Ruby 101

Intermediate Ruby:File Objects

Intermediate Ruby:

How do we read a file?

The File class...

Create an instance

```
my_file = File.new("my_file")
```

--SAME AS--

```
my_file = File.open("my_file")
```

Intermediate Ruby:

File inherits from IO: read methods available

`.read` [length [, output]]	-> reads *length* number of bytes into the *output* buffer. Without arguments, it reads till the end of the file.
`.readline`	-> reads one line at a time
`.pos`	-> returns the current position
`.pos =`	-> seeks to a new position
`.rewind`	-> goes back to the beginning
`.readlines`	-> return an array, each element representing a line of the file

Intermediate Ruby:

What about writing?

a) Must have permissions to write
b) File must be opened with correct flags

Intermediate Ruby:

File permissions:

May have encountered already; some directories cannot be read.

Permisisons are handled in octal:

Quick review:

4 = read; 2 = write; 1 = execute/search

sum the abilities for each of the last three permissions columns

Four columns (left to right)

1) Special properties -> probably want this to be 0
2) Owner permissions
3) Group permissions
4) Other user permissions

eg "0755"

owner can do everything, group and other can read and execute

sometimes written as "rwxr-xr-x"

Intermediate Ruby:

File permissions: Checking

File.world_readable? *file_name*
File.readable? *file_name*
File.readable_real? *file_name*

File.world_writable? *file_name*
File.writable? *file_name*
File.writable_real? *file_name*

File.executable? *file_name*
File.executable_real? *file_name*

Intermediate Ruby:

File permissions: Setting

.chmod(*octal_permission)*

eg

```
my_file = File.new("my_file.txt")
my_file.chmod(0600)
```

Intermediate Ruby:

File open flags:

(default: read)

```
File.new("my_file.txt", "r")
"r"         -> read only; start at beginning of file
"r+"        -> read and write; start at beginning of file
"w"         -> write only; truncate existing or create new
"w+"        -> read and write; truncate existing or create new
"a"         -> write only; append new data to the end
"a+"        -> read and write; append new data to the end
```

If creating a new file

```
File.new("my_new_file.txt, "w+", 0644)
```

Intermediate Ruby:

Writing to a file:

```
.puts
.print
.write
```

nb These will overwrite your file if .pos is not at the end, make sure you seek to the end of the file first.

```
my_file.pos = my_file.size`
```

Intermediate Ruby:

Locking the file:

.flock(LOCKING_CONSTANT)

Locking_constants

FILE::LOCK_EX -> Exclusive; only one process.
FILE::LOCK_NB -> Non-Blocking; return nil if the call would block
FILE::LOCK_SH -> Shared Lock
FILE::LOCK_UN -> Unlock

Intermediate Ruby:

File metadata:

- .atime -> Access time
- .ctime -> Change time for directory information about file
- .mtime -> Modification time

File Types:

- File.ftype *file_name*
- File.chardev? *file_name*
- File.blockdev? *file_name*
- File.file? *file_name*
- File.directory ? *file_name*
- File.exist? *file_name*

Delete:

- File.delete *file_name*