UTF-8

1 Code point = 1-4 code units, 1 code unit = 1 byte

- Starts with 0: 1 byte = 1 code unit → We have 7 bits left — 0xxx xxxx
- Starts with 110: 2 bytes = 2 code units → We have 11 bits left — 110x xxxx 10xx xxxx
- Starts with 1110: 3 bytes = 3 code units → We have 16 bits left — 1110 xxxx 10xx xxxx 10xx xxxx
- Starts with 1111 0: 4 bytes = 4 code units → We have 21 bits left — 1111 0xxx 10xx xxxx 10xx xxxx 10xx xxxx

Examples:

- 11110 000 (data) 10011111 (data) 10001101 (data) 10001110 (data)
  - 4 byte unit code because it starts with 11110
  - → 000 011111, 001101, 001110 code point
  - → separate into bytes
  - 0x0 0x1 0xF 0x3 0x4 0xE → U+01F34E
  - → apple emoji (unicode) = U+01F34E, (UTF-8) = F09F8D8E
- 00101011 (data)
  - 1 byte unit code because it starts with a 0
  - → 0101011 matches up with ASCII
  - → ASCII value is 0x2 0xB → U+002B
  - → +
- 1100 1111 (data) 1000 0000 (data)
  - 2 bytes code unit
  - → 00111100,0000
  - → 0x3 0xC 0x0 → U+03C0
  - → Lower case π
- 00100001 (data)
  - 1 byte code unit
  - → 0100001
  - → 0x2 0x1 → U+0021
  - → !

UTF-16

1 code point = 1-2 code units, 1 code unit = 2 bytes

- First code unit starts with 11010 → We have 11 bits left — 1101 1xxx xxxx xxxx
- Second code unit starts with 110111 → We have 10 bits left — 1101 11xx xxxx xxxx
- Code points under 65536 are in a single code unit

Examples:

- 11011000 (data) 00110100, 11011101 (data) 00011110,
  - 0000,1101,0001,0001,1110
  - 0x0 0xD 0x1 0x1 0xE → U+0000D11E
  - unicode = U+0000D11E, UTF-16 = D834DD1E

UTF-32

1 code point = 1 code unit, 1 code unit = 4 bytes

- Always uses 32 bits
- same value as the character