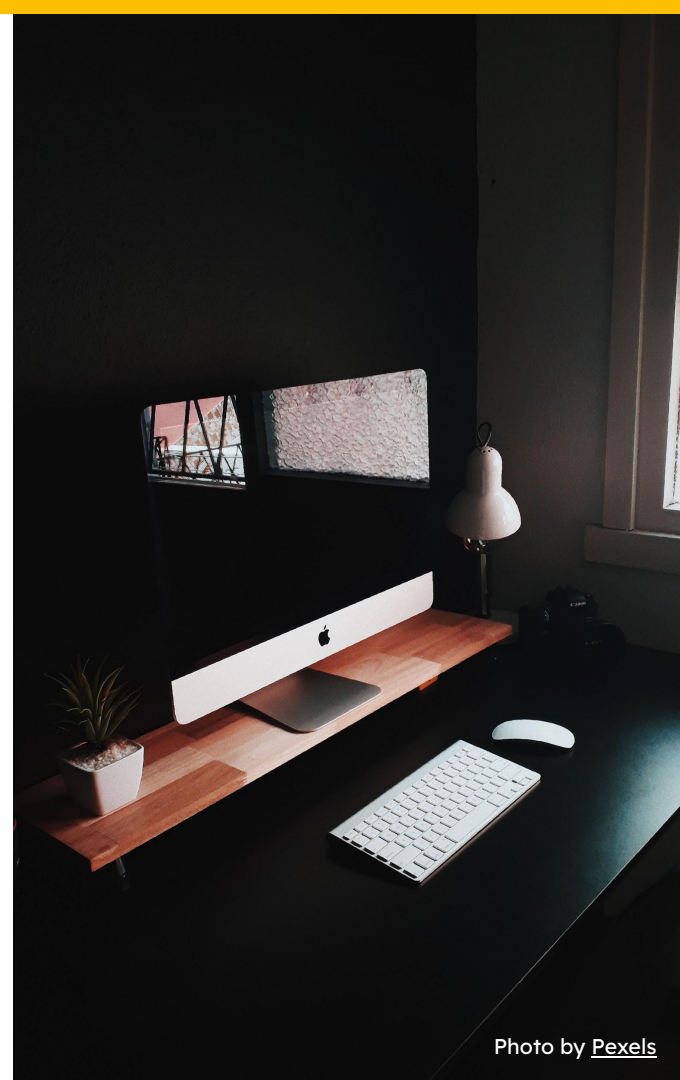


Introduction to Multi-Threading

This presentation provides an overview of multi-threading, including advantages, how to create threads in Python, and how to synchronize access to shared resources.

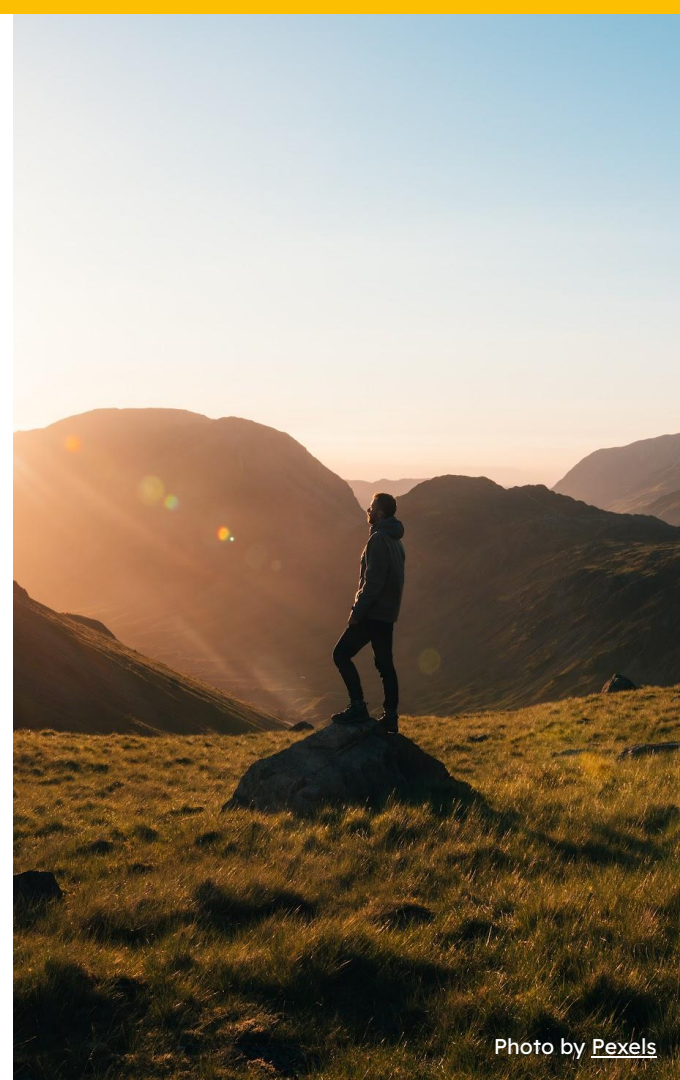
What is Multi-Threading?

- Ability of a program to execute multiple threads concurrently within a single process.
- Each thread runs independently, but can share common resources and memory.



Advantages of Multi-Threading

- Increased performance and responsiveness of the application.
- Better utilization of multi-core CPUs.
- Simplifies programming for tasks that can be parallelized.



Creating a Thread in Python

- Use the Thread class from the threading module.
- Pass a target function to be executed in the new thread.

```
,
E,
_TYPE,
YPE,
actSymbols';
ementType from 'shared/isValidElementType';

n typeof(object: any) {
object === 'object' && object !== null) {
typeof = object.$$typeof;
($$typeof) {
  REACT_ELEMENT_TYPE:
  const type = object.type;

  switch (type) {
    case REACT_FRAGMENT_TYPE:
    case REACT_PROFILER_TYPE:
    case REACT_STRICT_MODE_TYPE:
    case REACT_SUSPENSE_TYPE:
      return type;
    default:
      const $$typeofType = type && type.$$typeof;
      switch ($$typeofType) {
        case REACT_CONTEXT_TYPE:
        case REACT_FORWARD_REF_TYPE:
        case REACT_LAZY_TYPE:
```

Synchronizing Threads with Locks

- Use a Lock object from the threading module.
- Acquire lock to synchronize access to shared resource.



Thank you for your time and attention 😊