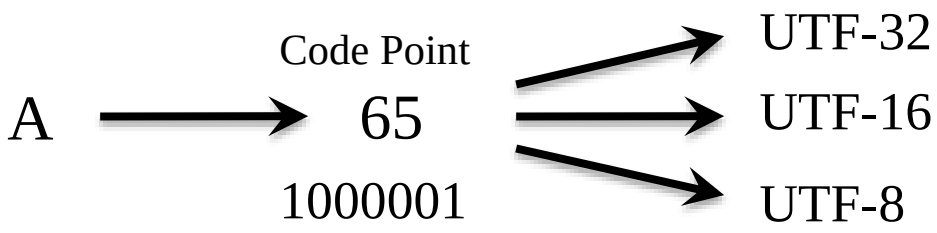
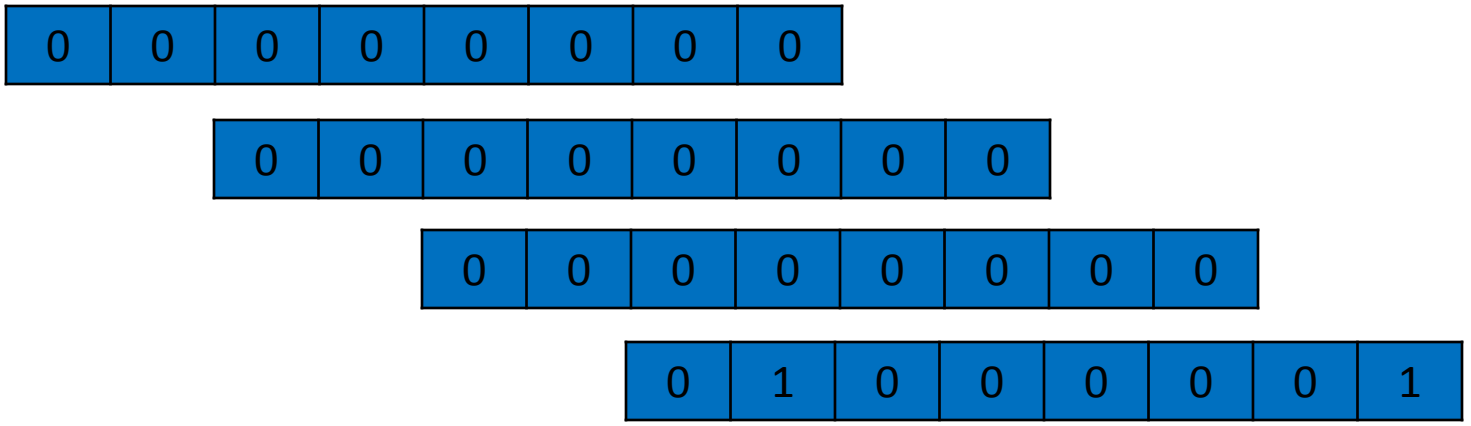
Unicode

- A standard character encoding designed to support all of the world's languages
- Unicode represents characters differently than ASCII
- Characters are mapped to a **code point**



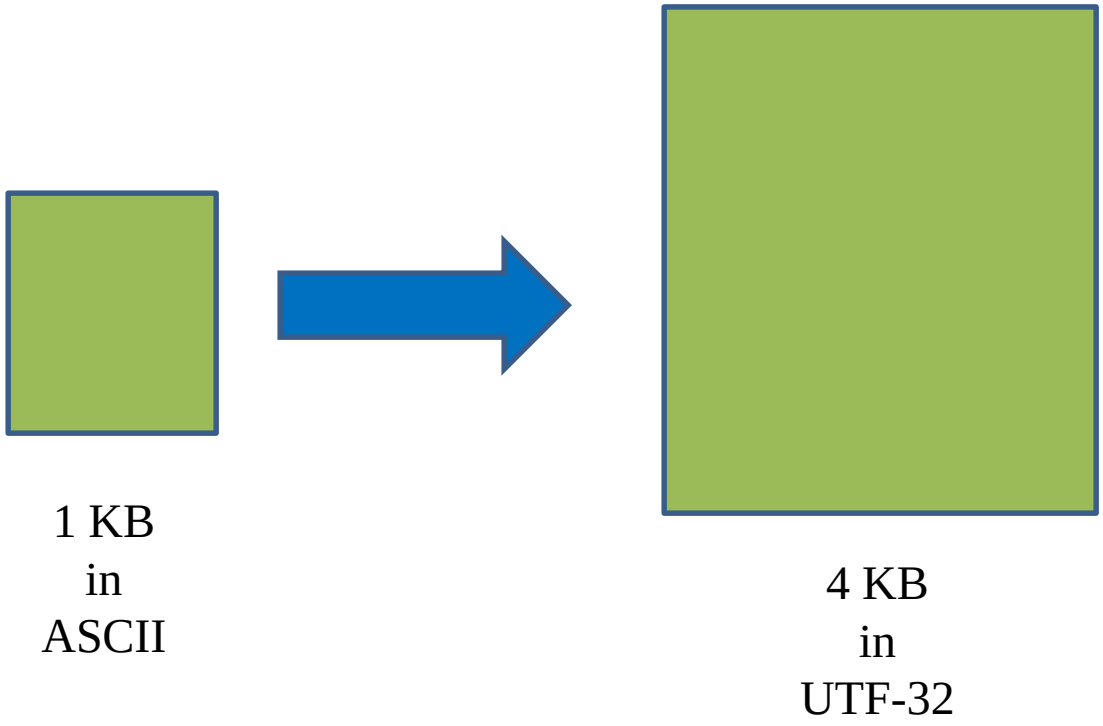
UTF-32

- Uses 4 bytes (32 bits)
- Example:
  - A (100 0001)



