

# Gemma model

---

[Gemma model](#)  
[Model scale](#)  
[Got Gemma](#)  
[Run Gemma](#)  
[Run Gemma](#)  
[Dialogue](#)  
[End conversation](#)  
[Reference material](#)

## Demonstration environment

**Development board** : Raspberry Pi 5B (8G RAM)

**SD(TF)card**: 64G (Above 16G, the larger the capacity, the more models can be experienced)

Raspberry Pi 5B (16G RAM): Run 14B and below parameter models  
Raspberry Pi 5B (8G RAM): Run 8B and below parameter models  
Raspberry Pi 5B (4G RAM): Run 3B and below parameter models  
Raspberry Pi 5B (2G RAM): Run 3B and below parameter models, can't run Gemma model

Gemma is a new open model developed by Google and its DeepMind team.

## Model scale

Model	Parameter
Gemma	2B
Gemma	7B

Raspberry Pi 5B (8G RAM): Test Gemma model with 7B parameters!  
Raspberry Pi 5B (4G RAM): Test Gemma Model with 2B Parameters!

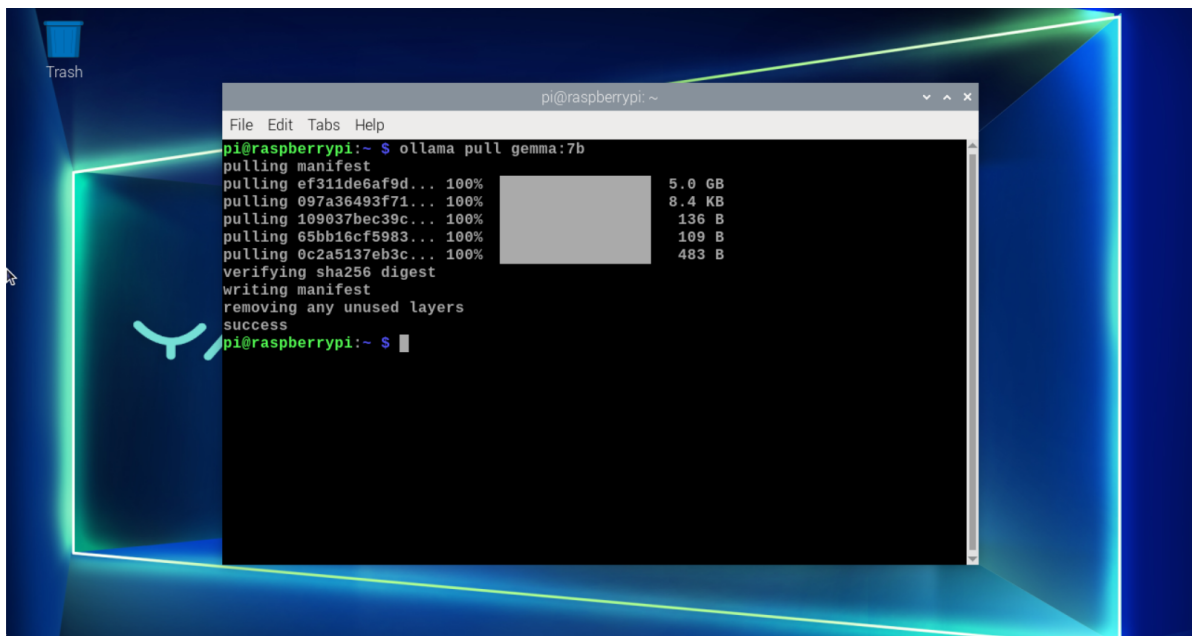
## Got Gemma

---

Using the pull command will automatically retrieve the models from the Ollama model library.

