

Level 1—Resources

Help Level 1 contains links to carefully chosen and curated excerpts of Manning books (preferably) and other online resources (if necessary) to help readers with specific workflow steps in the project and milestone they are currently in.

Create Level 1—Resources Content

Think of it as opening a book for your learner and pointing to a section, saying, “If you read these two sections, you'll learn how to [do something specific] that you need in step 3 of this workflow.”

1. Search for specific sections of Manning books and videos that teach concepts needed for a specific workflow step or steps.
2. Provide the book title, chapter, and section.
3. Provide a clear, specific description of what the user should be looking for in this material and which parts of the workflow it will help them with.

Here are two examples of how you could introduce a resource

Note that they point to specific sections and tell the user what they are trying to find there. They could even tell the user where in the project they'll use it.

- Sections 4. 1 and 4.2 of [book title] by [author name] will help you understand how to handle asynchronous events in JavaScript. You'll need to do this when you are working on steps 4 and 5 in the project workflow.
- Sections 6.1-6.5 of [book title] by [author name] will teach you how to work with callbacks in JavaScript.

Here's how we describe Level 1—Resources to your project's users

In this section, you'll find topic-specific reading material curated by the author. You can access these passages with live links embedded in the project.

Level 2—Help

Level 2—Help contains very specific hints and tips for how to solve a specific step or steps in the current workflow. You might include code snippets, a hint, or a keyword - whatever you think might help your user go from "puzzled" to "aha!"

The liveProject platform lets us include formatted code. Do keep it brief and targeted to specific steps.

Create Level 2—Help Content

1. Write a hint for a specific step or steps in the current workflow.
2. Indicate the step(s) and provide the formatted hint.
3. Start your hint with "Hint for Step n:" "Hint for Steps n-p:" "Hint for Workflow:"
4. Add at least one sentence of introductory text before any code to tell the user what it's helping them with
5. Not every step will have a hint. Some hints might go with more than one step. Try not to provide a hint that goes with more than a step or two, but if you need to do that, you can.

Here is an example of Level 2—Help content

```
help

Hint for Step 1:
Inspect shape by using the .shape numpy function e.g.
x_train.shape

Hint for Step 1:
Show random images from the loaded dataset by using matplotlib,
e.g.: random_num = np.random.randint(0, len(x_train))
plt.imshow(x_train[random_num], cmap =
plt.get_cmap('gray'))

Hint for Step 2:
To hot-one-encode your data labels you can use the tensorflow's
to_categorical function e.g.: y_train =
to_categorical(y_train)

Hint for Step 3:
To get you started making your Convolutional Neural Network you
need to import the required functions to build and start adding
Conv layers like this:

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Flatten
from tensorflow.keras.layers import Conv2D, MaxPooling2D
from tensorflow.keras import backend as k
from tensorflow.keras.optimizers import SGD

# create model
model = Sequential()

# Our First Convolution Layer, Filter size 32 which reduces our layer size to
26 x 26 x 32
# We use ReLU activation and specify our input_shape which is 28 x 28 x 1
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu',
input_shape=input_shape))
```

Here's how we describe Level 2—Help to your project's users

Sometimes you need a hint to get moving in the right direction. Here, the author provides clues to help you get past common pitfalls. It's like having a mentor ready with a useful suggestion when you need it.

Level 3—Partial Solution

Level 3 is a partial solution to a workflow in a milestone. If there is a deliverable, there should be a partial solution for that deliverable.

This level is for users that need a lot of help but aren't ready to give up yet and move to level 4 (the full solution). Make your partial solution something truly helpful; there's no need to be tricky. You want users to finish the project and feel like they learned something along the way.

Create Level 3—Partial Solution Content

Often a *full* solution is a very well commented Python or Jupyter notebook or code listing. In those cases, a *partial* solution could be the full solution *with all the code removed*, but the robust comments remaining.

1. Create your partial solution. Ideally, this is a skeleton of your full solution.
2. Read through it, trying to put yourself in the place of a possibly confused and frustrated user. Do you need to add any more comments, code descriptions, instructions, or tips to help them complete the project?
3. Use the same filename convention as you do for your full solution, but add PARTIAL at the beginning of the filename.

Here's how we describe Level 3—Partial Solution to your project's users:

If you've read the hints but you're still scratching your head, this section provides a partial solution. For many projects, this includes a commented code listing or notebook with the code removed so you can use the comments and author's solution structure as a guide for writing your own code.

Level 4—Full Solution

Level 4 is the full solution to your project. Do not create anything special for this level; we provide the user with your full solution file here.

Here's how we describe Level 4 —Full Solution to your project's users

In the final level of help, we provide the full solution. There's no shame in needing it: Sometimes you're just stumped. Click for the solution and move to the next part of the project.