El futuro digital es de todos

MinTIC

Complemento

Fundamentos de Programación II

Por: Willian Michel Vélez Candia

Misión TIC 2022

UNIVERSIDAD EL BOSQUE

El futuro digital es de todos

MinTIC

Temario - II

Introducción.

Estructuras de Control.

1. ¿Qué son las Estructuras de Control?
2. ¿Cuáles son?

Listas

1. ¿Qué son las Listas?

Archivos

1. ¿Qué son los Archivos?

MisiónTIC 2022

UNIVERSIDAD EL BOSQUE

El futuro digital es de todos

MinTIC

Pequeña - Introducción

El futuro digital es de todos

MinTIC

Primer Tema:

Tomado de:https://www.freepik.es/vector-premium/concepto-chico-confundido_10217031.htm

¿Qué son las Estructuras de Control?

El futuro digital es de todos

MinTIC

¿Qué son las Estructuras de Control?

Tomado de:https://www.freepik.es/vector-premium/concepto-chico-confundido_10217031.htm

Las **estructuras de control**, son instrucciones que permiten romper la secuencialidad de la ejecución de un programa; esto significa que una **estructura de control** permite que se realicen unas instrucciones y omitir otras, de acuerdo a la evaluación de una condición.

El futuro digital es de todos

MinTIC

¿Qué son las Estructuras de Control?

¿Cuales son?

El futuro digital es de todos

MinTIC

¿Qué son las Estructuras de Control?

Secuenciales

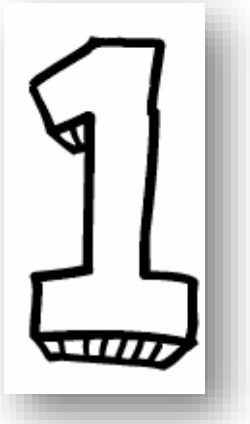
Condicionales

Iterativas

El futuro digital es de todos

MinTIC

¿Qué son las Estructuras de Control?



Secuenciales

Condicionales

Iterativas

El futuro digital es de todos

MinTIC

¿Qué son las Estructuras de Control?

Secuenciales

Condicionales

Iterativas

El futuro digital es de todos

MinTIC

¿Qué son las Estructuras de Control?

Secuenciales

Condicionales

Iterativas

¿Qué son las Estructuras de Control?

Secuenciales

Las instrucciones se ejecutan, una después de la otra, en el orden en que están escritas, es decir, en secuencia.

Este proceso se conoce como ejecución secuencial.

Ejemplo

Flujo normal de proceso, del programa hacia abajo

¿Qué son las Estructuras de Control?

Condicionales

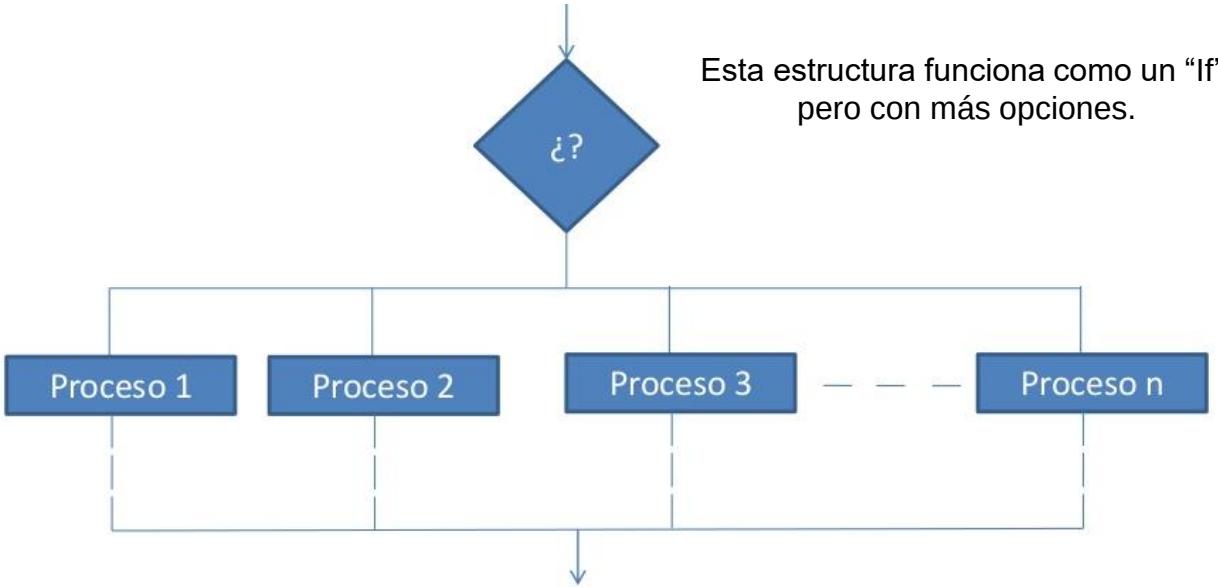
Las estructuras condicionales, ejecutan un bloque de instrucciones u otro, o saltan a un subprograma o subrutina, según se cumpla o no una condición (if-else, switch-case).

