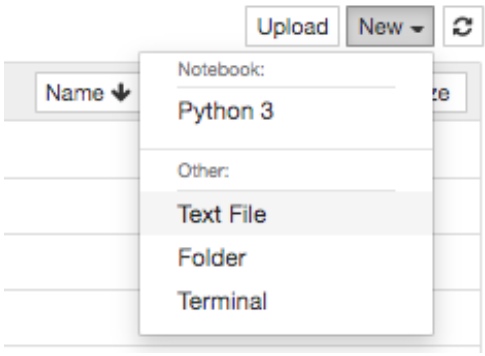
File I/O

Membuat file txt dengan jupyter



Rename nama file : missal : IO_contoh

Membuaka File :

```
Syntax
file object = open(file_name [, access_mode][, buffering])
```

Menutup file

```
Syntax
fileObject.close();
```

```
contoh_file = open("IO_contoh.txt", "w")
print("nama file : ", contoh_file.name)
print("tutup atau tidak ", contoh_file.closed)
print("mode buka : ", contoh_file.mode)
contoh_file.close()
```

Menulis file

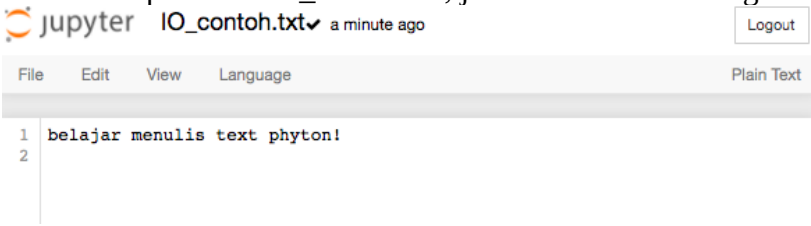
```
Syntax
fileObject.write(string);
```

Membaca file

```
Syntax
fileObject.read([count]);
```

```
contoh_file = open("IO_contoh.txt", "w")
contoh_file.write( "belajar menulis text phyton!\n")
contoh_file.close()
```

Perhatikan pada file IO_contoh.txt, jika di buka akan menghasilkan



Coba ketikan :

```
contoh_file = open("IO_contoh.txt", "w")
contoh_file.write( "belajar menulis text phyton!\n")
contoh_file.write( "Menyenangkan sekali !\n")
contoh_file.write( "Pusing Ga ya !\n")
contoh_file.close()
```

Kemudia refresh IO_contoh.txt

Percobaan 1 membaca file :

```
contoh_file = open("IO_contoh.txt", "r+")
str = contoh_file.read(7)
print(str)
posisi = contoh_file.tell()
print ("posisi : ", posisi)
position = contoh_file.seek(0,1)
str = contoh_file.read(7)
print(str)
```

Percobaan 2 membaca file:

```
contoh_file = open("IO_contoh.txt", "r+")
str = contoh_file.read()
print(str)
posisi = contoh_file.tell()
print ("posisi : ", posisi)
position = contoh_file.seek(0,0)
str = contoh_file.read(7)
print(str)
```

Tugas :

1. tuliskan beberapa daftar berbagai mode membuka file
Cobalah dan tuliskan perbedaannya.
2. Cobalah beberapa fungus untuk memproses dan memanipulasi file, tunjukan perubahannya.

Fungsi untuk memanipuasi file
file.close()
file.flush()
file.fileno()
file.isatty()
next(file)
file.read([size])
file.readline([size])
file.readlines([sizehint])
file.seek(offset[, whence])
file.tell()
file.truncate([size])
file.write(str)
file.writelines(sequence)

Fungsi Untuk memproses file
os.access(path, mode)
os.chdir(path)
os.chflags(path, flags)
os.chmod(path, mode)
os.chown(path, uid, gid)
os.chroot(path)
os.closerange(fd_low, fd_high)
os.dup(fd)
os.dup2(fd, fd2)
os.fchdir(fd)
os.fchmod(fd, mode)
os.fchown(fd, uid, gid)
os.fdatasync(fd)
os.fdopen(fd[, mode[, bufsize]])
os.fpathconf(fd, name)
os.fstat(fd)
os.fstatvfs(fd)
os.fsync(fd)
os.ftruncate(fd, length)
os.getcwd()
os.getcwdu()
os.isatty(fd)
os.lchflags(path, flags)
os.lchmod(path, mode)
os.lchown(path, uid, gid)
os.link(src, dst)
os.listdir(path)
os.lseek(fd, pos, how)
os.lstat(path)
os.major(device)
os.makedev(major, minor)

os.makedirs(path[, mode])
os.minor(device)
os.mkdir(path[, mode])
os.mkfifo(path[, mode])
os.mknod(filename[, mode = 0600, device])
os.open(file, flags[, mode])
os.openpty()
os.pathconf(path, name)
os.pipe()
os.popen(command[, mode[, bufsize]])
os.read(fd, n)
os.readlink(path)
os.remove(path)
os.removedirs(path)
os.rename(src, dst)
os.renames(old, new)
os.rmdir(path)
os.stat(path)
os.stat_float_times([newvalue])
os.statvfs(path)
os.symlink(src, dst)
os.tcgetpgrp(fd)
os.tcsetpgrp(fd, pg)
os.tempnam([dir[, prefix]])
os.tmpfile()
os.tmpnam()
os.ttyname(fd)
os.utime(path, times)
os.walk(top[, topdown = True[, onerror = None[, followlinks = False]]])
os.write(fd, str)