

Package utf8 go1.15.2 Latest

Published: **Sep 9, 2020** | License: [BSD-3-Clause](#) | [Standard library](#)

Doc Overview Subdirectories Versions Imports Imported By Licenses

Overview

Package utf8 implements functions and constants to support text encoded in UTF-8. It includes functions to translate between runes and UTF-8 byte sequences. See

<https://en.wikipedia.org/wiki/UTF-8>

Constants

```
const (  
    RuneError = '\uFFFD' // the "error" Rune or "Unicode replacement character"  
    RuneSelf  = 0x80     // characters below RuneSelf are represented as themselves  
    MaxRune   = '\U0010FFFF' // Maximum valid Unicode code point.  
    UTFMax    = 4         // maximum number of bytes of a UTF-8 encoded Unicode ch  
)
```

Numbers fundamental to the encoding.

func DecodeLastRune

```
func DecodeLastRune(p []byte) (r rune, size int)
```

DecodeLastRune unpacks the last UTF-8 encoding in p and returns the rune and its width in bytes. If p is empty it returns (RuneError, 0). Otherwise, if the encoding is invalid, it returns (RuneError, 1). Both are impossible results for correct, non-empty UTF-8.

An encoding is invalid if it is incorrect UTF-8, encodes a rune that is out of range, or is not the shortest possible UTF-8 encoding for the value. No other validation is performed.

func DecodeLastRuneInString

```
func DecodeLastRuneInString(s string) (r rune, size int)
```

DecodeLastRuneInString is like DecodeLastRune but its input is a string. If s is empty it returns (RuneError, 0). Otherwise, if the encoding is invalid, it returns (RuneError, 1). Both are impossible results for correct, non-empty UTF-8.

An encoding is invalid if it is incorrect UTF-8, encodes a rune that is out of range, or is not the shortest possible UTF-8 encoding for the value. No other validation is performed.

func DecodeRune

```
func DecodeRune(p []byte) (r rune, size int)
```

DecodeRune unpacks the first UTF-8 encoding in p and returns the rune and its width in bytes. If p is empty it returns (RuneError, 0). Otherwise, if the encoding is invalid, it returns (RuneError, 1). Both are impossible results for correct, non-empty UTF-8.

An encoding is invalid if it is incorrect UTF-8, encodes a rune that is out of range, or is not the shortest possible UTF-8 encoding for the value. No other validation is performed.

func DecodeRuneInString

```
func DecodeRuneInString(s string) (r rune, size int)
```

DecodeRuneInString is like DecodeRune but its input is a string. If s is empty it returns (RuneError, 0). Otherwise, if the encoding is invalid, it returns (RuneError, 1). Both are impossible results for correct, non-empty UTF-8.

An encoding is invalid if it is incorrect UTF-8, encodes a rune that is out of range, or is not the shortest possible UTF-8 encoding for the value. No other validation is performed.

func EncodeRune

```
func EncodeRune(p []byte, r rune) int
```

EncodeRune writes into p (which must be large enough) the UTF-8 encoding of the rune. It returns the number of bytes written.

func FullRune

```
func FullRune(p []byte) bool
```

FullRune reports whether the bytes in p begin with a full UTF-8 encoding of a rune. An invalid encoding is considered a full Rune since it will convert as a width-1 error rune.

func FullRuneInString

```
func FullRuneInString(s string) bool
```

FullRuneInString is like FullRune but its input is a string.

func RuneCount

```
func RuneCount(p []byte) int
```

RuneCount returns the number of runes in p. Erroneous and short encodings are treated as single runes of width 1 byte.

func RuneCountInString

```
func RuneCountInString(s string) (n int)
```

RuneCountInString is like RuneCount but its input is a string.

func RuneLen

```
func RuneLen(r rune) int
```

RuneLen returns the number of bytes required to encode the rune. It returns -1 if the rune is not a valid value to encode in UTF-8.

func RuneStart

```
func RuneStart(b byte) bool
```

RuneStart reports whether the byte could be the first byte of an encoded, possibly invalid rune. Second and subsequent bytes always have the top two bits set to 10.

func Valid

```
func Valid(p []byte) bool
```

Valid reports whether p consists entirely of valid UTF-8-encoded runes.

func ValidRune

```
func ValidRune(r rune) bool
```

ValidRune reports whether r can be legally encoded as UTF-8. Code points that are out of range or a surrogate half are illegal.

func ValidString

```
func ValidString(s string) bool
```

ValidString reports whether s consists entirely of valid UTF-8-encoded runes.