El futuro digital es de todos | MinTIC

¿Qué son las Estructuras de Control?

Condicionales

Estructura de control: Switch Case



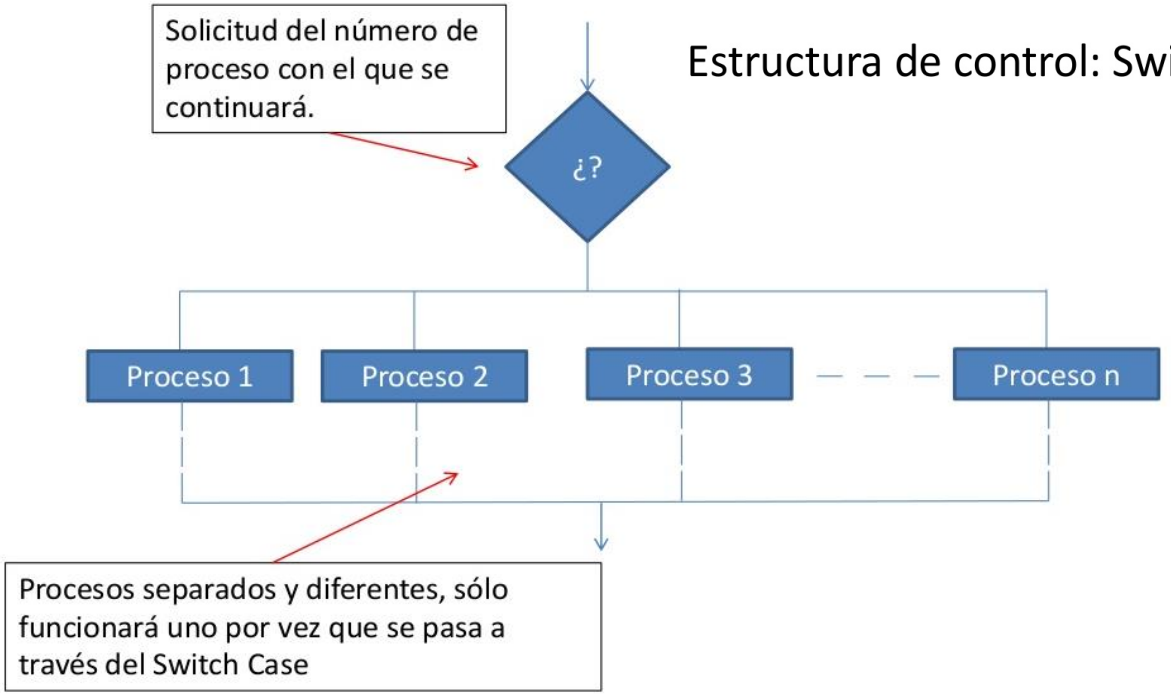
Esta estructura funciona como un “If”, pero con más opciones.

El futuro digital es de todos MinTIC

¿Qué son las Estructuras de Control?

Condicionales

Estructura de control: Switch Case



¿Qué son las Estructuras de Control?