UTF-32

- Problem:



UTF-16

- Stores each char in either 16-bit or two 16-bit

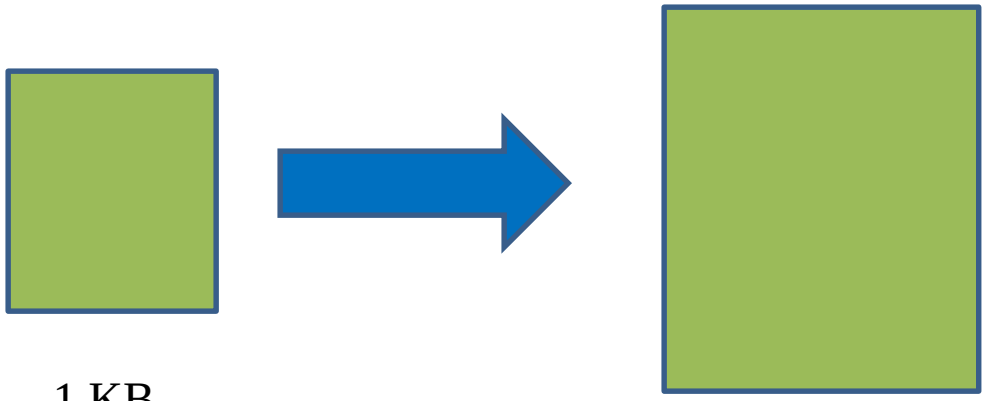
16 bits

16 bits

16 bits

UTF-16

- Problem:



1 KB
in
ASCII

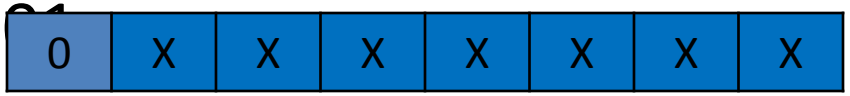
2 KB
in
UTF-16

UTF-8

- It supports every language you'll probably ever need.
- No need for Windows-1252 this and Windows-1253 that.
- Its code point range is from 0x00 to 0x10FFFF
- It uses a variable (1 to 4) byte encoding.

UTF-8 (1-byte)

- 1-byte UTF-8 is used for code points in the range 0x00 to 0x7F.
- 1-byte UTF-8 ≡ ASCII
  MSBit is 0
  code point ≡ representation
- Examples of 1-byte UTF-8:
  - “A” -> 0100 0001
  - “&” -> 0010 0110
  - “5” -> 0011 01



UTF-8 (2-byte)

- 2-byte UTF-8
  code point != representation
- The code point is broken apart into two pieces.
- The five MSBits of the code point are assigned to the first byte and the six LSBits are assigned to the second byte.

UTF-8 (2-byte)

For the first byte of 2-byte UTF-8:

- The three MSBits are set to 110
- The remaining bits are the five MSBits of the code point.

For the second byte of 2-byte UTF-8

- The two MSBits are set to 10
- The remaining bits are the six LSBits of the code point.

UTF-8 (2-byte)

Leading Byte

Continuation Byte

UTF-8 (3-byte)

- 3-byte UTF-8 is used for code points in the range 0x0800 to 0xFFFF.
- 3-byte UTF-8
  code point != representation
- The code point is broken apart into three pieces.

UTF-8 (3-byte)

- **The four MSBits of the code point are assigned to the first byte.**
- **The middle six bits are assigned to the second byte.**
- **The six LSBits are assigned to the third byte.**

UTF-8 (3-byte)

For the first byte of 3-byte UTF-8

- The four MSBits are set to 1110
- The remaining bits are the four MSBits of the code point.

For the second byte of 3-byte UTF-8

- The two MSBits are set to 10
- The remaining bits are the six middle bits of the code point.

UTF-8 (3-byte)

For the third byte of 3-byte UTF-8

- The two MSBits are set to 10
- The remaining bits are the six LSBits of the code point.

UTF-8 (3-byte)

Leading Byte

Continuation Byte

Continuation Byte

UTF-8 (4-byte)

- 4-byte UTF-8 is used for code points in the range 0x10000 to 0x10FFFF.
- 4-byte UTF-8
  code point != representation
- The code point is broken apart into four pieces.

UTF-8 (4-byte)

- The three MSBits of the code point are assigned to the first byte.
- The next six MSBits are assigned to the second byte.
- Another of the next six MSBits are assigned to the third byte.
- The six LSBits are assigned to the fourth byte.

UTF-8 (4-byte)

For the first byte of 4-byte UTF-8

- The five MSBits are set to 11110
- The remaining bits are the three MSBits of the code point.

For the second byte of 4-byte UTF-8

- The two MSBits are set to 10
- The remaining bits are the next six middle bits of the code point.

UTF-8 (4-byte)

For the third byte of 4-byte UTF-8

- The two MSBits are set to 10
- The remaining bits are the next six middle bits of the code point.

For the fourth byte of 4-byte UTF-8

- The two MSBits are set to 10
- The remaining bits are the six LSBits of the code point.

Examoles

10011100101001