

LEA ATENTAMENTE EL TEXTO Y REALICE LA ACTIVIDAD A CONTINUACIÓN

## Difference between Compiler and Interpreter

[BI India Bureau](#) May 27, 2019, 18:10 IST



Compilers and interpreters are programs that help convert the high-level language (Source Code) into machine codes to be understood by the computers. Computer programs are usually written on high level languages. A high-level language is one that can be understood by humans. To make it clear, they contain words and phrases from the languages in common use – English or other languages for example. However, computers cannot understand high level languages as we humans do. They can only understand the programs that are developed in binary systems known as a machine code. To start with, a computer program is usually written in high level language described as a source code. These source codes must be converted into machine language and here comes the role of compilers and interpreters.

Differences between Interpreter and Compiler	
Interpreter translates just one statement of the program at a time into machine code.	Compiler scans the entire program and translates the whole of it into machine code at once.
An interpreter takes very less time to analyze the source code. However, the	A compiler takes a lot of time to analyze the source code. However, the overall time taken

overall time to execute the process is much slower.	to execute the process is much faster.
An interpreter does not generate an intermediary code. Hence, an interpreter is highly efficient in terms of its memory.	A compiler always generates an intermediary object code. It will need further linking. Hence more memory is needed.
Keeps translating the program continuously till the first error is confronted. If any error is spotted, it stops working and hence debugging becomes easy.	A compiler generates the error message only after it scans the complete program and hence debugging is relatively harder while working with a compiler.
Interpreters are used by programming languages like Ruby and Python for example.	Compilers are used by programming languages like C and C++ for example.

### How Interpreter and Compiler work?

To start with, a compiler creates the program. It will analyze all the language statements to check if they are correct. If it comes across something incorrect, it will give an error message. If there are no errors spotted, the compiler will convert the source code into machine code. The compiler links the different code files into programs that can be run such as exe. Finally, the program runs. An interpreter creates the program. It neither links the files nor generates machine code. The source statements are executed line by line while executing the program.

### Advantages and disadvantages of Interpreter and Compiler

In case of using compilers, the program codes are translated into machine code already and hence the time to execute the code is very less. On the negative side, it is not possible to change the program without going back to the source code while working with a compiler.

Interpreters make working with the source code much easier. Hence, they are highly suitable especially for the beginners. On the negative side, interpreted programs can only run on the computers that have the respective interpreters.

ELIJA LA OPCIÓN CORRECTA:

1. Este texto contiene paratextos:
  - a. Sólo verbales
  - b. Sólo cónicos
  - c. Verbales e icónicos
2. Los paratextos icónicos están representados por:
  - a. Fotografía
  - b. Esquema
  - c. Diagramación
3. Los paratextos verbales que aparecen son
  - a. Notas al pie
  - b. Índice
  - c. Título
4. Las preguntas en negrita “**How Interpreter and Compiler work?**” y “**Advantages and Disadvantages of Interpreter and Compiler**” son ...
  - a. ... paratextos icónicos.
  - b. ... paratextos verbales.
5. Una de las oraciones principales del texto es:
  - a. They can only understand the programs that are developed in binary systems known as a machine code.
  - b. Compilers and interpreters are programs that help convert the high-level language (Source Code) into machine codes to be understood by the computers.
  - c. These source codes must be converted into machine language and here comes the role of compilers and interpreters.
6. Otra de las oraciones principales es:
  - a. To start with, a compiler creates the program.
  - b. It will analyze all the language statements to check if they are correct.
  - c. The source statements are executed line by line while executing the program.
7. “...A high-level language is one that can be understood by humans...”

- a. Es una de las oraciones principales del texto.
  - b. Es una de las oraciones secundarias del texto.
8. "...The compiler links the different code files into programs that can be run such as exe.."
- a. Es una de las oraciones principales del texto.
  - b. Es una de las oraciones secundarias del texto.
9. La Introducción del texto...:
- a. ...comienza en el primer párrafo.
  - b. ...comienza desde el primer párrafo hasta la primera pregunta "How Interpreter and Compiler work"?
  - c. No existe en el mismo.
10. El Desarrollo del texto...:
- a. ...comienza en el la pregunta "How Interpreter and Compiler work"?
  - b. ... comienza en la mitad del primer párrafo.
  - c. ... comienza desde el cuadro de las diferencias entre el intérprete y el compilador.
11. La Conclusión...
- a. ...comienza en la última oración del texto.
  - b. ...comienza en el último párrafo del texto.
  - c. ...no hay en este texto.
12. Puede decirse que la sección Ventajas y desventajas del Intérprete y el compilador...
- a. ...forma parte de la conclusión.
  - b. ...forma parte del desarrollo.
  - c. ... forma parte de la introducción.
13. Una de las diferencias entre el intérprete y el compilador es que:
- a. ...el intérprete toma el análisis del código fuente mientras que el compilador analiza el código fuente.
  - b. ...al intérprete le toma menos tiempo en analizar el código fuente mientras que al compilador le lleva mucho tiempo.
14. En caso de utilizar compiladores...:
- a. ...los códigos del programa ya se traducen en código de máquina y por lo tanto el tiempo de ejecutar el programa es menor.

- b. ... los códigos del programa se traducen a la máquina de códigos y el tiempo de ejecución es menor.
15. El intérprete es...:
- a. ...eficiente en términos de memoria.
  - b. ...eficiente al generar un código intermedio.
16. Un compilador ...
- a. Siempre genera un código objeto intermedio.
  - b. Siempre genera un intermediario en el código objeto.
17. Si el intérprete detecta un error...
- a. Deja de funcionar y la depuración se vuelve más fácil.
  - b. deja funcionando la depuración que es mas fácil.
18. Si el compilador...
- a. ... genera sólo el mensaje de error, después depura el programa.
  - b. ... genera un mensaje de error sólo después de escanear el programa completo y por lo tanto, la depuración es más difícil.
19. Una de las ventajas de los intérpretes es que...
- a. ...hacen que trabajar con el código fuente de manera más fácil.
  - b. ...están trabajando con el código fuente de manera más fácil.
20. El rol de los compiladores e intérpretes es...
- a. ...convertir el código fuente al lenguaje de máquina.
  - b. ...que los códigos fuentes conviertan el lenguaje de la máquina.

TOTAL SCORE : 20 points

PASSING SCORE: 10 /20

---

**Enlace para acceder al cuestionario.**

<https://forms.office.com/Pages/ResponsePage.aspx?id=uh2eZcyzzE2HMNI4d-erey1Ur3RaY4hBs93lG1hSRc1UMEJZSjdNT0JWWkM2VIRSUE1DNUFPU0VJTt4u>

**UNLaM – DIIT – Tecnicatura en Desarrollo Web / Aplicaciones Móviles**  
**INGLÉS TÉCNICO 1 2622 /2999**  
**1era EVALUACIÓN DE APRENDIZAJES**  
**1er cuatrimestre 2020**