Condicionales

¿Para qué sirve la estructura Switch Case?

Es una estructura de control que funciona como “If’s” anidados, es decir una condición dentro de otra.

Se usa mayormente para crear menús.

El futuro digital es de todos

MinTIC

¿Qué son las Estructuras de Control?

Iterativas

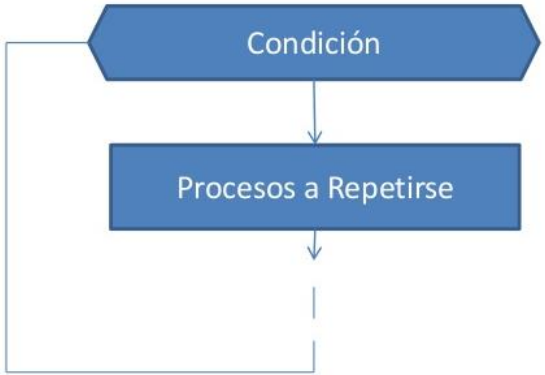
Las estructuras de control iterativas o de repetición, inician o repiten un bloque de instrucciones, si se cumple una condición o mientras se cumple una condición (while-for).

¿Qué son las Estructuras de Control?

Iterativas

Estructura Iterativa: While

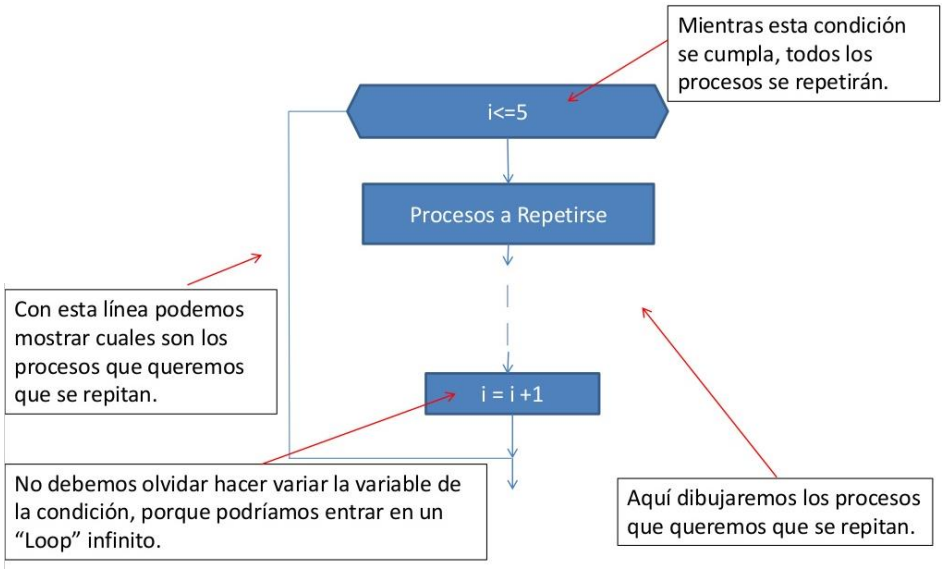
Este tipo de estructura la dibujaremos así:



¿Qué son las Estructuras de Control?

Iterativas

Analizando la Estructura: While

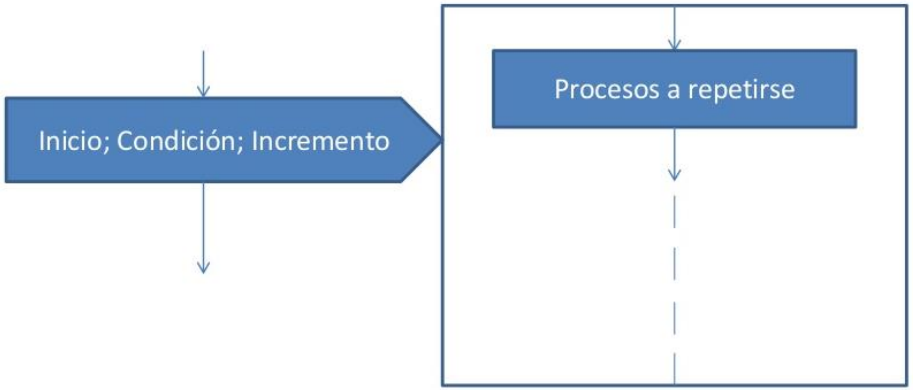


El futuro digital es de todos | MinTIC

¿Qué son las Estructuras de Control?

