**1- Preguntas: Topic General**

**¿Qué es Python?**

**¿Cómo se define Python?**

* Es un lenguaje de programación de alto alto nivel.

**¿Cuáles son las características de Python?**

**¿Qué características tiene Python?**

* Lenguaje de alto nivel.
* Tipado dinámico y fuerte: distingue de una manera muy clara las diferencias entre los tipos de variables, no es necesario declarar el tipo de la variable antes de ejecutar el programa, ya que el lenguaje de programación establece el tipo de variable durante la ejecución del programa.
* Orientado a objetos: permite la sobrecarga de constructores, herencia multiple, encapsulacion, interfaces, polimorfismo.
* Open Source: es código abierto.
* Fácil de aprender.
* Librería estándar muy amplia.
* Interpretado.
* Versatil: funciona para crear diferentes tipos de aplicaciones, ej: Web, IA, Servidor....
* Posee una gran comunidad.
* Multiplataforma: la ejecución de programación creados con este lenguaje de programación se puede hacer tanto en Sistemas Operativos como Windows, Mac o Linux/UNIX.

**¿Qué es un lenguaje de alto nivel?**

**¿Qué son los lenguajes de alto nivel?**

**¿Cuál es la diferencia entre un lenguaje de alto nivel y uno de bajo nivel?**

**¿Qué es un lenguaje de bajo nivel?**

**¿Qué son los lenguajes de bajo nivel?**

* Existen lenguajes de alto nivel y de bajo nivel. Un lenguaje de programación de bajo nivel, es aquel que cuya sintaxis se asemeja más a lo que es el código máquina que maneja una computadora, es decir los 1s y los 0s. Mientras que un lenguaje de alto nivel es aquel que se asemeja más al texto de un lenguaje natural (inglés, español, etc) que a un lenguaje máquina (1s y 0s). Python tiene una gran similitud con el idioma inglés, lo que lo hace más sencillo de aprender para personas que no están acostumbradas a programar.

**¿Qué es un lenguaje de programación?**

**¿Para qué sirve un lenguaje de programación?**

**¿Para qué se usa un lenguaje de programación?**

**¿Para qué se utilizan los lenguajes de programación?**

* Es la manera con la que se puede dar órdenes a una computadora, pero para ser entendido por la computadora debemos de hablar en un “idioma” que la computadora entienda. Las computadoras son máquinas muy veloces para procesar información y realizar tareas, pero necesitan de alguien que le diga que hacer, están esperando órdenes.

**¿Qué lenguajes de programación hay?**

**¿Cuáles lenguajes de programación hay?**

**¿Cuántos lenguajes de programación hay?**

* Existen una gran cantidad de lenguajes de programación entre ellos Go, R, Julia, Ruby, Haskel, Ada, C++, Perl, Pascal, Python, Prolog, C#, Java, Javascript.

**¿Qué se puede hacer con Python?**

**¿Para qué sirve Python?**

**¿Para qué puedo usar Python?**

**¿Qué puedo hacer con Python?**

* Básicamente cualquier cosa que las computadoras puedan hacer. Se puede hacer desde un sitio web, un videojuego, un software científico y mucho más. ¡Deja volar tu imaginación!

**Origen de Python**

**¿Cuándo nació Python?**

**¿Cuándo se inventó Python?**

**¿Cuándo surgió Python?**

**¿Cuál es el origen de Python?**

* Python inició a finales de los 80s y principios de los 90s.

**¿Cuántas versiones de Python existen?**

**¿Cuáles versiones de Python existen?**

**¿Cuántas versiones hay de Python?**

**¿Cuántas versiones tiene Python?**

**¿Cuántas versiones existen de Python?**

* Existen varias versiones de Python desde la 0.9.0, 1.0, 1.2, 1.4, 2.0, 2.1, 2.2, 2.6, 3.0… Actualmente pueden encontrarse aplicaciones realizadas con Python 2 pero se busca hacer el cambio a la versión utilizada actualmente, que es Python 3.

**¿Quién fue el creador?**

**¿Quién diseñó Python?**

**¿A quién se le ocurrió hacer Python?**

**¿Quién creó Python?**

**¿Quién inventó Python?**

**¿Quién fue el inventor de Python?**

* Python fue inventado por un holandés llamado Guido Van Rossum.

