



# Archivos

## File

# importo las librerías

import os

import os.path

import pickle

Import io

El primer paso para trabajar con archivos  
en Python,  
es importar las bibliotecas necesarias.

Veremos 4

# importo las librerías

import os

import os.path

import pickle

Import io

OS es una biblioteca que toda la información que maneja la hace en formato string,

por lo cual NO siempre será útil  
si necesitamos saber cantidades

o

verificar que un archivo exista (por ejemplo)

# importo las librerías

import os

import os.path

import pickle

Import io

PATH es una biblioteca que toda la información que maneja la hace en formato numérico o booleano

por lo cual SERA muy útil  
si necesitamos saber cantidades  
o  
verificar que un archivo exista (booleano)

# importo las librerías

import os

import os.path

import pickle

import io

PICKLE es la biblioteca que nos permitirá transmitir REGISTROS por el bus de datos.

Desde la ram a la memoria auxiliar,  
para guardarlo  
PICKLE.DUMP

ó

Mover registros de la memoria auxiliar  
hacia la ram,  
para leerlos  
PICKLE.LOAD

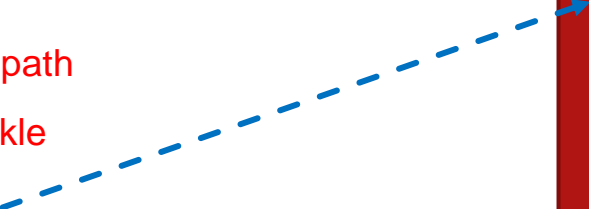
# importo las librerías

import os

import os.path

import pickle

Import io



IO es una biblioteca  
que nos permitirá acceder a los archivos  
de forma directa  
con la instrucción  
SEEK

# RAM



Para poder vincular  
los dos tipos de memorias,  
se trabajará con el bus de datos  
por donde viajará

**UN**  
registro

bus



Memoria  
Auxiliar

```
# importo las librerías
```

```
import os
```

```
import pickle
```

```
import os.path
```

```
Import io
```

```
# define el tipo de dato registro
```

```
class alu:
```

```
    def __init__(self):
```

```
        self.legajo = 0
```

```
        self.nombre = ""
```

```
        self.aula = 0
```


Por tal motivo el segundo paso  
que deben realizar  
es definir el tipo de dato record  
mediante la declarativa

CLASS



Finalizada la declarativa, comienza el programa principal

Y la acción que deben hacer es verificar si el archivo existe físicamente en la memoria auxiliar  
○  
Hay que crearlo



C-102	Navacerrada	400
C-305	Collado Mediano	350
C-215	Becerril	700
C-101	Centro	500
C-222	Moralzarzal	700
C-201	Navacerrada	900
C-217	Galapagar	750
C-110	Centro	600
C-218	Navacerrada	700

Memoria Auxiliar

Los pasos son dos

- 1) Guardar en un nombre muy cortito (para facilitar el trabajo), el URL o la dirección física donde estará guardado el archivo {arch-fisico}
- 2) Y como el archivo físico debe ser abierto, para poder trabajarlo, esa apertura quedará guardada en un nombre, (también muy cortito) que represente el archivo lógico o se uso interno del programa {arch-logico}.

La relación entre archivo físico y lógico, es solo una vinculación nemotécnica que se aplica en el proceso de programación para su posterior ejecución.

### URL ó dirección de memoria física

c:\\ayed\\alumnos.dat

arch-fisico = "c:\\ayed\\alumnos.dat" {dirección física}

arch-logico = **open** (arch-fisico, "w+b") {nombre}

La función interna **open**, crea y deja disponible el file (archivo), para poder acceder a los datos.

Contiene 2 parámetros:

- el 1ero es el nombre del archivo que puede contener su ruta,
- el 2do es el formato de apertura del mismo



Memoria Auxiliar

# importo las librerías

import os

import pickle

import os.path

Import io

# define el tipo de dato registro

**class** alu: { en Python}

**def** \_\_init\_\_(self):

self.legajo = 0

self.nombre = ""

self.aula = 0

# PP

# debo declarar variables archivo y registro

arch-fisico = "c:\\ayed\\alumnos.dat"

arch-logico = open (archivo-fisico, "w+b")

Reg = alu()

Volviendo a nuestro programa  
El próximo paso es el inicio del programa principal

Y es el momento de declarar  
las variables de trabajo

```
# importo las librerías
```

```
import os
```

```
import pickle
```

```
import os.path
```

```
Import io
```

```
# define el tipo de dato registro
```

```
class alu: { en Python}
```

```
    def __init__(self):
```

```
        self.legajo = 0
```

```
        self.nombre = ""
```

```
        self.aula = 0
```

```
# PP
```

```
# debo declarar variables archivo y registro
```

```
arch-fisico = "c:\\ayed\\alumnos.dat"
```

```
arch-logico = open (archivo-fisico, "w+b")
```