Iterativas

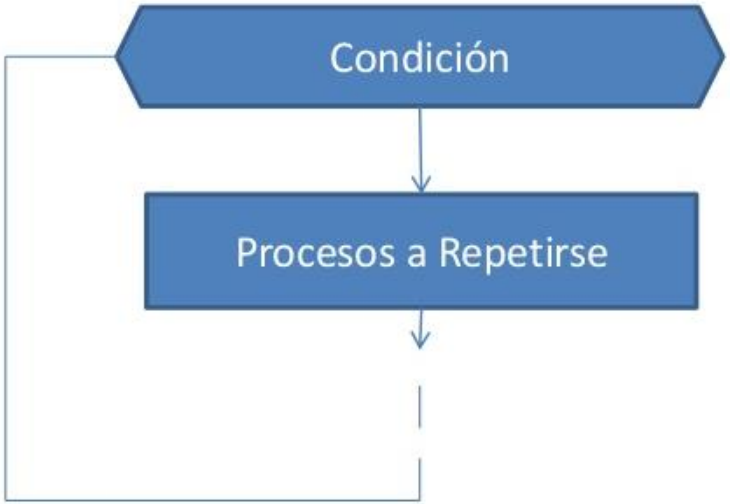
¿Cuál es la diferencia entre las estructuras While y For?



La estructura “For”, repite los procesos una cantidad determinada de veces, es decir, que conocemos cuantas veces se repetirán los procesos dentro de él.

¿Qué son las Estructuras de Control?

Iterativas



Y la estructura “While”, repite los procesos MIENTRAS la condición se cumpla y no siempre se tiene el control de cuantas veces se repetirá.

El futuro digital es de todos

MinTIC

Segundo Tema:

Listas

El futuro digital es de todos

MinTIC

Segundo Tema:

Antes de Continuar

DATO: Es una representación simbólica (numérica, alfabética, algorítmica, etc.) de un atributo o variable cuantitativa. Los datos describen hechos empíricos, sucesos y entidades.

Un dato es la representación de algo, que puede ser cuantitativa o cualitativa, que indica un valor que se le asignan a las cosas y se representa a través de una secuencia de símbolos, números o letras y esos pueden ser almacenados, guardados, de diferentes formas, dependiendo del requerimiento o la necesidad

El futuro digital es de todos

MinTIC

Segundo Tema:

Antes de Continuar

En programación es indispensable determinar a qué tipo o categoría corresponden los datos con los que se va a trabajar. Cada conjunto de datos de un tipo específico y se manipula de diferente manera para obtener los resultados deseados

Numérico

Entero. Tipo de dato formado por una variable numérica que no cuenta con parte decimal.

Real. Tipo de dato formado por una variable numérica que puede contar con parte decimal.

Texto

Carácter. Tipo de dato formado por una unidad o símbolo que puede ser una letra, un número, una mayúscula o un signo de puntuación.

Cadena. Tipo de dato formado por un conjunto de caracteres dispuestos de forma consecutiva que se representa entre comillas.

Lógico

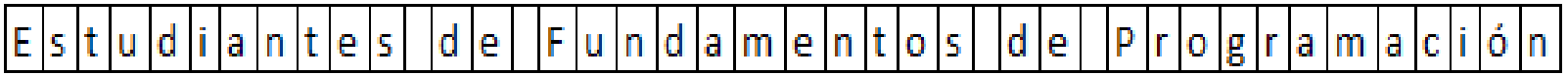
Boolean. Tipo de dato que puede representar dos valores: verdadero o falso.xzz6xxgbMoij

Segundo Tema:

