Why use context managers?

contextlib.suppress

```
def kill_process(pid):
    # Kill a process, ignore if it can't be found.
    with contextlib.suppress(ProcessLookupError):
        os.kill(pid, signal.SIGKILL)

# Nicer than:
def kill_process(pid):
    try:
        os.kill(pid, signal.SIGKILL)
    except ProcessLookupError:
        pass
```

<link to slides>

Why use context managers?

ThreadPoolExecutor - Bad version!

```
from concurrent.futures import ThreadPoolExecutor

# Bad!!!
pool = ThreadPoolExecutor()
for k, v in data.items():
    pool.submit(make_a_sentence, k, v)
# Wait on the results and do something with them.
pool.shutdown()
```

<link to slides>