**¿Qué dio origen a este lenguaje?**

**¿Por qué se creó Python?**

**¿Cómo se originó Python?**

**¿Por qué se originó Python?**

* Guido Van Rossum estaba trabajando con un sistema operativo llamado Amoeba junto con una interfaz de usuario llamada Bourne Shell, estaba presentando un problema de integración de esta interfaz con el sistema operativo, por lo que decide inventar un nuevo lenguaje de programación que comunicara a los dos de una forma más fluida. De esta manera nace Python.

**¿De dónde viene el nombre Python?**

**¿Por qué se llama Python?**

**¿Por qué le pusieron Python?**

**¿Por qué ese lenguaje de programación se llama así?**

* El logotipo de Python viene de las serpientes Python, el cual muestra dos serpientes que forman un símbolo de Yin Yang, por lo cual se cree que viene de esta serpiente, sin embargo el nombre del lenguaje viene del gusto de Guido Van Rossum por la serie de comedia Monty Python.

**¿En qué sistema operativo se puede utilizar?**

**¿En cuáles sistemas se puede utilizar Python?**

**¿En cuáles sistemas operativos puedo instalar Python?**

* Python es multiplataforma por lo que se puede usar en Linux, Windows o Mac.

**¿Qué desventajas tiene Python?**

**¿Cuáles son las desventajas de Python?**

**¿Cuáles desventajas tiene Python?**

* Manejo de números a un alto nivel.

**¿Qué es un programador?**

**¿Por qué son importantes los programadores?**

**¿Qué hace un programador?**

**¿Cuál es el trabajo de un programador?**

* Las computadoras son máquinas muy veloces y poderosas, capaces de realizar cosas increíbles, pero necesitan de alguien que le dé las órdenes, y esa persona es un programador, es quien sabe el lenguaje que entiende la computadora y con el cual le puede decir que hacer y cómo hacerlo.

**¿Qué es una variable?**

**¿Qué son las variables?**

* Es un espacio que se reserva en la memoria de la computadora donde se va a almacenar un valor el cual puede cambiar durante la ejecución del programa.

**¿Cómo declarar una variable en Python?**

**¿Cómo se declara una variable en Python?**

**¿Cómo puedo declarar una variable en Python?**

**¿Cómo se escribe una variable en Python?**

* No pueden empezar con un número.
* No pueden ser palabras claves como if, else, and, or, entre otras.
* Tienen que representar relativamente el elemento en cuestión.

**¿Cuáles tipos de variables hay en Python?**

**¿Cuáles son las variables que tiene Python?**

1. Numéricos: enteros (int), coma flotante (float), complejos
2. Textos (String): van entre comillas dobles o simples
3. Booleanos: True o False

**¿Cuáles operadores se pueden utilizar en Python?**

**¿Cuáles operadores tiene Python?**

**¿Cuáles son los operadores de Python?**

1. Aritméticos: son las operaciones matemáticas. Estos son muy utilizados para realizar cálculos en programación. Entre ellos están la suma (+), resta (-), multiplicación (\*), division (/), exponente (\*\*), módulo (%), división entera (//).
2. Comparación: permite comparar dos condiciones. Igual que (==), diferente que (!=), mayor que (>), menor que(<), mayor o igual que (>=), menor o igual que (<=)
3. Lógicos: permite evaluar más de una condición a la vez. AND, OR, NOT
4. Asignación: permite asignar valores a variables. Igual (=), incremento (+=), decremento (-=)...
5. Especiales: utilizados para comprobar si un valor se encuentra dentro de una secuencia de valores. IS, IS NOT, IN, NOT IN.

**¿Cómo se declara un comentario en Python?**

**¿Cómo pongo un comentario en Python?**

**¿Cómo escribir un comentario?**

**¿Cómo hago para poner un comentario en Python?**

* Se puede hacer un comentario de una sola línea, para esto se usará el símbolo # antes del comentario. También se puede hacer un comentario de varias líneas, para esto se ponen tres comillas simples o dobles en una línea de código, luego lo que se quiere dejar de comentario y luego finalizar con otra línea de tres comillas.

