

UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ACATLÁN

ACTUARÍA



---

GEOMETRÍA ANALÍTICA

---

Por:  
Morales Ramirez Ángel Francisco

Profesor:  
Isaac Ortigoza Suárez

# Índice

1. Resumen: Programación ATS Python Vídeos 1-17	2
2. Resumen: Programación ATS Python Vídeos 18-24	4
3. Resumen: Programación ATS Python Vídeos 25-36	5
4. Resumen: Programación ATS Python Vídeos 37-38	7
5. Comentario: Programación ATS Python	8

# 1. Resumen: Programación ATS Python Vídeos

## 1-17

Python es un lenguaje de programación creado por Guido van Rossum en 1991. Actualmente Python ocupa el puesto no. 1 en el índice TIOBE. Es un lenguaje de propósito general, es decir puede programarse en el cualquier tipo de aplicaciones. Python a diferencia de otros lenguajes de programación no utiliza un punto y coma al final de cada renglón. Para asignar un valor se ocupa el signo `=`. Los valores que usa son 3: numérico `int` si es un numero entero o `float` si es un numero con valores decimales, texto si son caracteres y booleanos son valores que solo aceptan 2 opciones `True` o `False`. Sin embargo este lenguaje tiene la peculiaridad que una variable puede ser dinámica y ser de varios tipos (numérica, texto o booleana) en un mismo programa.

Para hacer comentarios en un programa se utiliza `#` si el comentario es de una linea o `'''` para comentarios de varias lineas.

Los operadores aritméticos son:

- Suma: `+`
- Resta: `-`
- Multiplicación: `*`
- División: `/`
- División Redondeando a la baja: `//`
- Residuo: `%`
- Exponente: `**`

Los operadores relacionales son aquellas que operar con numero y retornan un valor de certeza `True` o `False` (Cierto o falso). Los operadores relacionales son:

- Igualdad: `==`
- Distinto: `!=`
- Menor que: `<`
- Mayor que: `>`
- Menor o igual: `<=`
- Mayor o igual: `>=`

Los operadores lógicos son operadores que dan como resultado un valor booleano, es decir cierto o falso. Los operadores lógicos son:

- Conjunción: and
- Disyunción: or
- Negación: not

El orden de prioridad de los operadores de manera general es:

- 1.- ()
- 2.- \*\*
- 3.- \*, /, %, not
- 4.- +, -, and
- 5.- < , > , ==, ≤, ≥, !=, or

La entrada de datos se hace asignando a una variable la función input y así el programa no avanzara hasta que el usuario ingrese un valor, input por defecto da un valor de texto o cadena entonces para cambiar eso tenemos que encerrar entre corchetes el input con el valor que deseamos que nos retorne ya sea un int o float. Mientras que la salida de datos se hace escribiendo una letra f antes de las comillas y escribiendo las variables entre corchetes. Las funciones integradas funcionan para retornar una variable de la manera que deseamos por ejemplo podemos agregar a una variable abs para que no de solo el valor absoluto de sus variables o round para que haga el redondeo de sus variables. Finalmente los últimos cinco vídeos aplicando todos estos conocimientos vistos resolviendo problemas.

## 2. Resumen: Programación ATS Python Vídeos 18-24

Los siguientes vídeos hablan sobre las estructuras condicionales que ejecutan una acción u otra, en base a una condición dicotómica (Solo existen 2 casos se cumple o no se cumple la condición). La estructura if evalúa una condición y la estructura elif evalúa otra condición si ninguna de las 2 se cumple se ejecuta la condición else.

A diferencia del lenguaje de programación C++, Python no utiliza llaves para las secuencias condicionales solo usa la indentación que es como usar una sangría o tabulación hacia la derecha del código y así cuando se cumpla la condición el programa ejecutara todas las líneas que estén correctamente espaciadas.

En los vídeos prácticos resolviendo ejercicios, nos explica que en python no existe la secuencia switch y case, entonces para realizar cosas como menús o un programa donde el usuario tenga que elegir sobre una lista de opciones usamos la estructura if, elif y else haciendo uso de los operadores lógicos en los condicionales, los operadores lógicos eran not, or y and estos nos ayudan a trabajar de manera mucho mas fácil con los programas que requieren una estructura de control secuencial. También explico lo que son los condicionales anidados, básicamente ocurre cuando dentro de una secuencia condicional tenemos otra secuencia condicional.

Otras cosas que explico durante los vídeos fue que para hacer un salto de línea solo es necesario escribir `print()`, y cuando usemos variables de tipo carácter existe la función `lower` que transforma a la variable siempre a minúscula sin importar que el usuario haya escrito el valor en mayúscula del mismo modo existe la función `upper` que transforma siempre la variable a mayúscula, esto es muy útil por que al momento de programar ya no tenemos la necesidad de preocuparnos por si el usuario escribe en mayúsculas o minúsculas. Finalmente durante el resto de vídeos hicimos ejercicios bastante interesantes como determinar el número mayor de 3 números, una calculadora aritmética y un menú de un cajero automático.

