CHAPTER

TWELVE

RENDER GROUPS

The *Group* class allows you to group several renderables together so they may be rendered in a context where only a single renderable may be supplied. For instance, you might want to display several renderables within a *Panel*.

To render two panels within a third panel, you would construct a Group with the *child* renderables as positional arguments then wrap the result in another Panel:

```python
from rich import print
from rich.console import Group
from rich.panel import Panel

panel_group = Group(
    Panel("Hello", style="on blue"),
    Panel("World", style="on red"),
)
print(Panel(panel_group))
```

This pattern is nice when you know in advance what renderables will be in a group, but can get awkward if you have a larger number of renderables, especially if they are dynamic. Rich provides a *group()* decorator to help with these situations. The decorator builds a group from an iterator of renderables. The following is the equivalent of the previous example using the decorator:

```python
from rich import print
from rich.console import group
from rich.panel import Panel

@group()
def get_panels():
    yield Panel("Hello", style="on blue")
    yield Panel("World", style="on red")

print(Panel(get_panels()))
```