投稿日 2020年04月04日 UnicodeをUTF-8やUTF-16に変換する方法 ● UTF-8, 文字コード, unicode, UTF-16

点

### 中などに記す場合などは U+ の後に16進数でその値を続けることで表します。 Unicodeの符号空間

面

Unicodeの符号

@yasushi-jp

10進数  $0 \sim 16$  $0 \sim 255$  $0 \sim 255$ 16進数  $0x00\sim0xFF$  $0x00\sim0xFF$  $0x00 \sim 0x10$ 

Unicodeは面(0~16)、句(0~255)、点(0~255)の符号空間を持っており、Unicode符号位置を文章

### 例えば a は U+0061 、 あ は U+3042 、 鰻 は U+29E7D と表記されます。

文字

甶

第7

面

第8

面

第9

面

第

10

面

第

11

面

第

12

面

第

13

面

第

14

面

第

15

面

第

16

面

変換手順

0000) とします。

n' = yyyy yyyy yyxx xxxx xxxx

 $w_1$  = 1101 1000 0000 0000

 $w_2$  = 1101 1100 0000 0000

U+6FFFF

U+70000 -

U+7FFFF

U+80000 -

U+90000 -

U+A0000 -

U+AFFFF

U+B0000 -

U+C0000 -

U+D0000 -

U+DFFFF

U+E0000 -

U+EFFFF

U+F0000 -

U+100000 -

UnicodeをUTF-16に変換

(1) 文字のUnicode符号位置をnとします。

n が 0x10000 以上 (サロゲートペア) の場合、(3) に進みます。

U+10FFFF

U+FFFFF

U+CFFFF

U+BFFFF

U+9FFFF

U+8FFFF

未使用

未使用

未使用

未使用

未使用

未使用

未使用

面)

Supplementary Special-

purpose Plane(追加特殊用途

Private Use Plane (私用面)

Private Use Plane (私用面)

(2) n が 0x10000 より小さい (第0面) 場合、 n を16ビットの符号なし整数として表現して終了です。

(3) n' = n - 0x10000 ,  $w_1$  = 0xD800 (1101 1000 0000 0000) ,  $w_2$  = 0xDC00 (1101 1100 0000

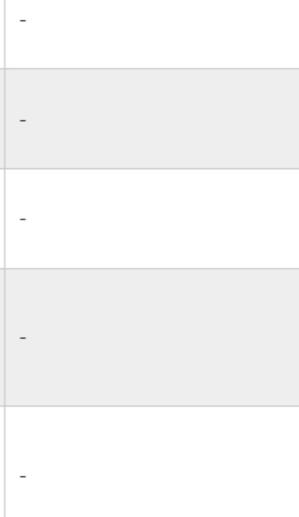
こうすると、n'は20ビット以内で表現可能となり、 $w_1$ 、 $w_2$ は下位10ビットが0となります。

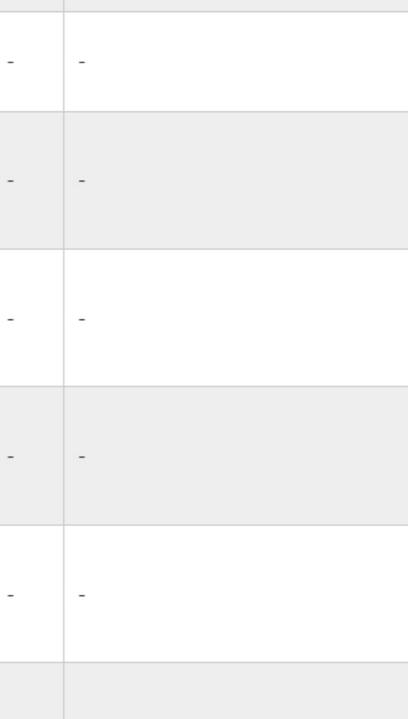
句

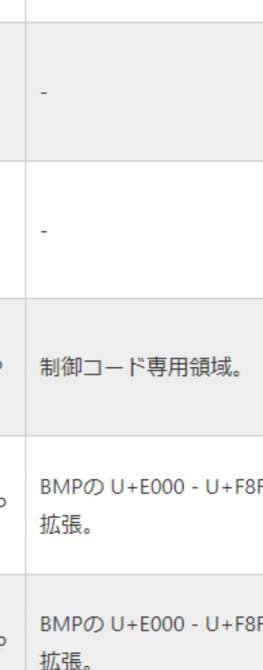
句

略称 収録されている主な文字 符号位置 名称 面

Щ	りつは旧	417	一口イン	状球と作じいるエなスチ
第0 面	U+0000 - U+FFFF	Basic Multilingual Plane(基本 多言語面)	ВМР	基本的な文字。
第1 面	U+10000 - U+1FFFF	Supplementary Multilingual Plane(追加多言語面)	SMP	古代文字や記号・絵文字類など。
第2 面	U+20000 - U+2FFFF	Supplementary Ideographic Plane(追加漢字面)	SIP	漢字専用領域。
第3	U+30000 - U+3FFFF	Tertiary Ideographic Plane(第 三漢字面)	TIP	追加漢字面に入りきらなかった漢字。また、将来的には古代漢字や甲骨文字などが収録される予定。
第4 面	U+40000 - U+4FFFF	未使用	-	_
第5 面	U+50000 - U+5FFFF	未使用	-	-
第6	U+60000 -	未使用	-	-









