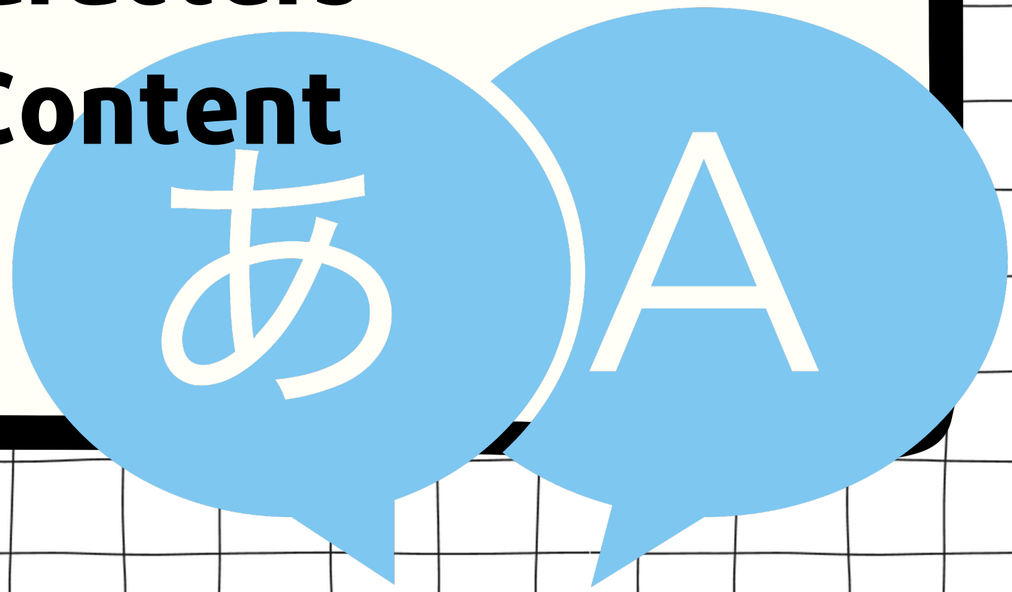
University College Dublin
Ireland's Global University

Working with Diverse Language Characters in Preserved Digital Content



Niamh Murphy
Digital Preservation Librarian

Introduction

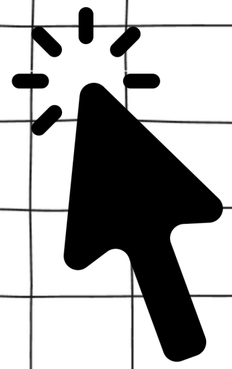
Diacritics and non-Latin scripts are fundamental to the accurate interpretation and understanding of our written languages, across the globe.

However, despite their significance, they pose notable challenges in the realm of digital preservation.

By understanding the importance of these characters and acknowledging the obstacles they present in digital preservation, we can strive to develop inclusive and equitable solutions that uphold linguistic diversity and cultural integrity in the digital landscape.

Diacritics

(á)	(à)	(ç)	(â)
Acute accent	Grave accent	Cedilla	Circumflex
(č)	(ã)	(ä)	(ā)
Haček	Tilde	Diaeresis	Macron



Non-Latin Scripts

- Arabic
- Armenian
- Chinese
- Cyrillic

- Greek
- Hebrew
- Japanese
- Korean

- N’Ko
- Tagalog
- Tamil
- Thai

Challenges in Digital Preservation

Default Optimisation:

Historically, digital environments prioritized English-based character encoding standards, leading to an unintentional exclusion of diacritics and non-Latin scripts, and subsequently inadequate support of these character sets. As a result, texts featuring diacritics and/or written in non-Latin scripts may be displayed incorrectly, garbled, or rendered illegible when transferred from one environment to another, or processed using a particular software. This is of particular relevance when working with legacy systems.

Inconsistency:

Even now, there are inconsistencies in relation to character encoding standards, schemas and normalisation forms across digital systems, platforms and software applications.
These inconsistencies can lead to diacritic erasure, mojibake, etc. all of which result in the misinterpretation or loss of meaning in text.

Placeholder Symbols:

On a related note, when lacking support, diacritics and non-Latin scripts may be replaced with placeholder symbols such as the "tofu" symbol (□) or the question mark within a diamond ().
These rendering issues also compromise the legibility and integrity of text featuring diacritics and non-Latin scripts.

