The `update()` method collects keyword arguments which are also associated with the task. Use this to supply any additional information you would like to render in the progress display. The additional arguments are stored in `task.fields` and may be referenced in *Column classes*.

16.2.3 Hiding tasks

You can show or hide tasks by updating the tasks `visible` value. Tasks are visible by default, but you can also add an invisible task by calling `add_task()` with `visible=False`.

16.2.4 Transient progress

Normally when you exit the progress context manager (or call `stop()`) the last refreshed display remains in the terminal with the cursor on the following line. You can also make the progress display disappear on exit by setting `transient=True` on the Progress constructor. Here's an example:

```
with Progress(transient=True) as progress:
    task = progress.add_task("Working", total=100)
    do_work(task)
```

Transient progress displays are useful if you want more minimal output in the terminal when tasks are complete.

16.2.5 Indeterminate progress

When you add a task it is automatically *started*, which means it will show a progress bar at 0% and the time remaining will be calculated from the current time. This may not work well if there is a long delay before you can start updating progress; you may need to wait for a response from a server or count files in a directory (for example). In these cases you can call `add_task()` with `start=False` or `total=None` which will display a pulsing animation that lets the user know something is working. This is known as an *indeterminate* progress bar. When you have the number of steps you can call `start_task()` which will display the progress bar at 0%, then `update()` as normal.

16.2.6 Auto refresh

By default, the progress information will refresh 10 times a second. You can set the refresh rate with the `refresh_per_second` argument on the `Progress` constructor. You should set this to something lower than 10 if you know your updates will not be that frequent.

You might want to disable auto-refresh entirely if your updates are not very frequent, which you can do by setting `auto_refresh=False` on the constructor. If you disable auto-refresh you will need to call `refresh()` manually after updating your task(s).

16.2.7 Expand

The progress bar(s) will use only as much of the width of the terminal as required to show the task information. If you set the `expand` argument on the Progress constructor, then Rich will stretch the progress display to the full available width.

16.2.8 Columns

You may customize the columns in the progress display with the positional arguments to the `Progress` constructor. The columns are specified as either a format string or a `ProgressColumn` object.

Format strings will be rendered with a single value *"task"* which will be a `Task` instance. For example `"{task.description}"` would display the task description in the column, and `"{task.completed} of {task.total}"` would display how many of the total steps have been completed. Additional fields passed via keyword arguments to *~rich.progress.Progress.update* are stored in `task.fields`. You can add them to a format string with the following syntax: `"extra info:  {task.fields[extra]}"`.