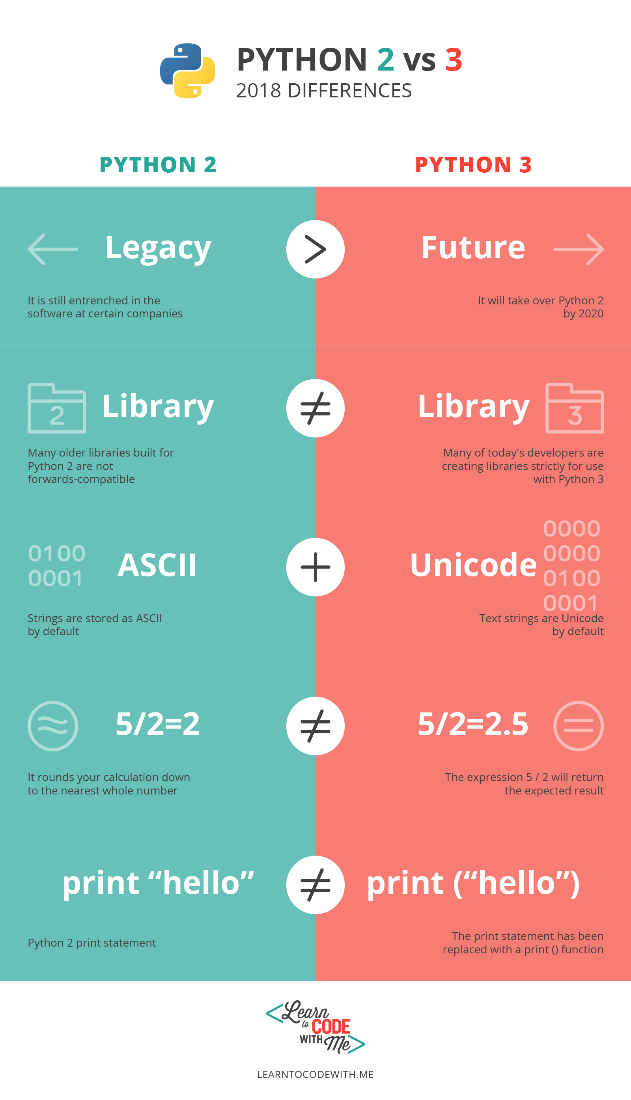
|  |  |  |
| --- | --- | --- |
| **No** | **Compiler** | **Interpreter** |
| **1** | Compiler Takes **Entire** program as input | Interpreter Takes **Single** instruction as input . |
| **2** | Intermediate Object Code is **Generated** | **No** Intermediate Object Code is **Generated** |
| **3** | Conditional Control Statements are Executes **faster** | Conditional Control Statements are Executes **slower** |
| **4** | **Memory Requirement** : **More**(Since Object Code is Generated) | **Memory Requirement** is **Less** |
| **5** | Program need not be **compiled**every time | Every time higher level program is converted into lower level program |
| **6** | **Errors** are displayed after **entire program** is checked | **Errors** are displayed for **every instruction** interpreted (if any) |
| **7** | **Example** : C Compiler | **Example** : BASIC |

E) Find 3 differences between a compiler and an interpreter.

F) Find the difference between Python 2 and 3?****

**G)** What is ASCII and UTF-8?

**ASCII :** it stands for American Standard Code for Information Interchange.

It's a method that codes [A-Z] , [a-z] and [0-9] (only American characters) in binary system , for example the character ( a ) have the code 110 0001 which equals in decimal the number 97. Also the symbol ( ! ) have the code 010 0001 which equals 33 in decimal system , and so on, it starts from 32 to 127 in decimal. Each symbol should consist of single byte with 8 bits. Life was good. Heck they even sent people to the moon! That is if you spoke English .

**UTF-8 :** it's used to encode text in any language , it came to over come the shortage in ASCII coding that only works for English language and the need of special coding for each language.

How does UTF-8 solve the problem?!  
In UTF-8, every code-point from 0–127 is stored in a single byte. Code points above 128 are stored using 2, 3, and in fact, up to 6 bytes.