Python

1- ¿Cuáles son las diferencias entre los operadores aritméticos y los operadores relacionales?

-los operadores aritméticos, son aquellos que tienen la capacidad de realizar cálculos, mientras que los relacionales se encargan de comparar ciertos valores para posteriormente decidir si es verdadero o falso.

2- ¿operadores lógicos and, oro y no de Python, ¿cuál es su funcionamiento?

-and: compara si un valor contiene tanto “a” como “b” para devolver un valor verdadero, si solo contiene “a” devolverá falso

-or: analiza si un valor contiene “a” o “b” para decidir que el valor es verdadero; si no tiene ni “a” o “b” devolverá que es falso

-not: este operador comprobara si el valor es verdadero o falso, si es verdadero devolverá que el valor es falso y si es falso devolverá que es verdadero.

-se utilizan para operaciones lógicas, en las que necesites buscar valores en específico.

3- ¿Qué es la precedencia de operadores en Python y como afecta la evaluación de expresiones?

-evalúa el orden en el cual se van a realizar las operaciones correspondientes, según las reglas de Python, por ejemplo, en el caso de 5+5\*5, primero realizaría la multiplicación.

1- ¿cómo se pueden anidar las estructuras condicionales (if, elif, else) en Python?

-para realizar una cadena anidada se debe comenzar por if seguido de la condición y al final colocar else para definir qué pasa si no se cumple, sin embargo, elif ya realiza ese trabajo solo que el evalúa distintas condiciones a la vez.

2- ¿Qué son las expresiones condicionales?

-se utilizan para determinar si una condición es verdadera o falsa y realizar una operación en base al resultado

3- ¿Cómo manejar múltiples condiciones en una estructura?

-primero se debe realizar la sentencia if inicial, seguido de la condición y sus (:) para luego realizar un sangrado en el código y seguir con la función elif realizar lo mismo y terminar en la función else.

- if room == "kit":

print("Looking around in the kitchen.")

elif room == "bed":

print("Looking around in the bedroom.")

else:

print("Looking around elsewhere.")

ese ejemplo lo copie, ni idea de cómo funciona profe

java

1- ¿Qué es JavaScript y cuál es su propósito en el desarrollo web?

- JavaScript es un lenguaje de programación que se utiliza para crear páginas web interactivas. Se usa para mejorar la experiencia del usuario al permitirle realizar acciones como actualizar contenido en tiempo real, validar formularios, o crear animaciones.

Es un programa de alto nivel que se utiliza para desarrollar programas u o realizar operaciones que te faciliten el desarrollo de diversas aplicaciones y juegos.

2- ¿Cómo se incluye JavaScript en un archivo HTML?

- Para incluir JavaScript en un archivo HTML, se puede usar la etiqueta <script> con el atributo src o incrustando el código JavaScript en el HTML.

3- ¿Qué diferencias existen entre las variables declaradas con var, let y const?

- En JavaScript, las variables declaradas con var, let y const se diferencian en si son mutables o no, y en el alcance que tienen.

var

* Se usa para declarar variables mutables
* Tiene un alcance de función, es decir, se puede acceder a ellas dentro de la función en la que se declararon
* Se inicializa con el valor undefined
* Se puede modificar dentro del alcance de la función
* Se recomienda evitar su uso, ya que puede dar lugar a errores

let

* Se usa para declarar variables mutables
* Tiene un alcance de bloque, es decir, se puede acceder a ellas dentro del bloque en el que se declararon
* Se puede modificar dentro del alcance del bloque
* Evita conflictos de nombres

const

* Se usa para declarar variables inmutables, es decir, no se pueden modificar
* Tiene un alcance de bloque
* No se puede reasignar

4- ¿Cuáles son los principales tipos de datos en JavaScript y como se utilizan?

- En JavaScript podemos decir que admite ocho tipos de datos: número, BigInt, booleano, string o cadena, valor nulo, indefinido, símbolo y objeto.

