server_name="0.0.0.0", server_port=7860
)
```

 Explain code

POWERED BY  datalab



```

264
265 demo.queue(concurrency_count=1, max_size=10).launch(
266     server_name="0.0.0.0", server_port=7860
267 )
268

```

This change ensures compatibility with the Docker URL and port.

4. Create a Docker image

To containerize the application, create a `Dockerfile` in the root directory of the project. Use the following content for the `Dockerfile`:

```
# Use the PyTorch base image
FROM pytorch/pytorch:latest

# Set the working directory inside the container
WORKDIR /app

# Copy the current directory into the container
COPY . /app

# Install necessary Python packages
RUN pip install -e .[gradio]

# Set the entrypoint for the container to launch your Gradio app
CMD ["python", "demo/app_januspro.py"]
```

[🔗 Explain code](#)
POWERED BY  datalab

The `Dockerfile` will:

- Create a Docker image using the PyTorch base image.
- It will set the working directory inside the container.
- All necessary files from your local machine will be copied to the `/app` directory within the Docker container.
- The Dockerfile will install all required dependencies for the application.
- Finally, it will launch the Gradio application within the container.

Building and Running the Docker Image

Once the `Dockerfile` is ready, we will build the Docker image and run the container.

You can take the [Introduction to Docker](#) course to learn about the basics of building and deploying a Docker image.

Type the following command in the terminal to create the Docker image. This command will use the `Dockerfile` located in the root folder and assign the name `janus` to the image:

```
docker build -t janus .
```

[🔗 Explain code](#)
POWERED BY  datalab

It takes 10 to 15 minutes, based on your internet speed, to build the image from scratch.

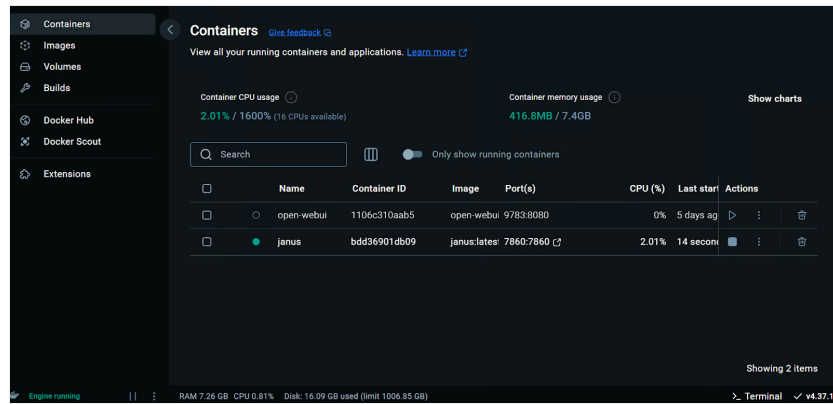
```
janus@janus:~$ docker build -t janus .
[+] Building 99.1s (5/9)                                docker:desktop-linux
=> [1/4] FROM docker.io/pytorch/pytorch:latest@sha256:11691e035a3651d25a87116b4f6adc113a27a29d8f5a6a583f8569e0ee5 95.5s
=> sha256:3d552c93e873f61562c7bc8d1e0a9436c890cb44d1a9c97c9a69b864dfb91277 139.46MB / 3.62GB 95.5s
=> sha256:3ad96c6a423ac845c1c88befd37f33e5cee0f9c46f6a8c8e7300de5420b19e9 7.23MB / 7.23MB 22.5s
=> extracting sha256:d6d6da6a368713979f9d09fad193991ae1af18b8ef3abfd70ade192887c1bd 0.8s
=> sha256:4f4fb700ef8461cfad2571ae0db9abdd1e0cd8357740b4a6d75e68dc30e8acc1 32B / 32B 17.1s
=> sha256:42a8919f2ed9b95dfec4eccc36b869098966254096e137778fbc94025efa338 99B / 99B 17.6s
=> extracting sha256:3ad96c6a423ac845c1c88befd37f33e5cee0f9c46f6a8c8e7300de5420b19e9 0.5s
=> [internal] load build context 0.4s
=> transferring context: 18.64MB 0.4s
```

The following command starts a `Docker` container for the Janus application. It enables GPU support, maps port 7860 for accessing the Gradio app, and ensures persistent storage for Hugging Face model files.

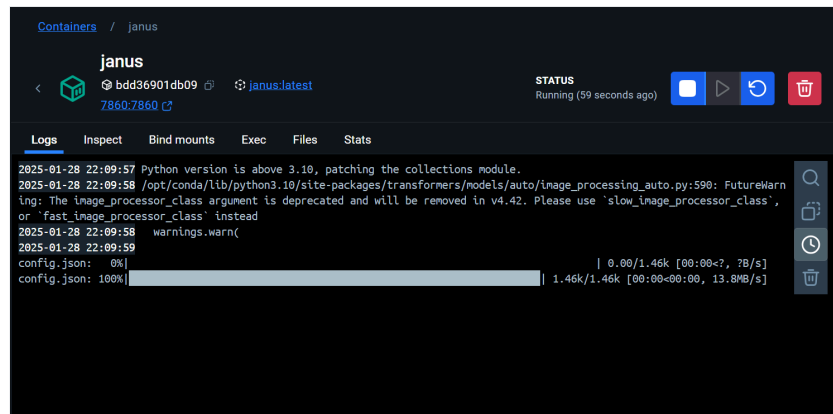
```
docker run -it -p 7860:7860 -d -v huggingface:/root/.cache/huggingface -w / --
```

[🔗 Explain code](#)
POWERED BY  datalab

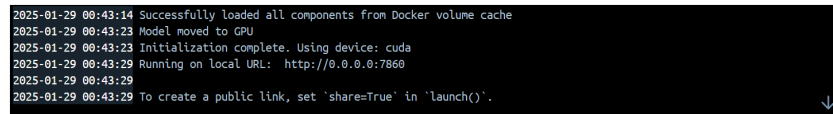
If you open the Docker Desktop application and navigate to the “Containers” tab, you will see that the `janus` container is running. However, it is not yet ready to use.



To check its progress, click on the `janus` container and then go to the “Logs” tab. Here, you will notice that the container is downloading the model file from the Hugging Face Hub.



Once the model has been successfully downloaded, the logs will display a message indicating that the application is running.



You can then access your application by visiting the following URL in your browser: `http://localhost:7860/`.

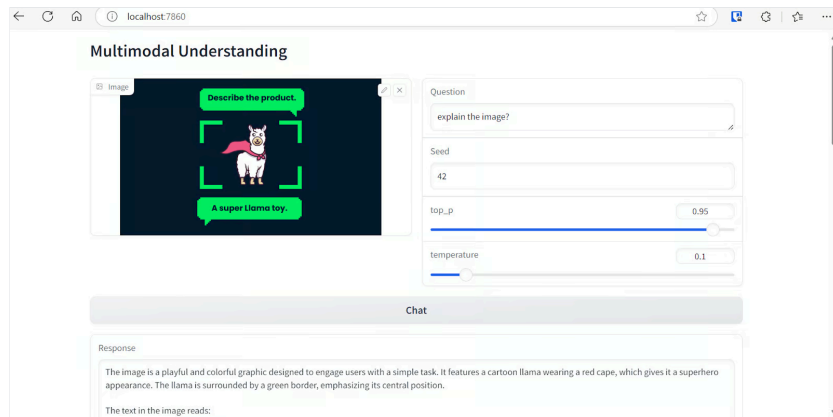
If you are experiencing issues, please check the updated version of the Janus project at [kingabzpro/Janus: Janus-Series](https://github.com/kingabzpro/Janus:Janus-Series).

Testing the Janus Pro Model

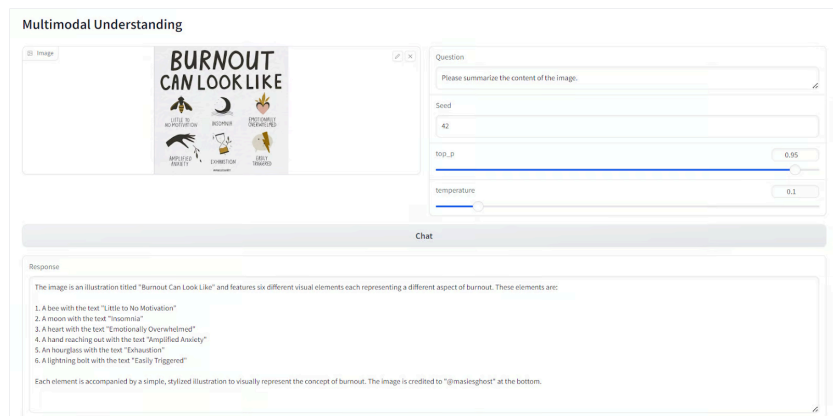
The web app has a clean interface, and everything appears to be working smoothly. In this section, we will test the Janus Pro's multimodal understanding and text-to-image generation capabilities.

Testing the multimodal understanding

To evaluate the model's multimodal understanding, we first load an image from the DataCamp tutorial and ask the model to explain it. The results are impressive—even with the smaller 1B model, the response is highly accurate and detailed.

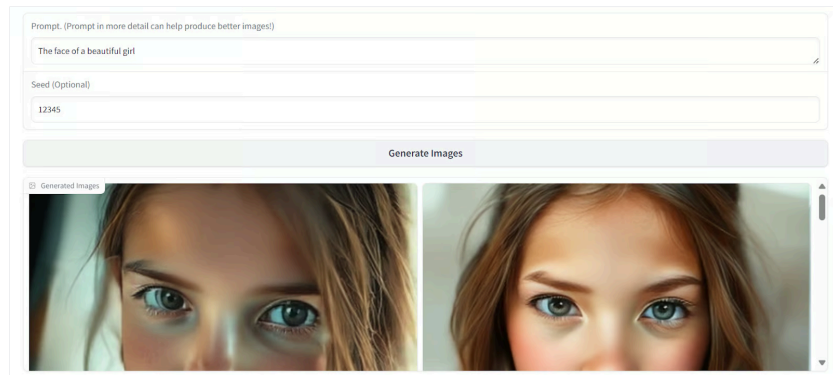


Next, we load another image and ask the model to summarize the content of an infographic. The model successfully understands the text within the image and provides a highly accurate and coherent response. This demonstrates the model's strong ability to process and interpret both visual and textual elements.