**¿Para qué funcionan los comentarios?**

**¿Para qué se usan los comentarios en Python?**

**¿Por qué son importantes los comentarios?**

**¿Por qué tengo que usar comentarios?**

* Los comentarios son muy útiles al momento de programar ya que se utilizan para dejar notas o explicaciones de la función que tienen ciertas líneas de código, esto sirve para recordatorio al mismo programador o a otros programadores que puedan verlo.

**¿Qué es la identación en Python?**

**¿Para qué sirve la indentación en Python?**

**¿Por qué es importante la indentación?**

**¿Para qué se usa la indentación?**

* Al momento de programar se recomienda hacer una tabulación en ciertas líneas de código para distinguir lo que pertenece a un bloque.

**2- Preguntas: Preparación Entorno de Trabajo**

**¿Saber si mi computadora tiene instalado python?**

* En el caso de Linux, en la terminal escribir el siguiente comando:
  + python --version

**¿Cuál es el primer paso para empezar a programar?**

Revisar si la computadora tiene instalado python, para ello ingresamos a la terminal.

**¿Cómo corro Python en mi computador?**

**¿Cómo puedo instalar Python en mi computadora?**

* Python es un programa, que se instala y corre de la misma manera que lo haría un programa estándar, esto dependerá del sistema operativo.

**¿Qué es un sistema operativo?**

* Es el programa complejo que sirve de base no solo para manejar el hardware del computador, sino además provee servicios a las aplicaciones que son instaladas en él.

**¿Cómo puedo instalar Python en Windows?**

* Primero es necesario obtener el ejecutable de la página oficial de Python.

<https://www.python.org/downloads/>

* Ejecuta el instalador.

**¿Cómo puedo instalar Python en Mac?**

**¿Cómo puedo instalar Python en UNIX?**

**¿Cómo puedo instalar Python en Solaris?**

**¿Cómo puedo instalar Python en FreeBSD?**

**¿Cómo puedo instalar Python en Centos?**

**¿Cómo puedo instalar Python en RedHat?**

**¿Cómo puedo instalar Python en Ubuntu?**

**¿Cómo puedo instalar Python en Debian?**

**¿Cómo puedo instalar Python en Linux?**

* Para todas las versiones se podría compilar el código de Python, sin embargo esto no le permitiría sacar el máximo provecho del manejador de paquetes, además podría ser una tarea laboriosa dependiendo del nivel de conocimiento. Lo más recomendable es dependiendo de la versión de Linux/UNIX, usar el manejador de paquetes correspondiente para hacer la instalación.

**¿Python es soportado en Docker?**

**¿Puedo instalar Python en Docker?**

* Python es soportado en Docker, y puede hacerle un pull a la imagen usando el comando:

# docker pull quay.io/python-devs/ci-image

* Puede correr la imagen usando:

# docker run quay.io/python-devs/ci-image:master python3.8 -c "import sys; print(sys.version)"

* Para más detalles puede mirar en la página oficial:

<https://gitlab.com/python-devs/ci-images/tree/master>

**¿Qué es un IDE?**

* Integrated Development Environment o IDE, es el software utilizado para crear un programa, este a diferencia de un editor básico de texto, tiene ciertas características que hacen que el desarrollador sea más eficiente y efectivo a la hora de programar una aplicación, que puede ir desde un script básico hasta un programa completo.

**¿Cuáles IDE se pueden usar con Python?**

* Existe una gran variedad de IDEs, dependerá en gran medida cuál la empresa recomiende o con la que el programador se sienta más cómodo.
* Algunos IDEs son: IDLE (el que viene por defecto con Python), Pycharm , Sublime, Spyder,Visual Studio Code, Atom.

**¿Cuál es la ventaja de tener un entorno virtual?**

**¿Cuál es la diferencia entre un IDE y entorno virtual?**

**¿Cuál es la diferencia entre pip y pip3?**

**¿Qué es pip ?**

**¿Para qué sirve pip ?**

**¿Qué es pyPI ?**

* Pip es el manejador de paquetes por defecto usado en Python, dependiendo de la versión de Python se usaría pip(Python 2) or pip3(Python 3).
* Con pip es posible instalar paquetes diferentes en distintos entornos virtuales, todos dentro del mismo servidor.