Cadenas

Una cadena es una secuencia de caracteres. Puedes acceder a los caracteres de uno en uno con el operador corchete “[“ “]”

cadena = [‘Estudiantes de Fundamentos de Programación‘]



El futuro digital es de todos

MinTIC

Segundo Tema:

Cadenas

Recorriendo una cadena mediante un bucle

```
cadena=['Estudiantes de Fundamentos de Programación']

indice = 0
while indice < len(cadena):
        letra = cadena[indice]
        print(letra)
        indice = indice + 1

# En la Variable almaceno el Tamaño o Longitud de la Cadena de Caracteres

print(indice)
```

Segundo Tema:

¿Que podemos hacer con las Cadenas?

1. Podemos, por ejemplo, eliminar los espacios en blanco que encontremos al principio y al final de una cadena de texto mediante el método *strip*:

" palabra ".strip()	eliminar los espacios en blanco que encontremos al principio y al final de una cadena de texto
palabra'.count('a')	'ATG'.replace('T', 'U')
'palABra'.upper()	'ATG'.find('T')
'paLABra'.lower()	len('Franklin')

El futuro digital es de todos

MinTIC

Segundo Tema:

¿Que podemos hacer con las Cadenas?

```
>>> 'Watson'.ljust(10)
'Watson    '
```

Justificación a la Izquierda

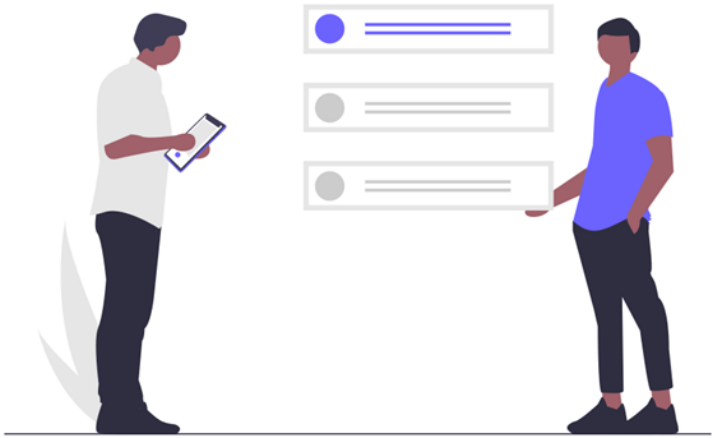
```
>>> 'Watson'.center(10)
'  Watson  '
```

Justificación en el Centro

```
>>> 'Watson'.rjust(10)
'    Watson'
```

Justificación al Derecha

¿Qué son las Listas?



¿Qué son las Listas?

¿Qué son las Listas?

Una **lista** o secuencia, es un dato, que representa una secuencia ordenada de valores, donde el mismo valor puede ocurrir más de una vez.

Así **como** una **cadena**, una **lista** es una secuencia de valores.

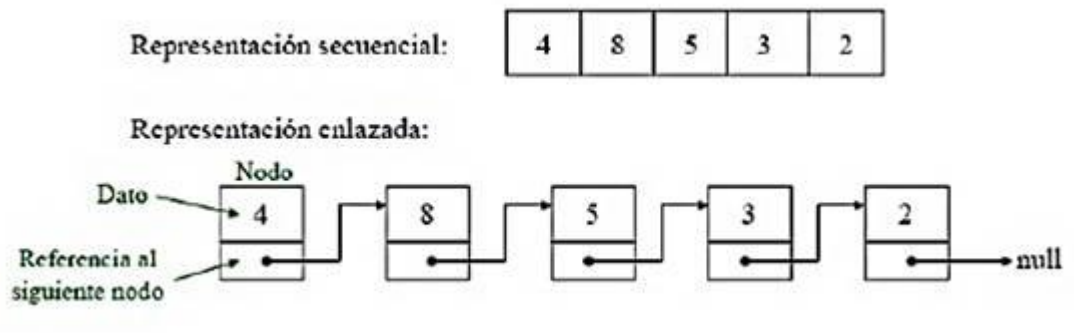
En una **cadena**, los valores son caracteres; en una **lista**, pueden ser cualquier tipo. Los valores en una **lista** son llamados elementos o a veces ítems.

