

Cuestionario 02

Carlos Emiliano Sandoval Amador
315025222

18 de enero de 2019

1. ¿Cómo funciona if?
Para utilizar **if** tenemos que colocar una sentencia booleana y **:**, después presionamos enter y idle indenta automáticamente el renglón de abajo, en este espacio se coloca lo que debe de hacer la computadora si la sentencia booleana es verdadera.
2. ¿Cómo funciona else?
Si la sentencia es falsa podemos colocar un **else** en el mismo nivel de indentación y debajo de **if**, después colocamos **:** y presionamos enter, idle indenta y en este bloque colocamos las instrucciones que sigue la computadora a continuación.
3. ¿Cómo funciona while?
La estructura de repetición funciona con dos cosas; una condición y un bloque de código. Primero evalúa si la condición es verdadera, si es así, ejecuta el bloque, de lo contrario avanza a la siguiente línea de código e ignora el bloque. Cuando la condición es cierta y el bloque ha sido ejecutado entonces se repite el proceso anterior.
4. ¿Qué es importante saber cuando usamos while?
Es importante tener la certeza de que en un número finito de veces la condición será falsa de lo contrario el programa no funcionará de manera adecuada.
5. ¿Para que sirven los espacios?
Los espacios sirven para indentar bloques que ejecutan las estructuras de control y de repetición bajo determinadas condiciones.

6. ¿Cómo puedo colocar fórmulas matemáticas en latex?
Para colocar fórmulas matemáticas tenemos que encerrar la fórmula entre signos de peso o podemos utilizar `begin` con el entorno `equation`. Para más información podemos consultar Matemáticas en la barra de herramientas.
7. ¿Cómo utilizo `format`?
Hay que colocar **`.format`** después del string seguido de un paréntesis dentro del cuál colocaremos el o los datos que queremos agregar, en lugar de usar porcentaje usamos llaves.
8. ¿Qué ventaja tiene `format` sobre `%`?
La ventaja que tiene **`.format`** sobre el porcentaje es que podemos agregar datos en orden diferente, pues podemos utilizar etiquetas para cada dato y podemos colocar un dato en más de una ocasión en el string con tan solo ponerlo una vez en **`format`**.
9. ¿Qué es un objeto?
Un objeto es un conjunto de datos y funciones relacionadas que tiene atributos y métodos.
10. ¿Qué es un método?
Un **método** es una acción que realiza el objeto.
11. ¿Qué es un atributo?
Un **atributo** es una característica del objeto.
12. ¿Qué es una clase?
Una clase es una plantilla para la creación de objetos de datos según un modelo predefinido. Cada clase es un modelo que define un conjunto de variables, los atributos y métodos apropiados para operar con dichos datos.
13. ¿Cómo se llama el objeto de una clase?
Un objeto creado a partir de la clase se denomina instancia de la clase.
14. ¿Cómo sabemos el tipo de un dato?
Para saber el tipo de dato de una variable podemos utilizar la función **`type()`**.

15. ¿Cómo cambiamos datos a entero?
Para cambiar datos a entero utilizamos **int()**.
16. ¿Cómo cambiamos datos a flotante?
Para cambiar a flotante utilizamos **float()**.
17. ¿Cómo cambiamos datos a string?
Para cambiar a string utilizamos **str()**.
18. ¿Qué paquete nos sirve para poner los contenidos en español en latex?
Para poner los contenidos en español podemos usar el paquete babel con la opción spanish.
19. ¿Con qué biblioteca podemos ejecutar comandos de bash desde python?
Con la biblioteca os.
20. ¿En qué biblioteca puedo encontrar aproximaciones de π y e?
En la biblioteca math.
21. ¿Cómo funciona bool()?
Con **bool()** podemos obtener True o False dependiendo del tipo de objeto que le demos. Con un string o una lista devuelve False si está vacía y True si tiene elementos. Con números devuelve True si no son cero y False en caso contrario.
22. ¿Cómo declaramos una lista?
Para declarar una lista tenemos que utilizar `[]` y dentro ponemos los elementos de la lista separados por comas.
23. ¿cómo funciona append()?
Con **append()** podemos agregar un nuevo elemento a una lista en el último lugar de esta. Para usarla solo tenemos que colocar el elemento deseado en el paréntesis.
24. ¿cómo funciona insert()?
Agrega un nuevo elemento a la lista en la posición que desees. Pide dos parámetros, la posición y el elemento.
25. ¿cómo funciona remove()?
Elimina el elemento que le indiquemos de la lista. El único parámetro que pide es el elemento que deseamos eliminar.

26. ¿cómo funciona `pop()`?

Con **`pop()`** podemos sacar un elemento de una lista y almacenarlo en una variable. Si no colocamos parámetro entonces se saca el último elemento de la lista. Podemos colocar la posición del elemento que queremos sacar de la lista.

27. ¿cómo funciona `len()`?

La función **`len()`** es de python y sirve para contar la longitud de una lista, es decir, cuántos elementos tiene.

28. ¿Cómo funciona `for`?

La palabra reservada **`for`** induce un ciclo de repetición con el cual podemos recorrer un arreglo en su totalidad. Para utilizar **`for`** tenemos que poner **`for`** seguido de una variable luego la palabra reservada **`in`** y un objeto iterable, es decir, una lista, un string, un rango, entre otros.