Language Default

For the purpose of this presentation,
language default refers to an English language default, which impacts billions of people, and is present across countless personal and professional environments.
It is an issue that is often perpetuated without detection, but there are ample opportunities for mitigation.

Challenges and Impact:

The perpetuation of an English language default can exacerbate linguistic inequality and contribute to the marginalization of languages outside of the default.

This is evident in digital contexts where English-centric encoding standards and practice may not adequately support or represent diacritics and non-Latin scripts.

As a result, texts written in languages such as Arabic, Chinese, French, Irish, Spanish or Welsh may face misrepresentation, rendering issues or character corruption when encountered in English-centric digital environments.

Opportunities for Mitigation:

While the prevalence of an English language default presents challenges, there are ample opportunities for mitigation. The first step in addressing this issue is raising awareness of the impacts of a language default.
By acknowledging the importance of diacritics and non-Latin scripts and advocating for their inclusion in digital preservation efforts, we can work towards developing more inclusive and equitable digital platforms and tools.
Additionally, by leveraging technologies we can help facilitate the accurate representation and preservation of all language in preserved digital content.

Points of Failure

In the preservation of diverse language characters in digital content, there are various points of failure where errors or deficiencies can occur.
Identifying these points of failure is important for understanding the challenges associated with handling diacritics and non-Latin scripts in digital preservation efforts.
Here are some key areas where failures may occur:

01 **Web Browsers**

02 **Operating Systems**

03 **Software**

04 **Source Code**

Web browsers serve as gateways to digital content, but they can sometimes fail to accurately render or display text containing diacritics or non-Latin scripts.
This can be due to:

- Limitations in font support
- Rendering algorithms

As a result, users may encounter garbled text, missing characters, or rendering errors.

Operating systems play a crucial role in managing and displaying digital content on various devices.
However, there are inconsistencies across operating systems in terms of encoding standards and forms.
As a result, users may encounter errors when transferring content from one environment to another.
Additionally, language settings and localisation options within operating systems may not fully accommodate the linguistic preferences and requirements of diverse language communities.

Software used for content creation, processing, and dissemination may encounter challenges in handling diverse language characters due to limitations in font support and inconsistencies across encoding standards, encoding forms and normalisation forms.

On the back end of all the above is the **source code**. The code behind your operating systems, software, and browsers may incorporate varying encoding standards, encoding forms, and normalization forms.
Additionally, different programming languages use different internal string representations and report length according to varying units such as ints, shorts, bytes.

Unicode

Unicode stands as the universal character encoding standard for written characters and text. It provides a consistent framework for encoding multilingual text, enabling the exchange of text data internationally and facilitating global interoperability in digital environments.
By adopting Unicode, the information technology industry has moved away from disparate character sets towards data stability and compatibility across languages and scripts.

Code Points: Unicode assigns a unique numeric value, known as a code point, to each character in its repertoire. This allows for precise identification and representation of characters from diverse writing systems.

Character Repertoire: The Unicode Standard encompasses over 1 million characters, covering the written languages of the world. This extensive repertoire includes alphabetic characters, ideographic characters, symbols, and diacritical marks, ensuring comprehensive support for linguistic diversity.

Character Encoding Forms:
Unicode supports the following encoding forms:

- **UTF-8**
- **UTF-16**
- **UTF-32**

They define how characters are represented in binary format. These encoding forms offer flexibility and efficiency in storing and transmitting text data across different digital platforms and systems.

UTF-8 Explained

- One of the encoding forms supported by Unicode.
- It is widely used for its compatibility with ASCII and its efficient use of storage space.
- UTF-8 encodes characters using one to four bytes, depending on the character's code point value.
- This encoding scheme accommodates the full range of characters in the Unicode repertoire, from Arabic to Chinese, Latin to Korean, etc.

A Tale of Two Code Points

Multibyte Characters and Extended Grapheme Clusters

In the context of Unicode encoding, characters (or graphemes) can sometimes consist of more than one code point, leading to the concept of extended grapheme clusters. An understanding of extended grapheme clusters aids the accurate representation and manipulation of text data in digital environments.