**¿ Cómo desinstalar un paquete usando pip ?**

**¿ Cómo instalar un paquete en un entorno virtual ?**

**¿ Hacer el upgrade de un paquete dentro de un entorno virtual ?**

**¿ Cómo instalar un paquete usando pip ?**

* Para instalar/desinstalar un paquete usando pip use la siguiente sintaxis:

# pip3 install <nombre del paquete>

# pip3 uninstall <nombre del paquete>

* Es posible hacer un upgrade de un paquete usando la siguiente sintaxis:

# pip3 install --upgrade <nombre del paquete>

* Si es necesario instalar varios paquetes puede usar una lista, separada por enters, e importarla de la siguiente manera.

# pip3 install -r <lista de paquetes>

**3- Preguntas: Estructuras de Control**

**¿Qué es una estructura de Control?**

* Define la manera en la que bajo cierta condición, se ejecutaría o no, un determinado paso o secuencia de instrucciones, el mismo que puede ser repetitivo o no.

**¿Qué es una condición en Python?**

**¿Cómo se evalúan condiciones en Python?**

* Una condición, es el mecanismo mediante el cual las características de uno o varios elementos son evaluados usando operadores condicionales, esta evaluación tendrá como resultado un True o Falso.

**¿Qué tipos de operadores existen?**

* Dependiendo del tipo de dato los operadores pueden ser:

\* Operadores aritméticos

\* Operadores de asignación

\* Operadores de comparación

\* Operadores lógicos

\* Operadores de identidad

\* Operadores de pertenencia

\* Operadores binarios

**¿Qué es un ciclo anidado?**

* Un ciclo anidado es cuando al menos una estructura de control, es introducida dentro de otra. El programa se ejecuta de manera ordinal recorriendo desde arriba la primera estructura, y profundizando conforme las condiciones se van cumpliendo.

**¿Qué es recursión?**

* Es cuando una función se llama a sí misma, esto hasta que una condición delimite su ejecución.

**¿Qué es un ciclo infinito en Python?**

* Un ciclo infinito ocurre cuando no existe una condición que delimite la ejecución.
* Un ciclo infinito no necesariamente está mal en todos los casos, alguna veces es necesario crear uno. Ej. Un (while True:) esperado una interacción externa para ejecutar determinado bloque de código.

**¿Cómo hago un case en Python?**

* Dado que no existe la palabra reservada case, la alternativa es crear un arreglo asociativo, más concretamente un diccionario.

Ejemplo:

def case\_demo(argument):

case = {

1: "Perro",

2: "Gato",

3: "Perico",

4: "Pato",

5: "Oso",

6: "Murciélago",

}

print(case.get(argument, "Invalid month"))

case\_demo(4)

Output:

Pato

**¿Cómo recorrer un while en Python?**

**¿Qué es un while?**

* Un while es una estructura de control que se ejecuta mientras la condición sea verdadera, en caso opuesto saldrá del ciclo.

while (condición verdadera):

relalizar\_tarea\_x

**¿Cómo recorrer un for en Python?**

* Un for se ejecuta mientras existan elementos, estos elementos pueden pertenecer a una lista o ser proporcionados por una función como lo es el range.

for <variable> in <elementos>:

relalizar\_tarea\_x

**¿Qué es un if en Python?**

**¿Qué es un else en Python?**

**¿Qué es un elif en Python?**

* Un if es una estructura de control que ejecuta un bloque de código si determinada condición se cumple.
* Existe el elif que permite evaluar de manera consecutiva, en caso de que un if o elif anterior no haya cumplido la condición.
* Existe el else que se ejecuta si ningún if o elif ha cumplido la condición.

**¿Qué es un break en Python?**

* Es una palabra reservada que permite salir de una estructura de control, indiferente si en ese punto la condición es True o False. Dentro de un bucle infinito, bajo ciertas condiciones se podría lanzar un break, para salir del mismo.