Hago los dos pasos para abrir el archivo

Y me queda  
la variable lógica,

como archivo de trabajo

# importo las librerías

import os

import pickle

import os.path

Import io

# define el tipo de dato registro

**class** alu: { en Python}

**def** \_\_init\_\_(self):

self.legajo = 0

self.nombre = ""

self.aula = 0

# PP

# debo declarar variables archivo y registro

arch-fisico = "c:\\ayed\\alumnos.dat"

arch-logico = open (archivo-fisico, "w+b")

Reg = alu()

Debo crear la variable de trabajo

Registro

Para poder moverme por el bus de datos

Que sea del tipo declarado en CLASS

# importo las librerías

import os

import pickle

import os.path

Import io

# define el tipo de dato registro

**class** alu: { en Python}

**def** \_\_init\_\_(self):

self.legajo = 0

self.nombre = ""

self.aula = 0

# PP

# debo declarar variables archivo y registro

arch-fisico = "c:\\ayed\\alumnos.dat"

arch-logico = open (archivo-fisico, "w+b")

Reg = alu()

En el momento de la apertura OPEN  
Deben definir bajo que modalidad abren el  
archivo

Ver el video de los formatos

"w+b"

"r+b"

"a+b"

# importo las librerías

import os

import pickle

import os.path

Import io

# define el tipo de dato registro

**class** alu: { en Python}

**def** \_\_init\_\_(self):

self.legajo = 0

self.nombre = ""

self.aula = 0

# PP

# debo declarar variables archivo y registro

arch-fisico = "c:\\ayed\\alumnos.dat"

arch-logico = open (archivo-fisico, "w+b")

Reg = alu()

En el momento de la apertura OPEN  
Deben definir bajo que modalidad abren el  
archivo

Ver el video de los formatos

"w+b"

"r+b"

"a+b"

Por lo cual se pueden equivocar  
si lo abren de forma "w+b"

sin validar que el archivo realmente exista,

lo cual los llevará  
a perder todo lo cargado

Lo correcto  
es

```
# importo las librerías
```

```
import os
```

```
import pickle
```

```
import os.path
```

```
Import io
```

```
# define el tipo de dato registro
```

```
class alu: { en Python}
```

```
    def __init__(self):
```

```
        self.legajo = 0
```

```
        self.nombre = ""
```

```
        self.aula = 0
```

```
# PP
```

```
# debo declarar variables archivo y registro
```

```
arch-fisico = "c:\\ayed\\alumnos.dat"
```

```
if os.path.exists (arch-lógico):
```

```
    arch-logico = open (archivo-fisico, "r+b")
```

```
    else
```

```
        arch-logico = open (archivo-fisico, "w+b")
```

```
Reg = alu()
```

Lo correcto es preguntar si existe

En caso afirmativo se lo abre en modo "r+b"  
Para iniciar desde la posición cero  
sin borrar nada

De lo contrario  
se lo crea y se lo abre en modo "w+b"



# RAM

## Leer un registro

`pickle.load (arch-logico)`



C-215	Becerril	700
-------	----------	-----

bus



C-102	Navacerrada	400
C-305	Collado Mediano	350
C-215	Becerril	700
C-101	Centro	500
C-222	Moralzarzal	700
C-201	Navacerrada	900
C-217	Galapagar	750
C-110	Centro	600
C-218	Navacerrada	700

Memoria  
Auxiliar

Las próximas acciones ya será trabajar sobre el archivo,  
y para ello se requerirá del uso  
del BUS de datos

En el caso de las lecturas del archivo, LOAD  
se harán desde la memoria auxiliar a la memoria ram,  
enviando UN record  
por el bus de datos  
mediante la librería PICKLE

# RAM



Para las escrituras en el archivo, DUMP  
se harán desde la ram a la memoria auxiliar,  
enviando UN record  
por el bus de datos  
mediante la librería PICKLE

bus

C-215	Becerril	700
-------	----------	-----

C-102	Navacerrada	400
C-305	Collado Mediano	350
C-215	Becerril	700
C-101	Centro	500
C-222	Moralzarzal	700
C-201	Navacerrada	900
C-217	Galapagar	750
C-110	Centro	600
C-218	Navacerrada	700

## Escribir un registro

`pickle.dump (reg, arch-logico)`



Memoria  
Auxiliar

```
# importo las librerías
```

```
import os
```

```
import pickle
```

```
import os.path
```

```
Import io
```

```
# define el tipo de dato registro
```

```
class alu: { en Python}
```

```
    def __init__(self):
```

```
        self.legajo = 0
```

```
        self.nombre = ""
```

```
        self.aula = 0
```

```
# PP
```

```
# debo declarar variables archivo y registro
```

```
arch-fisico = "c:\\ayed\\alumnos.dat"
```

```
if os.path.exists (arch-lógico):
```

```
    arch-logico = open (archivo-fisico, "r+b")
```

```
    else
```

```
        arch-logico = open (archivo-fisico, "w+b")
```

```
Reg = alu()
```

Pensemos que el archivo no existe, por lo cual hay que crearlo, está vacío, con el puntero apuntando al byte cero.