Lista anidada

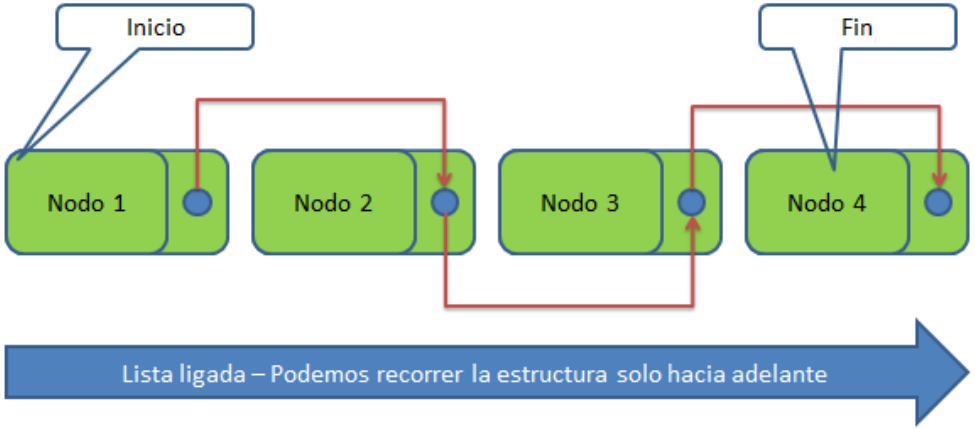
```
['spam', 2.0, 5, [10, 20]]
```

¿Qué son las Listas?

La Lista, es una estructura de datos muy importante en los lenguajes de programación donde: representa una colección de elementos ordenados. puede contener elementos repetidos. cada elemento de la lista tiene un índice que lo ubica dentro de la misma.



¿Qué son las Listas?



[10, 20, 30, 40] es una lista de 4 enteros

['Curso de Python', 'Universidad el Bosque', 'Animales Domesticos'] lista de tres cadenas

¿Qué son las Listas?

¿Que podemos hacer con las Listas?

Append()

Este método nos permite agregar nuevos elementos a una lista

```
my_list.append(10)    # [2, 5, 'DevCode', 1.2, 5, 10]

my_list.append([2,5]) # [2, 5, 'DevCode', 1.2, 5, [2, 5]]
```

¿Qué son las Listas?

¿Que podemos hacer con las Listas?

Extend()

Extend también nos permite agregar elementos dentro de una lista, pero a diferencia de append, al momento de agregar una lista, cada elemento de esta lista se agrega, como un elemento más dentro de la otra lista.

```
my_list.extend([2,5]) # [2, 5, 'DevCode', 1.2, 5, 5]
```

Remove()

El método remove va a remover un elemento que se le pase como parámentro de la lista

```
my  list.remove(2) # [5, 'DevCode', 1.2, 5]
```

¿Qué son las Listas?

¿Que podemos hacer con las Listas?

count()

Este método recibe un elemento como argumento, y cuenta la cantidad de veces que aparece en la lista

```
versiones_phone = [2.1, 2.5, 3.6, 4, 5, 6]

print "6 ->", versiones_phone.count(6)
6 -> 1
print "5 ->", versiones_phone.count(5)
5 -> 1
print "2.5 ->", versiones_phone.count(2.5)
2.5 -> 1
```

¿Qué son las Listas?

¿Que podemos hacer con las Listas?

index()

Este método recibe un elemento como argumento, y devuelve el índice de su primera aparición en la lista.

```
versiones_phone = [2.1, 2.5, 3.6, 4, 5, 6, 4]
print versiones_phone.index(4)
3
```

¿Qué son las Listas?

¿Que podemos hacer con las Listas?

insert() Este método inserta el elemento x en la lista, en el índice i.

```
versiones_phone = [2.1, 2.5, 3.6, 4, 5, 6]
print versiones_phone

[2.1, 2.5, 3.6, 4, 5, 6]

versiones_phone.insert(2, 3.7)
print versiones_phone

[2.1, 2.5, 3.7, 3.6, 4, 5, 6]
```

¿Qué son las Listas?

