

# Code Llama

## Code Llama

- Model size
- Pull Code Llama
- Use Code Llama
  - Run Code Llama
  - Have a conversation
  - End the conversation
- References

### Demo Environment

**Development board:** Jetson Orin series motherboard

**SSD:** 128G

**Tutorial application scope:** Whether the motherboard can run is related to the available memory of the system. The user's own environment and the programs running in the background may cause the model to fail to run

Motherboard model	Ollama	Open WebUI
Jetson Orin NX 16GB	√	√
Jetson Orin NX 8GB	√	√
Jetson Orin Nano 8GB	√	√
Jetson Orin Nano 4GB	×	×

Code Llama is an open source Large Language Model (LLM) designed by Meta AI specifically for understanding and generating code.

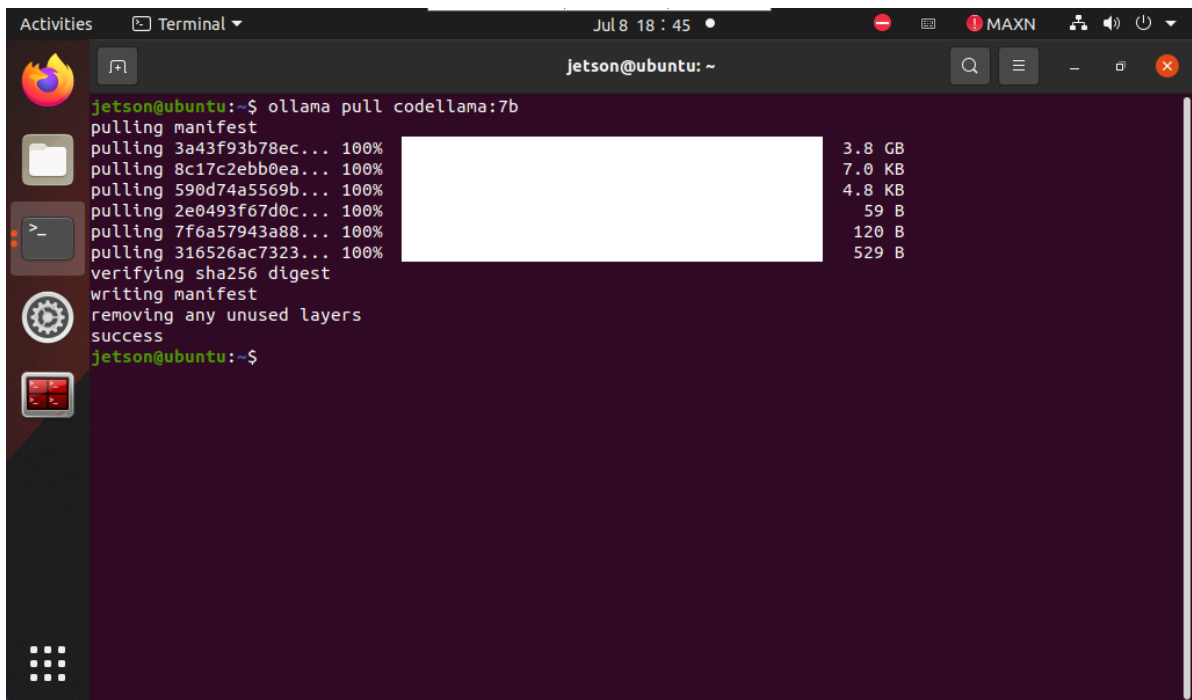
## Model size

Model	Parameters
Code Llama	7B
Code Llama	13B
Code Llama	34B
Code Llama	70B

## Pull Code Llama

Using the pull command will automatically pull the model from the Ollama model library:

```
ollama pull code llama:7b
```

A terminal window on a Jetson Ubuntu system showing the command 'ollama pull codellama:7b'. The output shows the model being pulled in layers: manifest, 3a43f93b78ec... (3.8 GB), 8c17c2ebb0ea... (7.0 KB), 590d74a5569b... (4.8 KB), 2e0493f67d0c... (59 B), 7f6a57943a88... (120 B), and 316526ac7323... (529 B). The process includes verifying the sha256 digest, writing the manifest, and removing any unused layers, resulting in a successful pull.

```
jetson@ubuntu:~$ ollama pull codellama:7b
pulling manifest
pulling 3a43f93b78ec... 100% 3.8 GB
pulling 8c17c2ebb0ea... 100% 7.0 KB
pulling 590d74a5569b... 100% 4.8 KB
pulling 2e0493f67d0c... 100% 59 B
pulling 7f6a57943a88... 100% 120 B
pulling 316526ac7323... 100% 529 B
verifying sha256 digest
writing manifest
removing any unused layers
success
jetson@ubuntu:~$
```