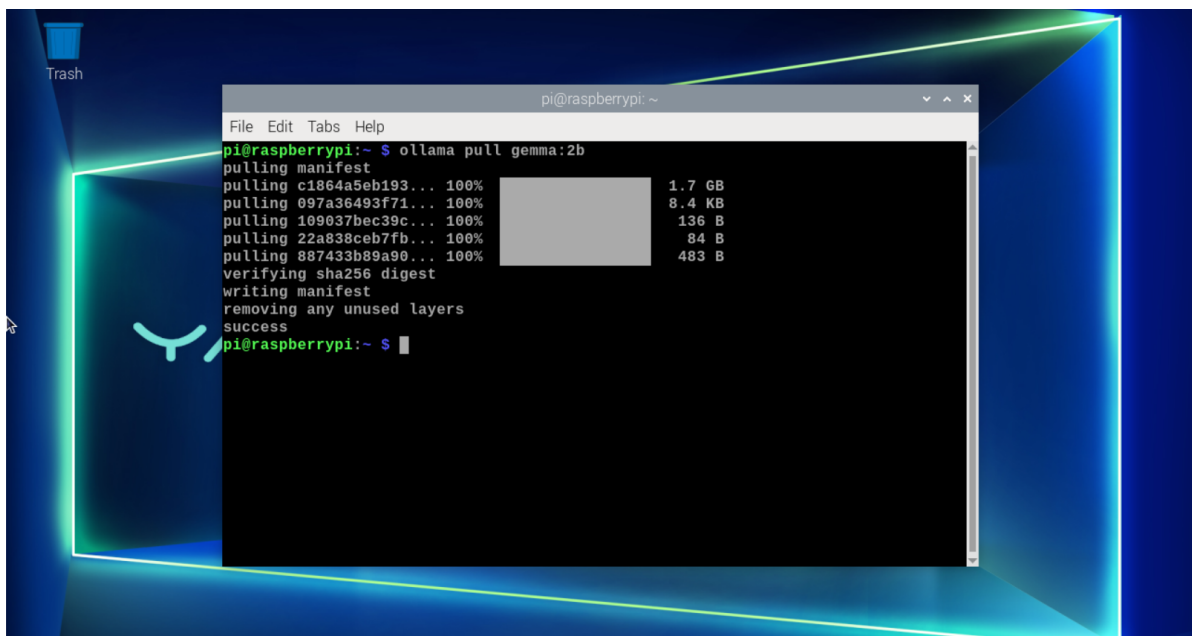
### Raspberry Pi 5B (8G RAM)

```
ollama pull gemma:7b
```



### Raspberry Pi 5B (4G RAM)

```
ollama pull gemma:2b
```



## Run Gemma

### Run Gemma

#### Raspberry Pi 5B (8G RAM)

If the system does not have a running model, the system will automatically pull the Gemma 7B model and run it: **The factory image does not provide this image, you need to download it yourself!**

```
ollama run gemma:7b
```

#### Raspberry Pi 5B (4G RAM)

If the system does not have a running model, the system will automatically obtain the Gemma 2B model and run it.

```
ollama run gemma:2b
```

## Dialogue

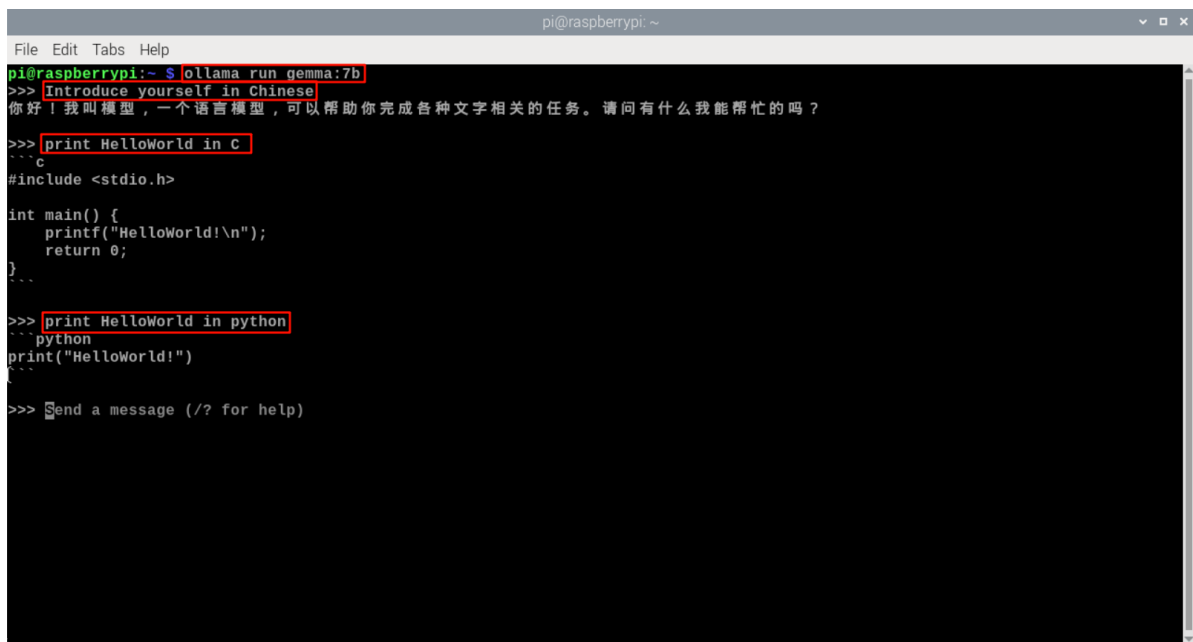
### Raspberry Pi 5B (8G RAM)

```
Introduce yourself in Chinese
```

```
print HelloWorld in C
```

```
print HelloWorld in python
```

The time to reply to the question is related to the hardware configuration, please be patient.



