2.2.1. Source Code Encoding

By default, Python source files are treated as encoded in UTF-8. In that encoding, characters of most languages in the world can be used simultaneously in string literals, identifiers and comments — although the standard library only uses ASCII characters for identifiers, a convention that any portable code should follow. To display all these characters properly, your editor must recognize that the file is UTF-8, and it must use a font that supports all the characters in the file.

It is also possible to specify a different encoding for source files. In order to do this, put one more special comment line right after the`#!` line to define the source file encoding:

```
# -*- coding: encoding -*-
```

With that declaration, everything in the source file will be treated as having the encoding *encoding*instead of UTF-8. The list of possible encodings can be found in the Python Library Reference, in the section on `codecs`.

For example, if your editor of choice does not support UTF-8 encoded files and insists on using some other encoding, say Windows-1252, you can write:

```
# -*- coding: cp-1252 -*-
```

and still use all characters in the Windows-1252 character set in the source files. The special encodingcomment must be in the *first or second* line within the file.