Sesión 05

Algoritmos y programación

Ing. Yerman Avila

https://github.com/yavilag-SENA/algoritmos_2749613

2023

PEMDAS: Orden de operaciones

- **El orden mayor es Paréntesis,** puede usarse para forzar la ejecución de una operación.

  `ej1: 2 * (3-1)` es 4

  `ej2: (1+1)**(5-2)` es 8.

- **Siguiente es Exponenciación**

  `ej3: 1 + 2**3` es 9, no 27

  `ej4:  2 * 3**2` es 18, no 36.

PEMDAS: Orden de operaciones

- **M**ultiplication y **D**ivision tienen mayor precedencia que **A**dición y **S**ustracción

  `ej5:  2*3-1` es 5, no 4

  `ej6:  6+4/2` es 8, no 5.

- Los operadores con la misma precedencia se evalúan de izquierda a derecha (excepto exponenciación).
- Puede agregar paréntesis para cambiar el orden al operar.

  `ej7: 2*(3-1)` es 4, no 5

Concatenación (Operaciones str)

No se pueden realizar operaciones matemáticas en *strings (str)* incluso si lucen como números.

'chinese'-'food' 'eggs'/'easy' 'third'*'a charm'

Hay 2 excepciones + y *. El operador + para **concatenación** que une las palabras final de la primera con inicio de la segunda:

```
>>> palabra1 = 'Juan'
>>> palabra2 = 'Perez'
>>> palabra1 + palabra2
JuanPerez
```

Repetición (Operaciones str)

El operador * que repite la palabra o cadena y concatena directamente.

```
>>> ‘perro’*3
perroperroperro
```

Símbolos especiales unicode en función print()

Existen símbolos unicode como por ejemplo las letras griegas que pueden ser utilizadas en la función print().

```
>>> print(u"\u03C9")
ϕ
```

https://www.rapidtables.com/code/text/unicode-characters.html

Switch Case (match - case)

```
var1 = input("Seleccione Opción")

match var1:
    case "Op1":
        print("Primer caso")

    case "Op2":
        print("Caso 2")

    case "Op3":
        print("3 de 3")

    case _:

         print("Condición default")
```

https://www.codigopiton.com/como-hacer-switch-case-en-python/

https://www.freecodecamp.org/news/python-switch-statement-switch-case-example/

Switch Case (match - case)

La estructura de control *switch* – *case* no existe en Python. Una forma directa de simularlo es encadenando diversas sentencias *if elif* para todos los casos necesarios. Otra alternativa común es el uso de diccionarios, donde a cada caso se asocia una función con su código correspondiente. (...)

https://www.codigopiton.com/como-hacer-switch-case-en-python/

Ciclo For

La sentencia `for` en Python difiere un poco de lo que uno puede estar acostumbrado en lenguajes como C o Pascal. En lugar de siempre iterar sobre una progresión aritmética de números (como en Pascal) o darle al usuario la posibilidad de definir tanto el paso de la iteración como la condición de fin (como en C), la sentencia `for` de Python itera sobre los ítems de cualquier secuencia (una lista o una cadena de texto), en el orden que aparecen en la secuencia. (...)

https://docs.python.org/es/3/tutorial/controlflow.html#for-statements

While

La sentencia `while` se usa para la ejecución repetida siempre que una expresión sea verdadera. evalúa una condición y luego ejecuta un bloque de código si la condición es verdadera. El bloque de código se ejecuta repetidamente hasta que la condición llega ser o es falsa.

```
contador = 0
while contador < 10:
    # Ejecuta el bloque de código aquí

    # Siempre que el contador sea inferior a 10
```

Ejercicios:

Evidencias en **github.** En su repositorio suba los archivos correspondientes al trabajo solicitado por el instructor.

Evidencias:

- 06_ciclo_while.py
- 07_ciclo_for.py
- 08_menu_seleccion.py
- 09_calculadora.py
- 10_circuitosAC.py
- generador_graficas.ipynb #Gráficas usando **matplotlib**. Uso de **import.** **Evidencia Goolgle COLABORATORY**

06_ciclo_while.py

Cree un programa simple donde solicite al usuario un número finito de veces que quiere repetir una instrucción. Debe almacenarlo en la variable **N**. Pida al usuario que ingrese un segundo dato entero cualquiera y guárdelo en una variable auxiliar **aux1**.

Use este dato como parámetro del ciclo while y realice la siguiente operación:

valor= aux1N**

Muestre en pantalla la variable valor para cada iteración.

07_ciclo_for.py

Cree un programa simple donde solicite al usuario un número finito de veces que quiere repetir una instrucción. Debe almacenarlo en la variable **N**. Pida al usuario que ingrese un segundo dato entero cualquiera y guárdelo en una variable auxiliar **aux1**.

Use este dato como parámetro del ciclo for y realice la siguiente operación:

valor= aux1N**

Muestre en pantalla la variable valor para cada iteración.

08_menu_seleccion.py

Mediante la emulación de **switch-case** mostrada en clase (sentencias match, case); plantee un menú de selección simple donde muestre con un print el resultado de un cálculo de áreas de 5 diferentes polígonos regulares teniendo en cuenta que los parámetros solicitados al usuario son el lado del polígono y el número de lados.

El caso “default” debe asumir el lado como el radio de un círculo y entregar el área del círculo. Se usará pi=3.1416

09_calculadora.py

Cree una calculador para operar 2 números A y B con las siguientes operaciones a manera de menú de selección por parte del usuario:

- Suma **A+B**
- Resta **A-B**
- Multiplicación **A*B**
- División **A/B**
- Potencia **A^B**
- Raiz **A^(1/B**)
- Suma de potencias: **(A^B)+(B^A)**
- Promedio **(A+B)/2**
- Comparación entre **A y B: igualdad y número mayor**.

Nota: El programa debe repetirse automáticamente y solicitar nuevamente A y B para que el usuario pueda seleccionar una operación diferente. Puede usar if, else, elif, match-case, for, while, etc.

10_circuitosAC.py

El usuario selecciona si el circuito RLC a solucionar es serie o paralelo.
Ingresa los siguientes datos:

- Valor RMS (Vrms)
- Frecuencia en Hz (f)
- Resistencia (R) en Ohmnios
- Inductancia (L) en Henrios
- Capacitancia (C) en Faradios

El programa calcula:

- Impedancia Total (Magnitud y ángulo)
- Corriente Total (Magnitud y ángulo)
- Potencias Activa, Aparente, Reactiva y factor de potencia (En adelanto o atraso)

generador_graficas.ipynb

Gráficas usando **matplotlib**. Uso de **import. Evidencia Goolgle COLABORATORY**

Se explicará directamente en clase. Es necesario Internet constante y desarrollo de actividad compartida con cuenta de google.