#### (4) n'の各ビットを $w_1$ と $w_2$ にそれぞれ10ビットずつ割り振ります。 n' = yyyy yyyy yyxx xxxx xxxx $w_1$ = 1101 10yy yyyy yyyy $w_2$ = 1101 11xx xxxx xxxx

 $w_1$ が上位サロゲート、 $w_2$ が下位サロゲートとなります。

UTF-16の変換例(あ(U+3042)の場合)

UTF-16の変換例(鰻(U+29E7D)の場合)

(3)  $n' = n - 0 \times 10000 = 0 \times 29 = 7D - 0 \times 10000 = 0 \times 19 = 7D$  (0001 1001 1110 0111 1101)

(4) n' = n - 0x10000 = 0x29E7D - 0x10000 = 0x19E7D (0001 1001 1110 0111 1101)

UTF-8のビット列 (2進)

110x xxxx 10xx xxxx

1110 xxxx 10xx xxxx 10xx xxxx

1111 0xxx 10xx xxxx 10xx xxxx 10xx xxxx

(3) 調べた範囲のUTF-8のビット列(2進)のxにUnicodeの符号のビット列を当てはめた値がUTF-8の値

0xxx xxxx

(1) n = 0x3042(2) n = 0x3042 は 0x10000 より小さいため、「あ (U+3042) 」のUTF-16のコードは「0x3042」と

なります。

(1) n = 0x29E7D

(2) n = 0x29E7D は 0x10000 以上。

 $w_1$  = 1101 1000 0000 0000

 $w_2$  = 1101 1100 0000 0000

 $w_1 = 1101\ 1000\ 0110\ 0111 = 0xD867$ 

 $w_2$  = 1101 1110 0111 1101 = **0xDE7D** 

UTF-16BEの場合、0xD867 0xDE7D

UTF-16LEの場合、0xDE7D 0xD867

(1) Unicodeの符号位置の範囲を調べます。

符号位置(16進)

U+0000 - U+007F

U+0080 - U+07FF

U+0800 - U+FFFF

数にすると「**0x61**」となります。

U+10000 - U+10FFFF

(2) Unicodeの符号位置のビット列を求めます。

節囲1

範囲2

範囲3

範囲4

となります。

UnicodeをUTF-8に変換 変換手順

# UTF-8の変換例 (a (U+0061) の場合) (1) a (U+0061) は「範囲1」となります。 (2) U+0061のビット列は「0110 0001」となります。 (3)「範囲1」のUTF-8のビット列(2進)「0xxx xxxx」に当てはめると、「0110 0001」となり、16進 UTF-8の変換例(®(U+00AE)の場合) (1)® (U+00AE)は「範囲2」となります。 (2) U+00AEのビット列は「0000 1010 1110」となります。

(3) 「範囲 2」のUTF-8のビット列(2進)「110x xxxx 10xx xxxx」に当てはめると、「1100 0010 1010

(3) 「範囲3」のUTF-8のビット列(2進)「1110 xxxx 10xx xxxx 10xx xxxx」に当てはめると、「1110

と、「1111 0000 1010 1001 1011 1001 1011 1101」となり、16進数にすると「OxFOA9B9BD」となりま

## UTF-8の変換例(あ(U+3042)の場合) (1) あ (U+3042) は「範囲3」となります。

1110」となり、16進数にすると「**0xC2AE**」となります。

(1) 鰻(U+29E7D)は「範囲4」となります。 (2) U+29E7Dのビット列は「0 0010 1001 1110 0111 1101」となります。 (3) 「範囲4」のUTF-8のビット列(2進) 「1111 0xxx 10xx xxxx 10xx xxxx 10xx xxxx」に当てはめる

UTF-8の変換例(鰻(U+29E7D)の場合)

(2) U+3042のビット列は「0011 0000 0100 0010」となります。

0011 1000 0001 1000 0010」となり、16進数にすると「OxE38182」となります。

す。 UnicodeをUTF-32に変換

Unicodeの符号位置の値を4バイト固定幅にしたものがそのままUTF-32となります。

プログラマのための文字コード技術入門 (ISBN978-4-7741-4164-0)

以上