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ ollama run gemma:7b  
>>> Introduce yourself in Chinese  
你好！我叫模型，一个语言模型，可以帮助你完成各种文字相关的任务。请问有什么我能帮忙的吗？  
>>> print HelloWorld in C  
...  
c  
#include <stdio.h>  
  
int main() {  
    printf("HelloWorld!\n");  
    return 0;  
}  
...  
>>> print HelloWorld in python  
...  
python  
print("HelloWorld!")  
...  
>>> Send a message (/? for help)
```

### Raspberry Pi 5B (4G RAM)

```
print HelloWorld in C
```

The time to reply to the question is related to the hardware configuration, please be patient.

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ ollama run gemma:2b  
>>> print HelloWorld in C  
Sure, here's the C code to print "HelloWorld" in C:  
  
```c  
#include <stdio.h>  
  
int main() {  
    printf("HelloWorld\n");  
  
    return 0;  
}  
```\n\n**Explanation:**  
1. We include the `stdio.h` header file, which provides the `printf` function.  
2. We define the `main` function.  
3. Inside the `main` function, we use the `printf` function to print the string "HelloWorld" to the console.  
4. The `\\n` character is used to insert a newline character into the output.  
  
**Output:**  
  
HelloWorld  
  
**Notes:**  
  
* You can compile this code using any C compiler.  
* You can run the compiled program by executing the `./your_file_name` command.  
* You can modify the string in the `printf` function to print different messages.
```

## End conversation

You can end the conversation by using the shortcut key 'Ctrl+d' or '/bye'.

### Raspberry Pi 5B (8G RAM)

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ ollama run gemma:7b  
>>> Introduce yourself in Chinese  
你好！我叫模型，一个语言模型，可以帮助你完成各种文字相关的任务。请问有什么我能帮忙的吗？  
  
>>> print HelloWorld in c  
```c  
#include <stdio.h>  
  
int main() {  
    printf("HelloWorld!\n");  
    return 0;  
}  
```\n\n>>> print HelloWorld in python  
```python  
print("HelloWorld!")  
```\n\n>>> /bye  
pi@raspberrypi:~ $
```

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ ollama run gemma:7b  
>>>  
Use Ctrl + d or /bye to exit.  
>>>  
pi@raspberrypi:~ $
```

### Raspberry Pi 5B (4G RAM)

The screenshot shows a terminal window titled 'pi@raspberrypi: ~'. The main window contains a C program that prints 'HelloWorld'. A smaller, semi-transparent terminal window is overlaid on top, showing the command 'ollama run gemma:2b' being executed. The output of the command is a list of instructions and a prompt to use Ctrl + d or /bye to exit. The main terminal window also shows the output of the C program, 'HelloWorld', and a list of notes about compiling and running the code. The prompt 'pi@raspberrypi:~ S' is visible at the bottom of the main window.

```
pi@raspberrypi: ~
File Edit Tabs Help

'''c
#include <stdio.h>

int main() {
    printf("HelloWorld\n");
}
return 0;
'''

**Explanation:**
1. We include the `stdio.h` header file, which
2. We define the `main` function.
3. Inside the `main` function, we use the `printf`
4. The `\n` character is used to insert a newl

**Output:**
'''
HelloWorld
'''

**Notes:**
* You can compile this code using any C compile
* You can run the compiled program by executing
* You can modify the string in the `printf` fur

>>> /bye
pi@raspberrypi:~ S
```

pi@raspberrypi: ~
File Edit Tabs Help

pi@raspberrypi:~ S ollama run gemma:2b
>>>
Use Ctrl + d or /bye to exit.
>>>
pi@raspberrypi:~ S

## Reference material

### Ollama

Website: <https://ollama.com/>

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

### Gemma

GitHub: <https://github.com/google-deepmind/gemma>

Ollama model: <https://ollama.com/library/gemma>