La idea es cargar ese archivo con diferentes registros

```
# importo las librerías
```

```
import os
```

```
import pickle
```

```
import os.path
```

```
Import io
```

```
# define el tipo de dato registro
```

```
class alu: { en Python}
```

```
    def __init__(self):
```

```
        self.legajo = "0"
```

```
        self.nombre = ""
```

```
        self.aula = 0
```

```
# PP
```

```
# debo declarar variables archivo y registro
```

```
arch-fisico = "c:\\ayed\\alumnos.dat"
```

```
if os.path.exists (arch-lógico):
```

```
    arch-logico = open (archivo-fisico, "r+b")
```

```
    else
```

```
        arch-logico = open (archivo-fisico, "w+b")
```

```
Reg = alu()
```

```
# para guardar registros en un archivo abierto
```

```
reg.legajo = input("legajo") {diseño el primer registro en RAM}
```

```
While reg.legajo <> "0":
```

```
    reg.nombre = input("nombre")
```

```
    reg.nombre = reg.nombre.ljust(30) {ajusto todos los string a la misma longitud}
```

```
    reg.aula = input("aula")
```

```
    pickle.dump (reg, arch-logico) {guardo el registro en el archivo}
```

```
    reg.legajo = input("legajo") {diseño un nuevo registro en RAM}
```

```
arch-logico.flush() {aseguro que no queda pendiente ningún registro en el bus }
```

```
arch-logico.close() {debo cerrarlo si no lo uso}
```

```
# - - - - -
```

La finalizar la carga, puedo asegurarme que el bus de datos descargó el último record en el archivo, con FLUSH y debo cerrarlo - CLOSE porque lo tenía abierto en modo creación.

```
# importo las librerías
```

```
import os
```

```
import pickle
```

```
import os.path
```

```
Import io
```

```
# define el tipo de dato registro
```

```
class alu: { en Python}
```

```
    def __init__(self):
```

```
        self.legajo = "0"
```

```
        self.nombre = ""
```

```
        self.aula = 0
```

```
# PP
```

```
# debo declarar variables archivo y registro
```

```
arch-fisico = "c:\\ayed\\alumnos.dat"
```

```
if os.path.exists (arch-lógico):
```

```
    arch-logico = open (archivo-fisico, "r+b")
```

```
    else
```

```
        arch-logico = open (archivo-fisico, "w+b")
```

```
Reg = alu()
```

```
# para guardar registros en un archivo abierto
```

```
reg.legajo = input("legajo") {diseño el primer registro en RAM}
```

```
While reg.legajo <> "0":
```

```
    reg.nombre = input("nombre")
```

```
    reg.nombre = reg.nombre.ljust(30) { ajusto todos los string a la misma longitud}
```

```
    reg.aula = input("aula")
```

```
    pickle.dump (reg, arch-logico) { guardo el registro en el archivo}
```

```
    reg.legajo = input("legajo") {diseño un nuevo registro en RAM}
```

```
arch-logico.flush() {aseguro que no queda pendiente ningún registro en el bus }
```

```
arch-logico.close() {debo cerrarlo si no lo uso}
```

```
# - - - - -
```

Si quiero saber que me quedó guardado  
Debo recorrerlo, pero para eso , debe estar abierto  
**OPEN**  
Bajo el modelo de lectura

```
# importo las librerías
```

```
import os
```

```
import pickle
```

```
import os.path
```

```
Import io
```

```
# define el tipo de dato registro
```

```
class alu: { en Python}
```

```
    def __init__(self):
```

```
        self.legajo = "0"
```

```
        self.nombre = ""
```

```
        self.aula = 0
```

```
# PP
```

```
# debo declarar variables archivo y registro
```

```
arch-fisico = "c:\\ayed\\alumnos.dat"
```

```
if os.path.exists (arch-lógico):
```

```
    arch-logico = open (archivo-fisico, "r+b")
```

```
    else
```

```
        arch-logico = open (archivo-fisico, "w+b")
```

```
Reg = alu()
```

```
# para guardar registros en un archivo abierto
```

```
reg.legajo = input("legajo") {diseño el primer registro en RAM}
```

```
While reg.legajo <> "0":
```

```
    reg.nombre = input("nombre")
```

```
    reg.nombre = reg.nombre.ljust(30) {ajusto todos los string a la misma longitud}
```

```
    reg.aula = input("aula")
```

```
    pickle.dump (reg, arch-logico) {guardo el registro en el archivo}
```

```
    reg.legajo = input("legajo") {diseño un nuevo registro en RAM}
```

```
arch-logico.flush() {aseguro que no queda pendiente ningún registro en el bus }
```

```
arch-logico.close() {debo cerrarlo si no lo uso}
```

```
# - - - - -
```

```
# para recorrer un archivo mediante su lectura
```

```
arch-logico = open (archivo-fisico, "r+b") {abierto modo lectura}
```

```
X = os.path.getsize (arch-fisico) {obtener el fin del archivo en byte}
```

```
While arch-logico.tell() < X: {verificar si el file point no llegó al final del archivo}
```

```
    reg = pickle.load (arch-lógico) {traigo por el bus de datos un registro a la variable}
```

```
    print (reg.legajo) {muestro los campos de la variable record que está en la RAM}
```

```
    print (reg.nombre)
```

```
    print (reg.aula)
```

```
arch-logico.close() {debo cerrarlo si no lo uso}
```