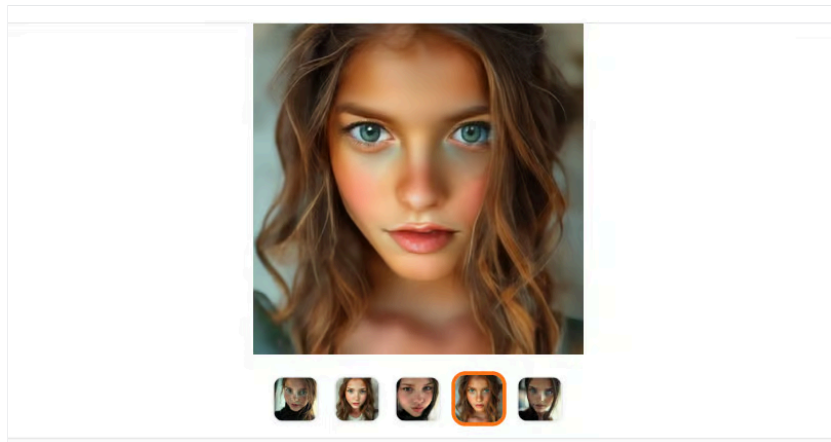
Testing text-to-image generation

Scrolling down the app, you'll find the "Text-to-Image Generation" section. Here, you can enter a prompt of your choice and click the "Generate Images" button. The model generates five variations of the image, which may take a few minutes to complete.



The results are remarkable, producing outputs that are comparable to Stable Diffusion XL in terms of quality and detail.

You can learn how to [Fine-tune Stable Diffusion XL with DreamBooth and LoRA](#) on your personal images.



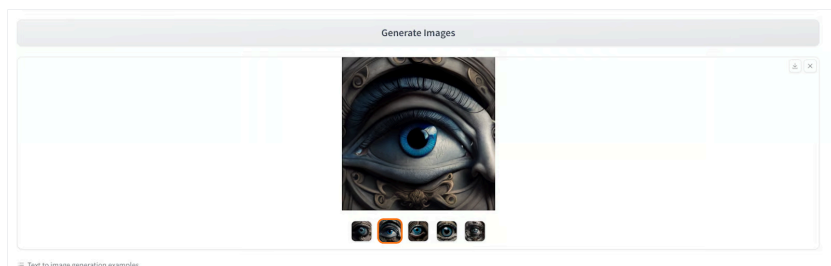
Let's try another prompt:

Prompt:

"The image features an intricately designed eye set against a circular backdrop adorned with ornate swirl patterns that evoke both realism and surrealism. At the center of attention is a strikingly vivid blue iris surrounded by delicate veins radiating outward from the pupil to create depth and intensity. The eyelashes are long and dark, casting subtle shadows on the skin around them, which appears smooth yet slightly textured as if aged or weathered over time.

Above the eye, there's a stone-like structure resembling part of classical architecture, adding layers of mystery and timeless elegance to the composition. This architectural element contrasts sharply but harmoniously with the organic curves surrounding it. Below the eye lies another decorative motif reminiscent of baroque artistry, further enhancing the overall sense of eternity encapsulated within each meticulously crafted detail.

Overall, the atmosphere exudes a mysterious aura intertwined seamlessly with elements suggesting timelessness, achieved through the juxtaposition of realistic textures and surreal artistic flourishes. Each component—from the intricate designs framing the eye to the ancient-looking stone piece above—contributes uniquely towards creating a visually captivating tableau imbued with enigmatic allure."



Once again, the results are stunning. The generated images capture the intricate details and surreal artistic elements described in the prompt.

Conclusion

To test the full capabilities of the model, DeepSeek has deployed the same version of the web application on Hugging Face Spaces. You can access it at [Chat With Janus-Pro-7B](#). One of the standout features of the Janus Pro model is its high accuracy, even when using smaller variants like the 1B model.

In this tutorial, we explored **Janus Pro**, a multimodal model capable of both understanding images and generating images from text prompts. Additionally, we built our own local solution to use the model privately on a laptop GPU. This setup is fast, efficient, and allows you to experiment freely with limited resources.

Check out our latest guide on [Fine-Tuning DeepSeek R1 \(Reasoning Model\)](#). You will learn to train the reasoning model on the medical chain of thought dataset to build better AI doctors for the future.



AUTHOR

Abid Ali Awan