Biglnt:  es un tipo de dato numérico que permite representar números enteros de gran tamaño. Se puede usar para almacenar y operar con números que están fuera del límite seguro de enteros para los números regulares.

Booleano: son un tipo de dato primitivo comúnmente usado en lenguajes de programación. Por definición, un booleano tiene 2 posibles valores: true (verdadero) o false (falso).

String: es una secuencia de caracteres que se utiliza para representar texto. Los strings son uno de los tipos de datos más utilizados en programación.

Las características de los strings en JavaScript son:

* Son inmutables, es decir, no se pueden modificar una vez creados.
* Están siempre entre comillas.
* Se almacenan en formato UTF-16.
* No existe un tipo de datos separado para caracteres unitarios.

Valor nulo: es un literal que representa la ausencia de un valor de objeto. Se trata de un valor primitivo que se usa para inicializar variables que aún no tienen un valor asignado.

Indefinido: indica que una variable no tiene valor asignado. Se trata de un valor primitivo que se asigna automáticamente a las variables que se declaran sin un valor inicial.

Símbolo: En JavaScript, hay varios símbolos que tienen un significado específico, como el operador lógico OR (||), el operador de asignación de adición (+=), el símbolo de dólar ($) y el símbolo "igual a mayor que" (=>).

Objeto: En JavaScript, un objeto es una entidad que almacena datos y propiedades. Los objetos pueden ser creados con llaves ({…}) y una lista de propiedades.

5- ¿que son los operadores aritméticos en JavaScript?

- Los operadores aritméticos en JavaScript son símbolos que permiten realizar operaciones matemáticas básicas, como sumas, restas, multiplicaciones y divisiones.

Ejemplo: suma “+”, resta”- “, multiplicación” \*”, exponenciación “\*\*”, división” /”, modulo “%”.

6- ¿Cuál es la diferencia entre los operadores == y ===?

1 - En JavaScript, el operador == (doble igual) verifica si dos operandos son iguales y devuelve un resultado booleano.

2 - En JavaScript, === es un operador de comparación estricta que verifica si dos valores son iguales y devuelve un resultado booleano.

- A diferencia del operador de igualdad (==), el operador de igualdad estricta (===) siempre considera que los operandos de distintos tipos son diferentes.

7- ¿Cómo funcionan los operadores lógicos &&, || y ! en JavaScript?

- El operador lógico NOT (!) en JavaScript invierte el valor de una expresión, es decir, cambia true a false y viceversa.

* No realiza ninguna comparación como lo hacen los operadores AND y OR.
* Se opera en solo un operando.
* Se puede usar para convertir la expresión en un booleano.
* Se puede usar para devolver el valor contrario del booleano obtenido en el último paso.

- El operador lógico OR (||) en JavaScript devuelve true si uno o más de sus operandos es verdadero. Se usa para determinar si alguna de las condiciones dadas es verdadera.

- El operador lógico AND (&&) en JavaScript devuelve true si todos los operandos son verdaderos y false si alguno de ellos es falso.

8- ¿Qué es la precedencia de operadores y como afecta la evaluación de expresiones en JavaScript?

- La precedencia de operadores en JavaScript es el orden en que se evalúan los operadores en una expresión. Esto es importante para escribir expresiones correctas y evitar errores.

Se puede redefinir el orden utilizando los paréntesis” ()”.

9- ¿cómo se pueden combinar diferentes operadores, en una sola expresión en JavaScript?

- En JavaScript, se pueden combinar diferentes operadores en una sola expresión usando los operadores lógicos, el operador coma, y el operador de propagación.

Operador coma

* El operador coma permite evaluar múltiples expresiones en una sola declaración y devuelve el resultado de la última expresión.

Operador de propagación

* El operador de propagación permite que una expresión sea expandida en situaciones donde se esperan múltiples argumentos.

Operador de concatenación

* El operador “+” se puede utilizar para sumar cadenas.
* El operador de asignación abreviada += también se puede utilizar para concatenar cadenas.