¿Que podemos hacer con las Listas?

pop() Este método devuelve el último elemento de la lista, y lo borra de la misma.

```
versiones_phone = [2.1, 2.5, 3.6, 4, 5, 6]
print versiones_plone.pop()
6
print (versiones_phone)
[2.1, 2.5, 3.6, 4, 5]
```

¿Qué son las Listas?

¿Que podemos hacer con las Listas?

remove() Este método recibe como argumento un elemento, y borra su primera aparición en la lista.

```
versiones_phone = [2.1, 2.5, 3.6, 4, 5, 6]
print (versiones_plone)
[2.1, 2.5, 3.6, 4, 5, 6]

versiones_phone.remove(2.5)
print (versiones_phone)
[2.1, 3.6, 4, 5, 6]
```

¿Qué son las Listas?

Como se accede a una Lista?.,
Mediante su índice

```
factura =  ['pan', 'huevos', 100, 1234]

print(factura)

resultado  ['pan', 'huevos', 100, 1234]
```

```
factura[0]
'pan'

factura[3]
1234
```

¿Qué son las Listas?

```
import random

lista_numeros = []

# Primer recorrido para leer la lista

for indice in range(1,10):
        lista_numeros.append(random.randint(1,10))

print(lista_numeros)
```

El futuro digital es de todos

MinTIC

Tercer Tema:

Archivos

El futuro digital es de todos

MinTIC

Tercer Tema:

¿Qué son los Archivos?

El futuro digital es de todos

MinTIC

Tercer Tema:

Todos los datos que un programa utiliza durante su ejecución se encuentran en sus variables, que están almacenadas en la memoria RAM del computador.

Un **archivo** es una secuencia de datos almacenados en un medio persistente que están disponibles para ser utilizados por un programa

El futuro digital es de todos

MinTIC

Tercer Tema:

Podemos dividir los archivos en dos grandes grupos.

Éstos son los **ejecutables** **y** **los no ejecutables o archivos de datos**. La diferencia fundamental entre ellos es que los primeros están creados para funcionar por si mismos y los segundos almacenan información que tendrá que ser utilizada con ayuda de algún programa.

El futuro digital es de todos

MinTIC

Tercer Tema:

¿Qué Tipos de Archivos Hay?

De texto: txt, doc, docx, etc.
De imagen: jpg, gif, bmp, png, etc.
De vídeo: avi, mp4, mpeg, mwv, etc.
De ejecución o del sistema: exe, bat, dll, sys, etc.
De audio: mp3, wav, wma, etc.
De archivo comprimido: zip, rar, tar, etc.
De lectura: pdf, epub, azw, ibook, etc.
De imagen de disco: iso, mds, img, etc.

Tercer Tema:

¿Podemos Crear Archivos desde Python?

El futuro digital es de todos

MinTIC

Tercer Tema:

¿Para Crear Archivos que debemos hacer?

Para crear un archivo de texto debemos llamar a la **función open** y **pasar dos parámetros**, el primero indica el nombre del archivo a crear y el segundo un string con el caracter "w" (la "w" write indica crear el archivo de texto):

```
archi1=open("datos.txt","w")
```

Si el archivo 'datos.txt' ya existe luego se crea uno nuevo y se borra el actual.

El archivo se crea en la misma carpeta donde se está ejecutando el script de Python, si necesitamos que se cree en otra carpeta podemos indicar el path del mismo:

```
archi1 = open("C:/Users/Lenovo/Desktop/Pyton/archivostxt/datos.txt","w")
```

Tercer Tema:

Si indicamos un path inexistente se genera un error.

