**Modulo 0:**

·

|  |
| --- |
| En este documento encontraras cuadros de código donde se podrá copiar y pegar en tu identidad  Los cuadros de salida esperada es el resultado de los códigos antes mencionados  Los cuadros celestes solo son para los docentes y la versión de los alumnos no lo tendrán visibles.  Este material fue pasado por gptchat de openia. Aunque se ha reescrito para el nivel del curso y se han solucionado varios errores que se han encontrados. |

**Programación:**

* Concepto de programa.
* Variables objetos con un datos, tipos de datos.-¿Constantes ?
  + Numéricos (int,float,complex) Strings (str) booleanos (True, False)
  + Mutables / inmutable
* Operadores lógicos y matemáticos.
* Estructuras de toma de decisiones.
* Colecciones objetos con mas de un datos
  + listas, tuplas, diccionarios.
  + Mutables / inmutable
* Estructuras de repeticiones.
* Procedimientos y funciones.
* Algoritmos y estructuras de datos básicas.

**Organigrama:**

Inicios:

* Instalación.
* Python en el mundo.
* Proyecciones y ramas
  + IOT, AI, Business Intelligence
  + Backend / Frontend
* El lenguaje Python en el mundo IT.
* Instalación del interprete Python.
* Instalación del IDE
* Nociones de IDE, Consola – Power Shell.

|  |
| --- |
| Depende del IDE seleccionado uno encontrara la ejecución del código Python dentro de un sector de la pantalla donde se escribe el codigo o en una consola aparte.  En el caso de visual studio code se ejecuta en powershell. |

* Ejecución – Lenguajes complicados e interpretados.

|  |
| --- |
| Conceptos y ejemplos de compiladores e interpretes.  Ventajas y desventajas de cada una.  Python y su posibilidad de trabajar como interpretado y poder ser compilador |

* Estadísticas de lenguajes.

|  |
| --- |
| github / gitlab  **stack over flow**  https://insights.stackoverflow.com/trends?tags=java%2Cc%2Cc%2B%2B%2Cpython%2Cc%23%2Cvb.net%2Cjavascript%2Cassembly%2Cphp%2Cperl%2Cruby%2Cswift%2Cr%2Cobjective-c  **tiobe**  https://www.tiobe.com/tiobe-index/ |

* Multiplataforma (Windows,GNU/Linux, MacOS)
* Forma parte de MacOS, de Linux, y de Microsoft con la incorporación de Guido Van Rossum, el creador de Python, a la división de desarrolladores ademas de implementar Python para PowerBI y muchas mas interacciones.
* Primeras Funciones (concepto Built-in).
* Función incorporada print () I
* Salida de información desde el script al usuario.
* Literales "Hola mundo IT curso Python"
* Objetos Python - Concepto de objetos.

|  |
| --- |
| Objetos como contenedores de datos – atributos, y con métodos propio |

* Función incorporada print () II
  + Salida de información desde un valor guardado en un objeto.

objetos = 123

print (objeto)

o

print (f"el contenido del objeto es {objeto} ")

|  |
| --- |
| Formato antiguo y formato moderno  print (objeto)  print (f"el contenido del objeto es {objeto} ")  print (f"el contenido del objeto es {objeto=} ")  texto imprime una definición o explicación  objeto imprime el contenido en memoria guardado con la etiqueta objeto  objeto= imprime el nombre del objeto y el contenido en memoria guardado con la etiqueta objeto |

* Type

print (f"el tipo de objeto es {type(objeto)} ")

* Uso de secuencias de escape

|  |
| --- |
| Objetos como contenedores de datos – atributos, con métodos propios |

* Objetos Python - Objetos con un dato:
  + Numérico.
  + Enteros.
  + Flotantes.
  + Complejos.
  + Booleanos.
  + \*Strings, cadena de caracteres.

|  |
| --- |
| Muchos alumnos han trabajado en otros lenguajes.  Debemos aclarar que Python no declaran sus variables, el interprete lee el dato y lo clasifica  En los string no hay una unidad tipo char (carácter) todo lo que este entre comillas es un string  si es un libro bien, un carácter o nada como comillas pegadas una a otra.  Las numéricas un tienen tamaño short, long, double, etc.  Aunque se puede incrementar la definición o cantidad de decimales mediante librerías y los tipos de enteros con por ejemplo Numpy (Numerical Python) |