An extended grapheme cluster is a sequence of one or more Unicode code points that must be treated as a single, unbreakable character. Unlike individual code points, which may not always correspond to a single character in the user's perception, extended grapheme clusters represent minimally distinctive units of writing in a particular writing system.
For example, in Unicode, characters like "ö" in German or "시" in Korean may be composed of multiple code points.

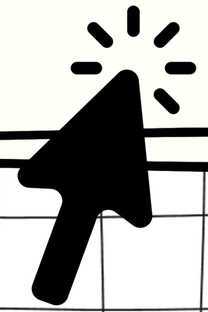
The encoding of multibyte characters and extended grapheme clusters presents unique challenges, particularly in relation to text manipulation operations such as selection, copying, editing, or deletion. Failure to respect multibyte characters and extended grapheme clusters can result in data corruption and rendering issues.

In some programming languages and frameworks, the default behaviour is to treat strings as sequences of bytes, where each byte represents a single character.
This approach works well with the use of single-byte character encodings like ASCII, however, with Unicode, and the need to handle multibyte characters and extended grapheme clusters, this approach can lead to problems.
For example, when iterating through a string using a byte-based approach, it might incorrectly split multibyte characters into separate entities, causing data corruption.
This extends to grapheme clusters, which should also be treated as one indivisible unit.
To detect extended grapheme clusters and determine string length accurately,
the use of a Unicode library is advisable.

Normalisation

To address the variability in character encoding and representation, Unicode offers normalisation forms, including **NFD**, **NFC**, **NFKD**, and **NFKC**. These normalisation forms aim to standardise the representation of characters by either decomposing them into their smallest possible pieces (**NFD**), combining them into pre-composed forms (**NFC**), or replacing visual variants with default ones (**NFKD** and **NFKC**).

Normalisation is essential before comparing strings or searching for substrings to ensure consistent and accurate text processing.



By leveraging the Unicode standard and adhering to normalisation practices, we can mitigate the issues associated with variable-length character encoding and promote accurate representation of text data in digital preservation efforts.

Strategies

- Advocating for the adoption of the Unicode standard to ensure consistent encoding and representation of diverse language characters.
- Collaborating with technology developers to improve support for diacritics and non-Latin scripts in digital platforms and tools.
- Providing training and resources to digital archivists and librarians, and information professionals on best practices for handling diverse language characters in digital content.
- Implementing quality assurance measures and testing protocols to identify and address rendering issues or compatibility issues across digital environments.

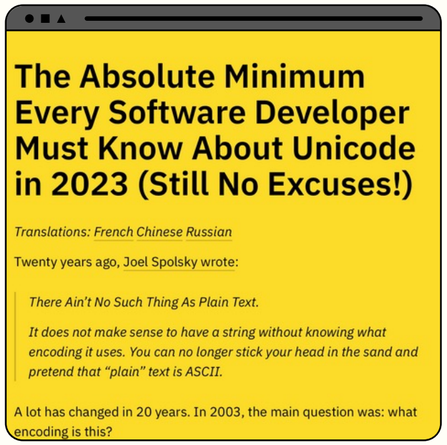
Conclusion

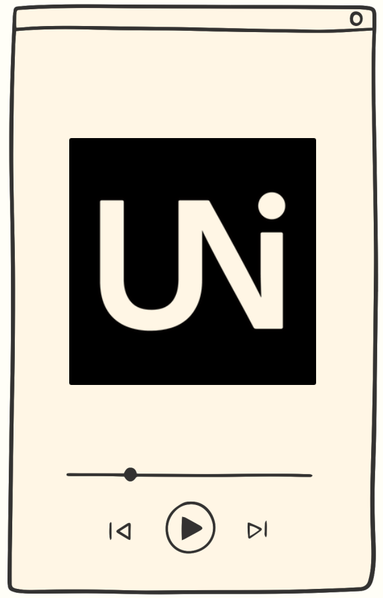
The preservation of diverse language characters in digital content is not solely a technical issue, but a matter of cultural significance and social equity.

By leveraging the Unicode standard and understanding the complexities of character encoding and normalisation forms, we can overcome barriers to language representation and ensure that all languages and scripts are accurately represented and preserved in digital content.