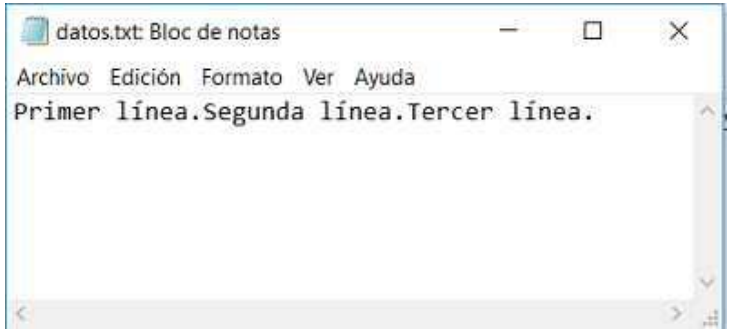
Para grabar caracteres en el archivo de texto utilizamos el método 'write' y le pasamos un string a grabar:

```
archi1.write("Primer línea.\n")
```

Mediante la sintaxis \n indicamos que debe almacenarse un salto de línea en el archivo de texto.

Si no incluimos los respectivos \n:

```
archi1.write("Primer línea.")
archi1.write("Segunda línea.")
archi1.write("Tercer línea.")
```



El futuro digital es de todos

MinTIC

Tercer Tema:

¿Para el manejo de qué debemos hacer?

Se necesita lo siguiente

1. Indicarle siempre la instrucción open.
2. Indicarle un nombre, con extensión .txt (“Porque son los mas fáciles de manipular”).
3. Luego hay que indicarle el modo, hay tres modos así:
    1. El modo “r” = abre el archivo en modo lectura, aquí hay una primera excepción que es que el archivo debe existir.
    2. El modo “w” = este modo crea el archivo y lo abre para escribir.
    3. El modo “a” = este modo si no existe, también lo crea, es como el “append” y agrega datos al final del mismo, ósea no lo sobre escribe.,

Tercer Tema:

¿Para el manejo de qué debemos hacer?

```
import os

def creararchivo():
    archivo=open(‘C:\mis_archivos\datos.txt‘,'w')
    archivo.close() # Cierro el archive en memoria

creararchivo()  # Ejecuto la función, para que cree el archivo
```

Tercer Tema:

Escribir en los Archivos

Para escribir en el archivo debemos hacer lo siguiente :

```
def escribir():

    archivo=open('datos.txt‘,'a')

    archivo.write('Chachara\n’)
    archivo.write('Chachara\n’)
    archivo.write('Chachara\n’)
    archivo.write('Chachara\n’)
    archive.close()

escribir()
```

El futuro digital es de todos

MinTIC

Tercer Tema:

Leer Archivos

Para leer el archivo debemos hacer lo siguiente :

```
import os

def leer():

    archivo_texto=open('datos.txt‘, ‘r') # Debemos Primero Abrir el Archivo, para leerlo con “r”

    texto= archivo_texto.read()          # Tambien podemos utilizar ‘.readlines’, que lee linea a linea y la
                                         # almacena el la variable texto
    archivo_texto.close()

     print(texto)
```

El futuro digital es de todos MinTIC

Tercer Tema:

Imprimir una liena de un Archivo

Para buscar algún dato en el archivo debemos hacer lo siguiente :

```
import os

def imprimir():

        archivo_texto=open('archivo.txt','r')
        lineas_texto=archivo_texto.readlines()  # Convertimmos el Archivo en una Lista para poder
                                                  realizar la busqueda más facil
                                                  # que se almacena en la variable 'lineas_texto'

        archivo_texto.close()
        print(lineas_texto[0]) # Si quiero imprimir solo la primera Linea o [1] etc-
                               # mediante el indice ya que el archivo, lo convertinos el una Lista Ok.,
```

El futuro digital es de todos

MinTIC

Tercer Tema:

Agregarle Datos o información a Archivos

Para agregar algún dato en el archivo debemos hacer lo siguiente :

```
import os

def agregar():

        archivo_texto=open('archivo.txt','a')

        archivo_texto.write('\n chachara y mas chachara cuantas veces se quiera')
        archivo_texto.write('\n Otra linea de solo chachara y mas chachara')

archivo_texto.close()
```

El futuro digital es de todos

MinTIC

!!! Exitos...

(Gracias por su Asistencia)

Por: Willian Michel Vélez Candia

Mision TIC 2022

UNIVERSIDAD EL BOSQUE