Sentinel objects

> This instance of `PyTypeObject` represents the Python `sentinel` type. This is the same object as `sentinel`.
>
> 3.15

> Return true if *o* is a `sentinel` object or a subtype. The `sentinel` type does not currently allow subclasses, so this check is exact. Future Python versions may choose to allow subtyping. This function always succeeds.
>
> 3.15

> Return true if *o* is a `sentinel` object, but not a subtype. The `sentinel` type does not currently allow subclasses. Future Python versions may choose to allow subtyping. This function always succeeds.
>
> 3.15

> Return a new `sentinel` object with `~sentinel.__name__` set to *name* and `~sentinel.__module__` set to *module_name*. *name* must not be `NULL`. If *module_name* is `NULL`, `~sentinel.__module__` is set to `None`. If *repr* is `NULL`, `repr()` returns `~sentinel.__name__`. Return `NULL` with an exception set on failure.
>
> For pickling to work, *module_name* must be the name of an importable module, and the sentinel must be accessible from that module under a path matching *name*. Pickle treats *name* as a global variable name in *module_name* (see `object.__reduce__`).
>
> 3.15

□□□ #2
□ □□□ □□□ 2026-10-31 16:53:52 UTC
□ □□□ □□□ 2026-10-31 16:53:54 UTC