presentation available at

**bit.ly/multiprocessing**


# Multiprocessing in Python

Karol Horosin

ardigen

Artificial Intelligence & Bioinformatics
for Precision Medicine

CODE AGAINST CANCER

**karol@horosin.com**


why?

# why not threading?

the talk:

# practical example

# +

# useful features

# process

execution context
memory, binary code, resources
contains threads (1+)
has separate address space

# thread

smallest unit of execution in OS
shares memory within a process

# Process vs thread for computing

| **process** | **thread** |
| --- | --- |
| can run in parallel in Python | can run only concurrently in Python (GIL) - no multicore |
| separate memory space (easy handling, harder communications - IPC) | shared memory space (hard management, easy communication) |
| larger memory footprint (usually used in tens - hundreds) | lightweight (can be used in hundreds - thousands), in linux 4MB base size |

# Let’s code

# takeaways

- It’s doable
- Remember not to copy large objects between processes


eventory

# Q&A