## Use Code Llama

### Run Code Llama

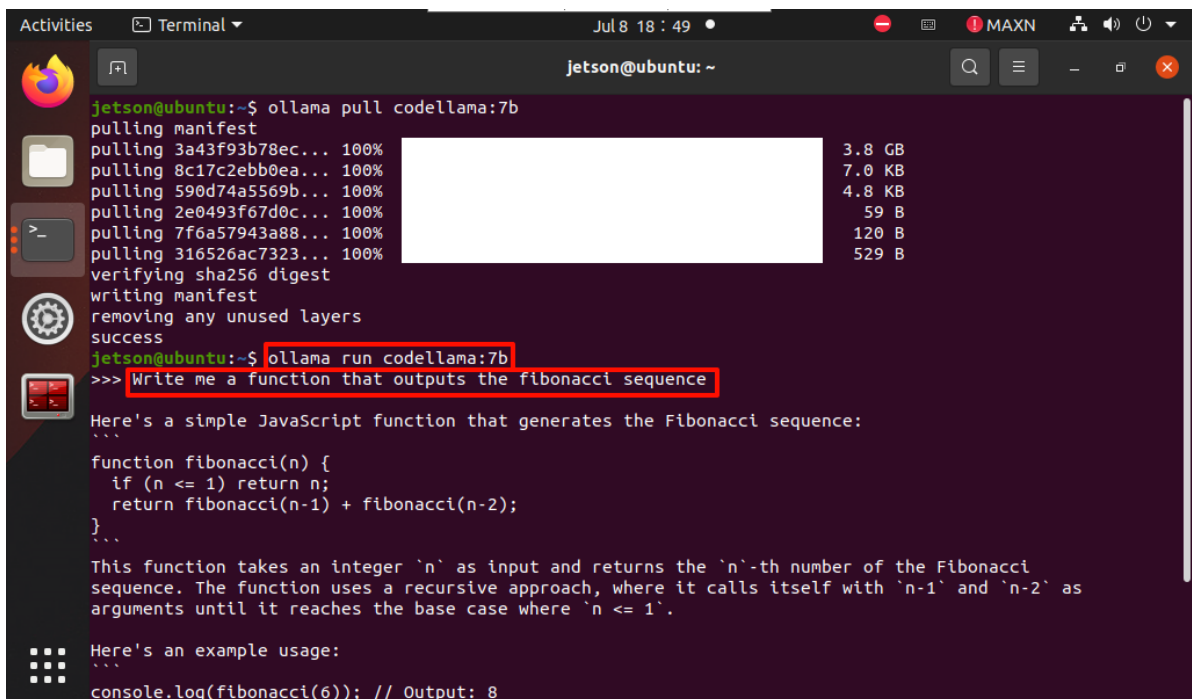
If the system does not have a running model, the system will automatically pull the Code Llama 7B model and run it:

```
ollama run codellama:7b
```

### Have a conversation

```
write me a function that outputs the fibonacci sequence
```

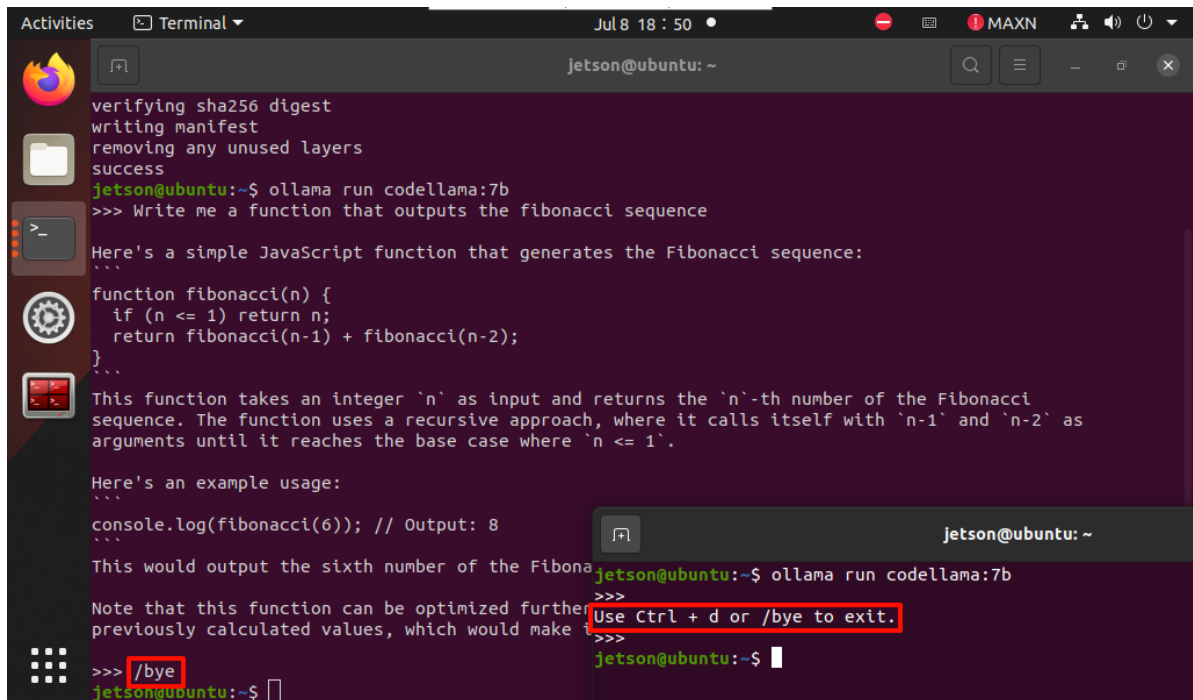
The time to reply to the question depends on the hardware configuration, please be patient!