**¿Qué es un continue en Python?**

* Es una palabra reservada que permite regresar al inicio de la estructura de control,no termina la ejecución del código, simplemente regresa al inicio de la estructura de control para continuar la ejecución.

**4- Preguntas: Tipos de Datos**

**¿Qué tipos de datos existen?**

Numeros enteros

Numeros de punto flotante

Numeros complejos

Cadena de caracteres

Iterables

**¿Que son numeros enteros?**

Son numeros que no tienen decimales, por ejemplo: 1 , 2, 3, 4, 5, entre otros.

**¿Que son numeros de punto flotante?**

Son números que tienen punto decimal, por ejemplo: 1.00, 2.1, entre otros.

**¿Qué son numeros complejos?**

Un número complejo está realizado por dos números decimales, uno para la parte real y otro para la parte imaginaria.

**¿Qué son cadena de caracteres?**

Un objeto *string* (simple o Unicode) es una secuencia de caracteres utilizado para guardar y representar textos.

Los *strings* en Python son inmutables.

Los objetos *string* nos van a proporcionar muchos métodos.

Para definir un *string* utilizaremos comillas simples o dobles:

**¿Que son iterables?**

El concepto de Python que generaliza la idea de "secuencia" es el concepto de "iterable".

Todas las secuencias son iterables, pero existen más iterables, como las "listas", por ejemplo.

Cuando hablamos de iterables nos referimos a "iterables acotados" (que tienen un principio y un final).

**5- Preguntas: Estructuras de Datos**

**¿Estructura de Datos?**

* Es una forma de acomodar datos que están relacionados entre sí. Estas estructuras consisten en un conjunto de nodos que podría verse como vasijas que encapsulan el valor real y cuya función es asociar los datos para así formar la estructura. El cómo se entrelazan los nodos define el tipo de estructura.

**Tipos de Estructuras**

* Arreglos,Listas,Árboles,Pilas,Colas.

**Arreglo**

* Un arreglo es una estructura de datos que permite ordenar los datos de manera secuencial y

**Lista**  
-

**Árbol**-

**Pila**

**-**

**Cola**

**Principales funciones sobre las estructuras?**

**Pop?**- Se utiliza para sacar un dato en la primera posición de una pila o cola. Al sacar el dato utilizando esta función el nodo se pierde o elimina.

**Push?**  
Inserta un elemento dentro de la pila o cola

**Insert?**

Inserta un elemento en una lista

**Delete?**

Elimina un elemento de la lista. La mayoría de las veces utiliza el índice.

**Merge?**

Crea una lista a partir de la unión de dos

**Sort?**

Acomoda un arreglo o lista de un orden dado.

Copyright 2020 Comunidad Lingüística Computacional

Por la presente se concede permiso, libre de cargos, a cualquier persona que obtenga una copia de este software y de los archivos de documentación asociados (el "Software"), a utilizar el Software sin restricción, incluyendo sin limitación los derechos a usar, copiar, modificar, fusionar, publicar, distribuir, sublicenciar, y/o vender copias del Software, y a permitir a las personas a las que se les proporcione el Software a hacer lo mismo, sujeto a las siguientes condiciones:

El aviso de copyright anterior y este aviso de permiso se incluirán en todas las copias o partes sustanciales del Software.

EL SOFTWARE SE PROPORCIONA "COMO ESTÁ", SIN GARANTÍA DE NINGÚN TIPO, EXPRESA O IMPLÍCITA, INCLUYENDO PERO NO LIMITADO A GARANTÍAS DE COMERCIALIZACIÓN, IDONEIDAD PARA UN PROPÓSITO PARTICULAR E INCUMPLIMIENTO. EN NINGÚN CASO LOS AUTORES O PROPIETARIOS DE LOS DERECHOS DE AUTOR SERÁN RESPONSABLES DE NINGUNA RECLAMACIÓN, DAÑOS U OTRAS RESPONSABILIDADES, YA SEA EN UNA ACCIÓN DE CONTRATO, AGRAVIO O CUALQUIER OTRO MOTIVO, DERIVADAS DE, FUERA DE O EN CONEXIÓN CON EL SOFTWARE O SU USO U OTRO TIPO DE ACCIONES EN EL SOFTWARE.