# [**Code examples**](https://keras.io/examples/)

## <https://keras.io/examples/>

Our code examples are short (less than 300 lines of code), focused demonstrations of vertical deep learning workflows.

All of our examples are written as Jupyter notebooks and can be run in one click in [Google Colab](https://colab.research.google.com/notebooks/welcome.ipynb), a hosted notebook environment that requires no setup and runs in the cloud. Google Colab includes GPU and TPU runtimes.

### [**Computer Vision**](https://keras.io/examples/vision/)

* [Image classification from scratch](https://keras.io/examples/vision/image_classification_from_scratch)
* [Simple MNIST convnet](https://keras.io/examples/vision/mnist_convnet)
* [Image segmentation with a U-Net-like architecture](https://keras.io/examples/vision/oxford_pets_image_segmentation)
* [3D Image Classification from CT Scans](https://keras.io/examples/vision/3D_image_classification)
* [Convolutional Autoencoder For Image Denoising](https://keras.io/examples/vision/autoencoder)
* [OCR model for reading Captchas](https://keras.io/examples/vision/captcha_ocr)
* [Consistency Training with Supervision](https://keras.io/examples/vision/consistency_training)
* [Next-frame prediction with Conv-LSTM](https://keras.io/examples/vision/conv_lstm)
* [Grad-CAM class activation visualization](https://keras.io/examples/vision/grad_cam)
* [Image classification via fine-tuning with EfficientNet](https://keras.io/examples/vision/image_classification_efficientnet_fine_tuning)
* [Image Classification with Vision Transformer](https://keras.io/examples/vision/image_classification_with_vision_transformer)
* [Model interpretability with Integrated Gradients](https://keras.io/examples/vision/integrated_gradients)
* [Knowledge Distillation](https://keras.io/examples/vision/knowledge_distillation)
* [Learning to Resize in Computer Vision](https://keras.io/examples/vision/learnable_resizer)
* [Metric learning for image similarity search](https://keras.io/examples/vision/metric_learning)
* [MixUp augmentation for image classification](https://keras.io/examples/vision/mixup)
* [Image Classification with Perceiver](https://keras.io/examples/vision/perceiver_image_classification)
* [Point cloud classification with PointNet](https://keras.io/examples/vision/pointnet)
* [RandAugment for Image Classification for Improved Robustness](https://keras.io/examples/vision/randaugment)
* [Few-Shot learning with Reptile](https://keras.io/examples/vision/reptile)
* [Object Detection with RetinaNet](https://keras.io/examples/vision/retinanet)
* [Semantic Image Clustering](https://keras.io/examples/vision/semantic_image_clustering)
* [Semi-supervised image classification using contrastive pretraining with SimCLR](https://keras.io/examples/vision/semisupervised_simclr)
* [Image similarity estimation using a Siamese Network with a triplet loss](https://keras.io/examples/vision/siamese_network)
* [Self-supervised contrastive learning with SimSiam](https://keras.io/examples/vision/simsiam)
* [Image Super-Resolution using an Efficient Sub-Pixel CNN](https://keras.io/examples/vision/super_resolution_sub_pixel)
* [Supervised Contrastive Learning](https://keras.io/examples/vision/supervised-contrastive-learning)
* [Visualizing what convnets learn](https://keras.io/examples/vision/visualizing_what_convnets_learn)
* [Pneumonia Classification on TPU](https://keras.io/examples/vision/xray_classification_with_tpus)

### [**Natural language processing**](https://keras.io/examples/nlp/)

* [Text classification from scratch](https://keras.io/examples/nlp/text_classification_from_scratch)
* [Sequence to sequence learning for performing number addition](https://keras.io/examples/nlp/addition_rnn)
* [Bidirectional LSTM on IMDB](https://keras.io/examples/nlp/bidirectional_lstm_imdb)
* [Character-level recurrent sequence-to-sequence model](https://keras.io/examples/nlp/lstm_seq2seq)
* [End-to-end Masked Language Modeling with BERT](https://keras.io/examples/nlp/masked_language_modeling)
* [Natural language image search with a Dual Encoder](https://keras.io/examples/nlp/nl_image_search)
* [Using pre-trained word embeddings](https://keras.io/examples/nlp/pretrained_word_embeddings)
* [Semantic Similarity with BERT](https://keras.io/examples/nlp/semantic_similarity_with_bert)
* [Text classification with Switch Transformer](https://keras.io/examples/nlp/text_classification_with_switch_transformer)
* [Text classification with Transformer](https://keras.io/examples/nlp/text_classification_with_transformer)
* [Text Extraction with BERT](https://keras.io/examples/nlp/text_extraction_with_bert)