A terminal window on a Jetson Ubuntu system showing the command 'ollama run codellama:7b'. The output shows the model being pulled in layers: manifest, 3a43f93b78ec... (3.8 GB), 8c17c2ebb0ea... (7.0 KB), 590d74a5569b... (4.8 KB), 2e0493f67d0c... (59 B), 7f6a57943a88... (120 B), and 316526ac7323... (529 B). The process includes verifying the sha256 digest, writing the manifest, and removing any unused layers, resulting in a successful pull. The terminal then shows the command 'ollama run codellama:7b' and the prompt 'Write me a function that outputs the fibonacci sequence'. The model responds with a JavaScript function that generates the Fibonacci sequence, followed by an example usage: 'console.log(fibonacci(6)); // Output: 8'.

```
jetson@ubuntu:~$ ollama pull codellama:7b
pulling manifest
pulling 3a43f93b78ec... 100% 3.8 GB
pulling 8c17c2ebb0ea... 100% 7.0 KB
pulling 590d74a5569b... 100% 4.8 KB
pulling 2e0493f67d0c... 100% 59 B
pulling 7f6a57943a88... 100% 120 B
pulling 316526ac7323... 100% 529 B
verifying sha256 digest
writing manifest
removing any unused layers
success
jetson@ubuntu:~$ ollama run codellama:7b
>>> Write me a function that outputs the fibonacci sequence
Here's a simple JavaScript function that generates the Fibonacci sequence:
...
function fibonacci(n) {
  if (n <= 1) return n;
  return fibonacci(n-1) + fibonacci(n-2);
}
...
This function takes an integer 'n' as input and returns the 'n'-th number of the Fibonacci
sequence. The function uses a recursive approach, where it calls itself with 'n-1' and 'n-2' as
arguments until it reaches the base case where 'n <= 1'.
Here's an example usage:
...
console.log(fibonacci(6)); // Output: 8
```

## End the conversation

Use the `Ctrl+d` shortcut key or `/bye` to end the conversation!



```
Activities Terminal Jul 8 18:50 MAXN
jetson@ubuntu: ~
verifying sha256 digest
writing manifest
removing any unused layers
success
jetson@ubuntu:~$ ollama run codellama:7b
>>> Write me a function that outputs the fibonacci sequence

Here's a simple JavaScript function that generates the Fibonacci sequence:
...
function fibonacci(n) {
  if (n <= 1) return n;
  return fibonacci(n-1) + fibonacci(n-2);
}
...
This function takes an integer 'n' as input and returns the 'n'-th number of the Fibonacci
sequence. The function uses a recursive approach, where it calls itself with 'n-1' and 'n-2' as
arguments until it reaches the base case where 'n <= 1'.

Here's an example usage:
...
console.log(fibonacci(6)); // Output: 8
...
This would output the sixth number of the Fibonacci sequence.

Note that this function can be optimized further by storing previously calculated values, which would make it faster.

>>> /bye
jetson@ubuntu:~$
```

## References

### Ollama

Official website: <https://ollama.com/>

GitHub: <https://github.com/ollama/ollama>

### Code Llama

Ollama corresponding model: <https://ollama.com/library/codellama>

GitHub: <https://github.com/meta-llama/codellama>