tf.gfile.FastGFile

Class **FastGFile**

Defined in `tensorflow/python/platform/gfile.py` .

File I/O wrappers without thread locking.

Properties

mode

Returns the mode in which the file was opened.

name

Returns the file name.

Methods

__init__

```
__init__(
    name,
    mode='r'
)
```

__enter__

```
__enter__()
```

Make usable with "with" statement.

__exit__

```
__exit__(
    unused_type,
    unused_value,
    unused_traceback
)
```

Make usable with "with" statement.

__iter__

```
__iter__()
```

__next__

```
__next__()
```

close

```
close()
```

Closes FileIO. Should be called for the WritableFile to be flushed.

flush

```
flush()
```

Flushes the Writable file.

This only ensures that the data has made its way out of the process without any guarantees on whether it's written to disk. This means that the data would survive an application crash but not necessarily an OS crash.

next

```
next()
```

read

```
read(n=-1)
```

Returns the contents of a file as a string.

Starts reading from current position in file.

Args:

- `n`: Read 'n' bytes if n != -1. If n = -1, reads to end of file.

Returns:

'n' bytes of the file (or whole file) in bytes mode or 'n' bytes of the string if in string (regular) mode.

readline

```
readline()
```

Reads the next line from the file. Leaves the '\n' at the end.

readlines

```
readlines()
```

Returns all lines from the file in a list.

seek

```
seek(
    offset=None,
    whence=0,
    position=None
)
```

Seeks to the offset in the file. (deprecated arguments)

SOME ARGUMENTS ARE DEPRECATED. They will be removed in a future version. Instructions for updating: position is deprecated in favor of the offset argument.

Args:

- `offset` : The byte count relative to the whence argument.
- `whence` : Valid values for whence are:
- `0` : start of the file (default)
- `1` : relative to the current position of the file
- `2` : relative to the end of file. offset is usually negative.

size

```
size()
```

Returns the size of the file.

tell

```
tell()
```

Returns the current position in the file.

write

```
write(file_content)
```

Writes file_content to the file. Appends to the end of the file.

Stay Connected

Blog

GitHub

Twitter

Support

Issue Tracker

Release Notes

Stack Overflow

English

Terms | **Privacy**