### [**Structured Data**](https://keras.io/examples/structured_data/)

* [Structured data classification from scratch](https://keras.io/examples/structured_data/structured_data_classification_from_scratch)
* [Classification with Gated Residual and Variable Selection Networks](https://keras.io/examples/structured_data/classification_with_grn_and_vsn)
* [Collaborative Filtering for Movie Recommendations](https://keras.io/examples/structured_data/collaborative_filtering_movielens)
* [Classification with Neural Decision Forests](https://keras.io/examples/structured_data/deep_neural_decision_forests)
* [Imbalanced classification: credit card fraud detection](https://keras.io/examples/structured_data/imbalanced_classification)
* [A Transformer-based recommendation system](https://keras.io/examples/structured_data/movielens_recommendations_transformers)
* [Structured data learning with Wide, Deep, and Cross networks](https://keras.io/examples/structured_data/wide_deep_cross_networks)

### [**Timeseries**](https://keras.io/examples/timeseries/)

* [Timeseries anomaly detection using an Autoencoder](https://keras.io/examples/timeseries/timeseries_anomaly_detection)
* [Timeseries classification from scratch](https://keras.io/examples/timeseries/timeseries_classification_from_scratch)
* [Timeseries forecasting for weather prediction](https://keras.io/examples/timeseries/timeseries_weather_forecasting)

### [**Audio Data**](https://keras.io/examples/audio/)

* [Speaker Recognition](https://keras.io/examples/audio/speaker_recognition_using_cnn)
* [Automatic Speech Recognition with Transformer](https://keras.io/examples/audio/transformer_asr)

### [**Generative Deep Learning**](https://keras.io/examples/generative/)

* [Variational AutoEncoder](https://keras.io/examples/generative/vae)
* [GAN overriding Model.train\_step](https://keras.io/examples/generative/dcgan_overriding_train_step)
* [WGAN-GP overriding Model.train\_step](https://keras.io/examples/generative/wgan_gp)
* [Neural style transfer](https://keras.io/examples/generative/neural_style_transfer)
* [Deep Dream](https://keras.io/examples/generative/deep_dream)
* [CycleGAN](https://keras.io/examples/generative/cyclegan)
* [Character-level text generation with LSTM](https://keras.io/examples/generative/lstm_character_level_text_generation)
* [PixelCNN](https://keras.io/examples/generative/pixelcnn)
* [Density estimation using Real NVP](https://keras.io/examples/generative/real_nvp)
* [Text generation with a miniature GPT](https://keras.io/examples/generative/text_generation_with_miniature_gpt)

### [**Reinforcement learning**](https://keras.io/examples/rl/)

* [Actor Critic Method](https://keras.io/examples/rl/actor_critic_cartpole)
* [Deep Deterministic Policy Gradient (DDPG)](https://keras.io/examples/rl/ddpg_pendulum)
* [Deep Q-Learning for Atari Breakout](https://keras.io/examples/rl/deep_q_network_breakout)

### [**Graph data**](https://keras.io/examples/graph/)

* [Graph representation learning with node2vec](https://keras.io/examples/graph/node2vec_movielens)

### [**Quick Keras recipes**](https://keras.io/examples/keras_recipes/)

* [Simple custom layer example: Antirectifier](https://keras.io/examples/keras_recipes/antirectifier)
* [Probabilistic Bayesian Neural Networks](https://keras.io/examples/keras_recipes/bayesian_neural_networks)
* [Creating TFRecords](https://keras.io/examples/keras_recipes/creating_tfrecords)
* [Keras debugging tips](https://keras.io/examples/keras_recipes/debugging_tips)
* [Endpoint layer pattern](https://keras.io/examples/keras_recipes/endpoint_layer_pattern)
* [Memory-efficient embeddings for recommendation systems](https://keras.io/examples/keras_recipes/memory_efficient_embeddings)
* [A Quasi-SVM in Keras](https://keras.io/examples/keras_recipes/quasi_svm)
* [How to train a Keras model on TFRecord files](https://keras.io/examples/keras_recipes/tfrecord)

## **Adding a new code example**

We welcome new code examples! Here are our rules:

* They should be shorter than 300 lines of code (comments may be as long as you want).
* They should demonstrate modern Keras / TensorFlow 2.0 best practices.
* They should be substantially different in topic from all examples listed above.
* They should be extensively documented & commented.

New examples are added via Pull Requests to the [keras.io repository](https://github.com/keras-team/keras-io). They must be submitted as a .py file that follows a specific format. They are usually generated from Jupyter notebooks. See the [tutobooks documentation](https://github.com/keras-team/keras-io/blob/master/README.md) for more details.