Strings are **not** required to be UTF-8. Go source code **is** required to be UTF-8. There is a complex path between the two.

In short, there are three kinds of strings. They are:

1. the substring of the source that lexes into a string literal.
2. a string literal.
3. a value of type string.

Only the first is required to be UTF-8. The second is required to be written in UTF-8, but its contents are interpreted various ways and may encode arbitrary bytes. The third can contain any bytes at all.

Try this on:

```
var s string = "\xFF語"
```

Source substring: `"\xFF語"` , UTF-8 encoded. The data:

```
22
5c
78
46
46
e8
aa
9e
22
```

String literal: `\xFF語` (between the quotes). The data:

```
5c
78
46
46
e8
aa
9e
```

The string value (unprintable; this is a UTF-8 stream). The data:

```
ff
e8
aa
9e
```

And for record, the characters (code points):

```
<erroneous byte FF, will appear as U+FFFD if you range over the string value>
語 U+8a9e
```