

20.6 Vertical overflow

By default, the live display will display ellipsis if the renderable is too large for the terminal. You can adjust this by setting the `vertical_overflow` argument on the `Live` constructor.

- “crop” Show renderable up to the terminal height. The rest is hidden.
- “ellipsis” Similar to crop except last line of the terminal is replaced with “...”. This is the default behavior.
- “visible” Will allow the whole renderable to be shown. Note that the display cannot be properly cleared in this mode.

Note

Once the live display stops on a non-transient renderable, the last frame will render as **visible** since it doesn’t have to be cleared.

20.7 Print / log

The `Live` class will create an internal `Console` object which you can access via `live.console`. If you print or log to this console, the output will be displayed *above* the live display. Here’s an example:

```
import time

from rich.live import Live
from rich.table import Table

table = Table()
table.add_column("Row ID")
table.add_column("Description")
table.add_column("Level")

with Live(table, refresh_per_second=4) as live: # update 4 times a second to feel fluid
    for row in range(12):
        live.console.print(f"Working on row #{row}")
        time.sleep(0.4)
        table.add_row(f"{row}", f"description {row}", "[red]ERROR")
```

If you have another `Console` object you want to use, pass it in to the `Live` constructor. Here’s an example:

```
from my_project import my_console

with Live(console=my_console) as live:
    my_console.print("[bold blue]Starting work!")
    ...
```

Note

If you are passing in a file `console`, the live display only show the last item once the live context is left.