Pending removal in Python 3.18

- The following private functions are deprecated and planned for removal in Python 3.18:
  - `!_PyBytes_Join`: use `PyBytes_Join`.
  - `!_PyDict_GetItemStringWithError`: use `PyDict_GetItemStringRef`.
  - `!_PyDict_Pop()`: use `PyDict_Pop`.
  - `!_PyLong_Sign()`: use `PyLong_GetSign`.
  - `!_PyLong_FromDigits` and `!_PyLong_New`: use `PyLongWriter_Create`.
  - `!_PyThreadState_UncheckedGet`: use `PyThreadState_GetUnchecked`.
  - `!_PyUnicode_AsString`: use `PyUnicode_AsUTF8`.
  - `!_PyUnicodeWriter_Init`: replace `_PyUnicodeWriter_Init(&writer)` with `writer = PyUnicodeWriter_Create(0) <PyUnicodeWriter_Create>`.
  - `!_PyUnicodeWriter_Finish`: replace `_PyUnicodeWriter_Finish(&writer)` with `PyUnicodeWriter_Finish(writer) <PyUnicodeWriter_Finish>`.
  - `!_PyUnicodeWriter_Dealloc`: replace `_PyUnicodeWriter_Dealloc(&writer)` with `PyUnicodeWriter_Discard(writer) <PyUnicodeWriter_Discard>`.
  - `!_PyUnicodeWriter_WriteChar`: replace `_PyUnicodeWriter_WriteChar(&writer, ch)` with `PyUnicodeWriter_WriteChar(writer, ch) <PyUnicodeWriter_WriteChar>`.
  - `!_PyUnicodeWriter_WriteStr`: replace `_PyUnicodeWriter_WriteStr(&writer, str)` with `PyUnicodeWriter_WriteStr(writer, str) <PyUnicodeWriter_WriteStr>`.
  - `!_PyUnicodeWriter_WriteSubstring`: replace `_PyUnicodeWriter_WriteSubstring(&writer, str, start, end)` with `PyUnicodeWriter_WriteSubstring(writer, str, start, end) <PyUnicodeWriter_WriteSubstring>`.
  - `!_PyUnicodeWriter_WriteASCIIString`: replace `_PyUnicodeWriter_WriteASCIIString(&writer, str, size)` with `PyUnicodeWriter_WriteASCII(writer, str, size) <PyUnicodeWriter_WriteASCII>`.
  - `!_PyUnicodeWriter_WriteLatin1String`: replace `_PyUnicodeWriter_WriteLatin1String(&writer, str, size)` by decoding the string with `PyUnicode_DecodeLatin1`, writing the result with `PyUnicodeWriter_WriteStr`, and releasing the result with `Py_DECREF`.
  - `!_PyUnicodeWriter_Prepare`: (no replacement).
  - `!_PyUnicodeWriter_PrepareKind`: (no replacement).
  - `!_Py_HashPointer`: use `Py_HashPointer`.
  - `!_Py_fopen_obj`: use `Py_fopen`.
  - `PyGen_New`: (no replacement).
  - `PyGen_NewWithQualName`: (no replacement).
  - `PyCoro_New`: (no replacement).
  - `PyAsyncGen_New`: (no replacement).

The [pythoncapi-compat project] can be used to get these new public functions on Python 3.13 and older. (Contributed by Victor Stinner in `128863`.)

□□□ #2
□ □□□ □□□ 2026-10-31 16:55:45 UTC
□ □□□ □□□ 2026-10-31 16:55:46 UTC