### 3. Resumen: Programación ATS Python Vídeos 25-36

Es este tercer capítulo del curso de Python impartido por el canal de Youtube programación ATS nos explica lo que son las colecciones: Una colección permite agrupar varios objetos bajo un mismo nombre, existen muchos tipos de colecciones pero las principales son listas, tuplas, conjuntos y diccionarios.

- **Listas:** Se expresan entre corchetes, utilizan índices donde el primer elemento de la lista tiene el índice 0 y el último elemento de la lista tiene el índice n-1 (n es el número de elementos de la lista), esto sirve cuando solo queremos mostrar una parte de la lista para eso escribimos [a:b] y a es el elemento desde donde se muestra la lista y b es el elemento hasta el cual se muestra la lista.

En general las listas son contenedores de datos flexibles pues aceptan cualquier tipo de dato (entero, flotante, carácter, booleano, etc) incluso hasta otras listas dentro de ellas. Algunas de sus funciones más importantes son:

- `append()` para agregar un elemento (Se agrega al final de la lista)
  - `extend()` para unir 2 listas
  - `in` para verificar si un elemento pertenece a la lista
  - `count()` cuanta el número de veces que está un elemento en la lista
  - `pop()` elimina el último elemento de la lista
  - `clear()` elimina todos los elementos de la lista
  - `remove()` elimina un valor específico de la lista
  - `sort()` ordena la lista de manera ascendente
  - `sort(reverse=True)` ordena la lista de manera descendente
- **Tuplas:** Se expresan entre paréntesis y la principal diferencia a las listas es que las tuplas son colecciones inmutables, es decir, no permiten añadir, eliminar o modificar ninguno de sus elementos. Su principal ventaja es que son más rápidas y consumen menos memoria. Podemos cambiar una tupla a lista con la función `list` o podemos convertir una lista a tupla con la función `tuple`.
  - **Conjuntos:** Los conjuntos son otro tipo de colección que tiene 2 peculiaridades que lo distinguen de las otras colecciones: que sus elementos nunca tienen un orden en particular y que no aceptan elementos duplicados. Se expresan entre llaves y dentro de los conjuntos no puede haber otro tipo de colecciones. Como bien conocemos en matemáticas los conjuntos tienen varias operaciones y en Python también podemos realizar estas operaciones:

- `|` unión de conjuntos
  - `&` intersección de conjuntos
  - `-` Diferencia de conjuntos
  - `^` Diferencia simétrica de conjuntos
  - `issubset` Si es un subconjunto
  - `issuperset` Si es un conjunto que engloba
  - `frozenset` Para hacer un conjunto inmutable
- **Diccionarios:** Son una colección que al igual que los conjuntos se escriben entre llaves y se caracterizan porque cada elemento tiene 2 partes la primera es la 'keys' o clave y la segunda 'values' o valor.
  - **Pilas:** Las pilas son colecciones particulares formadas por listas que cumplen con la estructura de datos lifo (Last in first out), es decir el ultimo valor en entrar es el primero en salir, como si se tratara de una pila de libros.
  - **Colas:** Las colas son colecciones particulares formadas por listas que cumplen con la estructura de datos fifo (First in first out), es decir el primer valor en entrar es el ultimo valor en salir, como cuando hacemos cola cuando vamos al banco.

## 4. Resumen: Programación ATS Python Vídeos 37-38

En el capítulo 4 sobre el curso de Python, nos explican los ciclos: los ciclos o bucles son una estructura de control de repetición que permite ejecutar varias veces una serie de instrucciones hasta que se cumpla o mientras se cumpla cierta condición. En Python se usan principalmente 2 ciclos:

- **Ciclo While** :Se utiliza para ejecutar un bloque de instrucciones de forma continuada mientras se cumpla una condición determinada. En python haciendo uso de los operadores lógicos podemos aumentar las condiciones para que se ejecute el ciclo. Este ciclo puede llegar a ser infinito pues mientras la condición sea verdadera lo único que se ejecutara del código sera el ciclo while.
- **Ciclo For** :Permite ejecutar el ciclo de una forma iterativa, conociendo un valor específico inicial y final, en este ciclo normalmente el programador sabe el numero de repeticiones que tendrá el bucle, ya que para programarlo se hace uso de un iterador. Este ciclo for es diferente en python a los otros lenguajes de programación, pues en python se usa para recorrer de manera sencilla todo tipo de colecciones.

Recordemos que ambos ciclos para programarse necesitan evaluar una condición y que todo lo que se encuentre indentado es lo que pertenece a el ciclo que se ejecutara de forma repetida según lo que hayamos programado.



## 5. Comentario: Programación ATS Python

Tristemente este curso de Python quedo incompleto ya que el creador del canal Programación ATS, Alejandro sufrió unas complicaciones médicas que le costaron la vida. La verdad que su partida es un hecho muy lamentable pues ciertamente explicaba muy bien en sus vídeos. En mi opinión en estos vídeos le entendí muy bien y explicaba de forma breve los conceptos básicos de la programación en python.

Creo que este curso me ayudo mucho a entender los conceptos básicos de programación en python desde la sintaxis del programa, pasando por todos los tipos de datos y operadores hasta las sentencias condicionales y los bucles o ciclos, si bien creo que todavía no domino en su totalidad python este breve curso me ayudo a entender los conceptos básicos y necesarios para programar.