As a certified data scientist, I am passionate about leveraging cutting-edge technology to create innovative machine learning applications. With a strong background in speech recognition, data analysis and reporting, MLOps, conversational AI, and NLP, I have honed my skills in developing intelligent systems that can make a real impact. In addition to my technical expertise, I am also a skilled communicator with a talent for distilling complex concepts into clear and concise language. As a result, I have become a sought-after blogger on data science, sharing my insights and experiences with a growing community of fellow data professionals. Currently, I am focusing on content creation and editing, working with large language models to develop powerful and engaging content that can help businesses and individuals alike make the most of their data.

TOPICS

Artificial Intelligence

Large Language Models

Top AI Courses

COURSE

Introduction to LLMs in Python

🕒 4 hr 🧑 12K

Learn the nuts and bolts of LLMs and the revolutionary transformer architecture they are based on!

[See Details →](#)[Start Course](#)[See More →](#)

Related



BLOG

DeepSeek's Janus Pro: Features, DALL-E 3 Comparison & More

BLOG

I Tested DeepSeek R1 Lite Preview to See if It's Better Than...



TUTORIAL

How to Set Up and Run DeepSeek R1 Locally With...

[See More →](#)

Grow your data skills with DataCamp for Mobile

Make progress on the go with our mobile courses and daily 5-minute coding challenges.



LEARN

[Learn Python](#)

[Learn AI](#)

[Learn Power BI](#)

[Learn Data Engineering](#)

[Assessments](#)

[Career Tracks](#)

[Skill Tracks](#)

[Courses](#)

[Data Science Roadmap](#)

DATA COURSES

[Python Courses](#)

[R Courses](#)

[SQL Courses](#)

[Power BI Courses](#)

[Tableau Courses](#)

[Alteryx Courses](#)

[Azure Courses](#)

[AWS Courses](#)

[Google Sheets Courses](#)

[Excel Courses](#)

[AI Courses](#)

[Data Analysis Courses](#)

[Data Visualization Courses](#)

[Machine Learning Courses](#)

[Data Engineering Courses](#)

[Probability & Statistics Courses](#)

DATALAB[Get Started](#)[Pricing](#)[Security](#)[Documentation](#)**CERTIFICATION**[Certifications](#)[Data Scientist](#)[Data Analyst](#)[Data Engineer](#)[SQL Associate](#)[Power BI Data Analyst](#)[Tableau Certified Data Analyst](#)[Azure Fundamentals](#)[AI Fundamentals](#)**RESOURCES**[Resource Center](#)[Upcoming Events](#)[Blog](#)[Code-Alongs](#)[Tutorials](#)[Docs](#)[Open Source](#)[RDocumentation](#)[Course Editor](#)[Book a Demo with DataCamp for Business](#)[Data Portfolio](#)**PLANS**[Pricing](#)[For Students](#)[For Business](#)[For Universities](#)[Discounts, Promos & Sales](#)

[DataCamp Donates](#)

FOR BUSINESS

[Business Pricing](#)

[Teams Plan](#)

[Data & AI Unlimited Plan](#)

[Customer Stories](#)

[Partner Program](#)

ABOUT

[About Us](#)

[Learner Stories](#)

[Careers](#)

[Become an Instructor](#)

[Press](#)

[Leadership](#)

[Contact Us](#)

[DataCamp Español](#)

[DataCamp Português](#)

[DataCamp Deutsch](#)

[DataCamp Français](#)

SUPPORT

[Help Center](#)

[Become an Affiliate](#)



[Privacy Policy](#) [Cookie Notice](#) [Do Not Sell My Personal Information](#) [Accessibility](#) [Security](#) [Terms of Use](#)

© 2025 DataCamp, Inc. All Rights Reserved.