* Operadores aritméticos en Python

|  |
| --- |
| * En Python ademas de los estándar suma(+), resta (-), división estándar (/), división entera (//) , resto de división (%) , multiplicación (\*) * ademas   Potenciación o exponenciación (\*\*) x\*\*2 x al cuadrado  Radicación (\*\*(1/r) x\*\*(1/2) raiz cuadrada de x |

* Expresiones.
* Flujo de un programa.

|  |
| --- |
| Recalcar no solo de arriba a abajo sino también de izquierda a derecha  Cuando usemos mas de un método o en las condiciones es necesario que recuerden el concepto que cada vez se tenga mas de uno se van ejecutando en el orden aclarado. |

* Condicionales.
  + Función incorporada condiciones if elif else / match case

|  |
| --- |
| En Python **elif** no es igual que **else if** ya que se generan estructuras de anidación distintas. |

* + Simples y múltiples.
  + Anidación.

|  |
| --- |
| Tengan en cuenta que Python recién ingreso la función match case en la versión 3.10  hay muchas información en paginas web y bibliografía donde se dice que no existe. |

* Función incorporada input
  + Ingreso de información desde el script al usuario.

|  |
| --- |
| Todos los que vengan de otros lenguajes observaran que en Python solo hay una función para ingresar datos desde consola.  Esta solo ingresa objetos tipo string, debiendo luego hacer un casting. |

* Casting
* Manejo de strings.
  + Métodos y atributos
* Operadores de condición o relacionales.
* Operadores Lógicos o booleanos.
  + Métodos y atributos de objetos string
* Operadores de asignación igual (=,+=,-=,/=,//=,%=,\*=,\*\*=).
* Modificación de flujo mientras: while:

while (condición):

bloque

de

iteración

* + - else
    - break
    - continue
    - anidación
* Validación de datos

|  |
| --- |
| Solo aceptamos datos del tipo o dentro de parámetros pre-establecidos |

* Objetos Python – Colecciones - Objetos con un conjunto de datos.

|  |
| --- |
| En Python no existen como objetos básicos los arrays con sus características de que todos sus elementos deben ser del mismo tipo. |

* Objetos con múltiples datos – colecciones:
  + Contenido mutables / inmutables.
  + Ubicación indexables / no indexables.
  + Simples / conjunto pares – valores.
* slicing

|  |
| --- |
| Concepto de index |

·

|  |
| --- |
| Las colecciones anidadas por ejemplo listas, con listas internas no son matrices, solo pseudo matrices.  Las matrices reales no tienen los mismos métodos y atributos de las matrices anidadas. |

* + Pilas y colas

|  |
| --- |
| * FIFO First In > Fist Out * LIFO Last In > First Out –o- First In > Last Out |

* Modificación de flujo Bucles: for:

for dato in coleccion :

bloque

de

iteración

* break
* continue
* anidación
* enumerate
* zip

|  |
| --- |
| En Python la función solo recorre objetos iterables.  Es un for each en otros lenguajes |

·

|  |  |  |
| --- | --- | --- |
| **1ra Revisión de cursada** | | |
| **tipos de objetos** | **Funciones built in** | **operadores** |
| class int  class float  class str  ubicación  estilos  ordenan  condicionales  otros  class bool  class range  (inicio,  limite\_finaliza,  step paso) | print () |  |
| expresiones | de asignación  aritméticos |
| input() | de asignación |
| if  elif  else | comparación  booleanos o lógicos  de identidad  membresía o pertenencia |
| match  case  other | de identidad  membresía o pertenencia  index |
| While  else  break  continue | de identidad  membresía o pertenencia  index |
| for  break  continue  zip  enumerate | de identidad  membresía o pertenencia  index |
| class tuple  información | count ()  index () | slicing  index |
| class list  agregar contenido  eliminar contenido  ordenar contenido  información  otros | append ()  insert ()  extend ()  pop ()  remove ()  clear ()  sort ()  reverse ()  index ()  count ()  copy() Ver alias | slicing  index |
| class dict | get ()  keys ()  values ()  items ()  pop ()  popitem ()  clear ()  fromkeys  setdefault update  copy() Ver alias | Claves (keys)  Valores (values)  Claves, Valores |

·