

```
import json
import rich.progress

with rich.progress.open("data.json", "rb") as file:
    data = json.load(file)
print(data)
```

If you already have a file object, you can call `wrap_file()` which returns a context manager that wraps your file so that it displays a progress bar. If you use this function you will need to set the number of bytes or characters you expect to read.

Here's an example that reads a url from the internet:

```
from time import sleep
from urllib.request import urlopen

from rich.progress import wrap_file

response = urlopen("https://www.textualize.io")
size = int(response.headers["Content-Length"])

with wrap_file(response, size) as file:
    for line in file:
        print(line.decode("utf-8"), end="")
        sleep(0.1)
```

If you expect to be reading from multiple files, you can use `open()` or `wrap_file()` to add a file progress to an existing Progress instance.

See `cp_progress.py` for a minimal clone of the `cp` command which shows a progress bar as the file is copied.

16.3 Nesting Progress bars

If you create a new progress bar within the context of an existing progress bar (with the context manager or `track` function), then Rich will display the inner progress bar(s) under the initial bar.

Here's an example that nests progress bars:

```
from rich.progress import track
from time import sleep

for count in track(range(10)):
    for letter in track("ABCDEF", transient=True):
        print(f"Stage {count}{letter}")
        sleep(0.1)
    sleep(0.1)
```

The inner loop creates a new progress bar below the first, but both can update.

Note that if you nest progress bars like this, then the nested bars will updating according to the `refresh_per_second` attribute of the outer bar.