UNICODE SANDWICHES

PYTHON USER GROUP FREIBURG

OMAR KOHL - 09. NOVEMBER 2016

TRANSLATE

0x5F	0x3B	0x58	0x53	0x28	0x57	0x58	0x48
95	59	88	83	40	87	88	72

What encoding is this?

SOLUTION

Every nº is the Nth word in the "History" section of the English article about the city Freiburg.

Solution: This city is Hamburg and it is German

What do we learn from this example?

DEFINITIONS

Code: System of rules to convert information into another form or representation

Encode: Converting from source into symbols for communication or storage

Decode: Converting code symbols back into a form that the recipient understands

Definitions adapted from Wikipedia

ENCODING/DECODING: A MATTER OF CONVENTION

Encoding: Converting from source to symbols.

Decoding: Converting from symbols to source.

What is the source?

Example: Text ---ASCII---> bytes (numbers)

Example: Numbers ---ASCII---> letters

TWO COMPUTERS COMMUNICATE

Microphone, Loudspeakers

- ASCII

TWO COMPUTERS COMMUNICATE

Microphone, Loudspeakers

- Custom encoding

TWO COMPUTERS COMMUNICATE

Microphone, Loudspeakers

- MP3

TWO COMPUTERS COMMUNICATE

What happens if the second partner receives a stream of bits and mistakes the encoding?

ASCII

- American Standard Code for Information Interchange
- Work on the standard began 1960
- Basis for most other character encodings
- 7 bits (1 byte with most significant bit always 0)
- Decimal: 0 - 127
- Printable characters (decimal): 32 - 126
- Example 1

ASCII "EXTENSIONS" (1 BYTE)

- latin1 (ISO 8859-1)
- ISCII (Indian characters)
- VISCII (Vietnamese characters)
- Windows 1252 (latin1 extension, sometimes wrongly called ANSI)
- IBM PC code page 437
- ISO 8859-2, ISO 8859-3, ...
- etc.

What is the main limitation of these encodings?

Example 2

LIMITATION

Nomos: From the Greek term for "law" (νόμος, nómos; pl. νόμοι, nómoi).

Omar: (Arabic: عمر, Hebrew: עומר), is a male given name of Arabic and Hebrew origin

UNICODE: MOTIVATION

- 1 byte not enough
- N bytes: Not efficient (waste of space)
- All problems in computer science can be solved by another level of indirection (David Wheeler)

UNICODE

- Idea born in 1987. Since 1991 "Unicode Consortium"
- Abstract representation (a number) for every character of every past and current human language
- Numbers: Code points (e.g. U+00F5, U+062A, U+1F31B)
- The first 256 code points match latin1
- How are these numbers converted to bytes? "We don't care" (sort of)

CONVERTING UNICODE TO BYTES: UTF

- Unicode Transformation Format
- All code points can be encoded!
- UTF-8: 1 to 4 bytes. Most common encoding for the web. Backwards compatible with ASCII.
- UTF-16: 2 to 4 bytes. Superseded UCS-2 (the original Unicode 1.0 16 bit encoding)
- UTF-32: 4 bytes (aka UCS-4)
- Example 3

THE UNICODE SANDWICH

From: Net Batchelder "Pr agmatic Unicode" PyCon 2012

PYTHON - BEST PRACTICES

- Don't rely on default encodings! They are crazy! REALLY!
- Forget 1 character == 1 byte. Nowadays this is false!
- Open text files in text mode (mode='wt', mode='rt')
- Decode as early as possible, encode as late as possible (Unicode sandwich)
- Generally be explicit about the encoding (DB connection, HTTP etc.)
- If possible only use UTF-8
- catch and handle UnicodeErrors
- Normalize Unicode data
- Example 4

PYTHON 2 VS 3: STR, UNICODE, BYTES

- Python2 str: Bytes (sequence of 8bit numbers)
- Python2 bytes: Synonym for str type(bytes()) == type(str())
- Python2 unicode: Unicode strings. Literals prefixed with u
- Python3 str: Unicode strings. No prefix for literals.
- Python3 bytes: Bytes. Literals prefixed with b
- Python3 unicode: Doesn't exist.
- Example 5

PYTHON 2 - BEST PRACTICES

- Always specify "# -*- coding: utf-8 -*-"?
- File open() is always binary. Decode early or use io.open()
- Use u'' literals or carefully use from __future__ import unicode_literals
- Use the six library for compatibility

SOURCES

- **Fluent Python** by Luciano Ramalho - August 2015 - First edition - O'Reilly - Chapter 4 "Text versus Bytes"
- **Pragmatic Unicode, or, How do I stop the pain?** Ned Batchelder - PyCon 2012 - https://www.youtube.com/watch?v=sgHbC6udIqc
- **Character encoding and Unicode in Python** Travis Fischer, Esther Nam - PyCon 2014 - https://www.youtube.com/watch?v=Mx70n1dL534
- **The Absolute Minimum Every Software Developer Absolutely, Positively Must Know About Unicode and Character Sets (No Excuses!)** Joel Spolsky - http://www.joelonsoftware.com/articles/Unicode.html