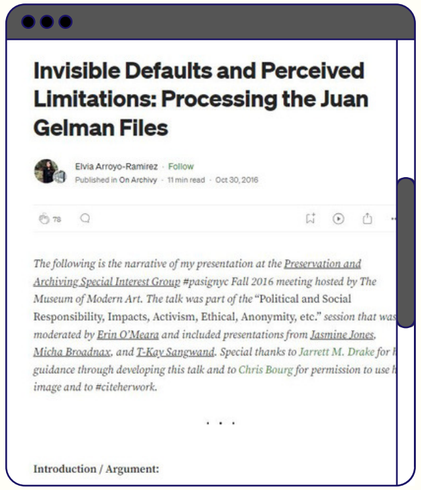
With that, we promote linguistic diversity and cultural inclusivity in digital preservation efforts, but also further afield.

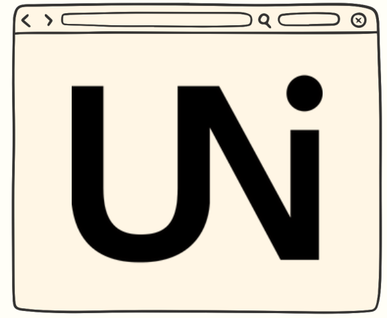
Resources

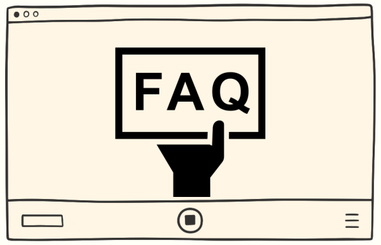

The Absolute Minimum Every Software Developer Must Know About Unicode in 2023 (Still No Excuses!)
Translations: French Chinese Russian
Twenty years ago, Joel Spolsky wrote:
There Ain't No Such Thing As Plain Text.
It does not make sense to have a string without knowing what encoding it uses. You can no longer stick your head in the sand and pretend that "plain" text is ASCII.
A lot has changed in 20 years. In 2003, the main question was: what encoding is this?



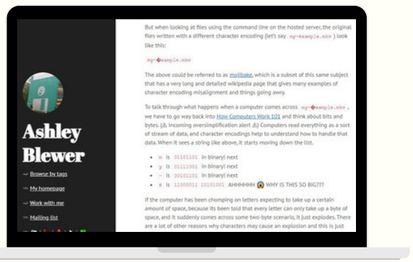

17:45
I'm Joel Spolsky, a software developer in New York City. More about me.
OCTOBER 8, 2003 by JOEL SPOLSKY
The Absolute Minimum Every Software Developer Absolutely, Positively Must Know About Unicode and Character Sets (No Excuses!)
joelonsoftware.com

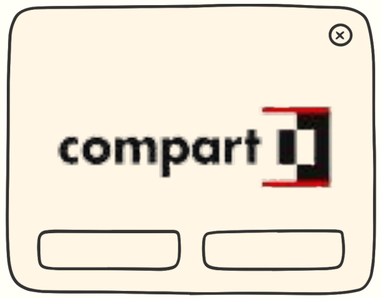

Invisible Defaults and Perceived Limitations: Processing the Juan Gelman Files
Elvia Arroyo-Ramirez · Follow
Published in On Archivy · 11 min read · Oct 30, 2016
The following is the narrative of my presentation at the Preservation and Archiving Special Interest Group #pasignyc Fall 2016 meeting hosted by The Museum of Modern Art. The talk was part of the "Political and Social Responsibility, Impacts, Activism, Ethical, Anonymity, etc." session that was moderated by Erin O'Meara and included presentations from Jasmine Jones, Micha Broadnax, and T-Kay Sangwand. Special thanks to Jarrett M. Drake for
guidance through developing this talk and to Chris Bourg for permission to use
image and to #citeherwork.
Introduction / Argument:



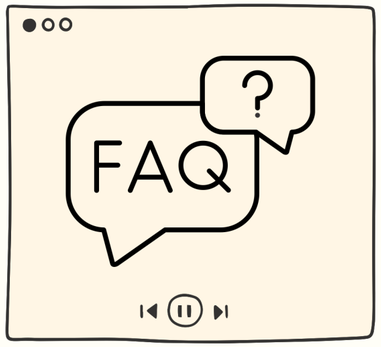

FAQ


cryptii

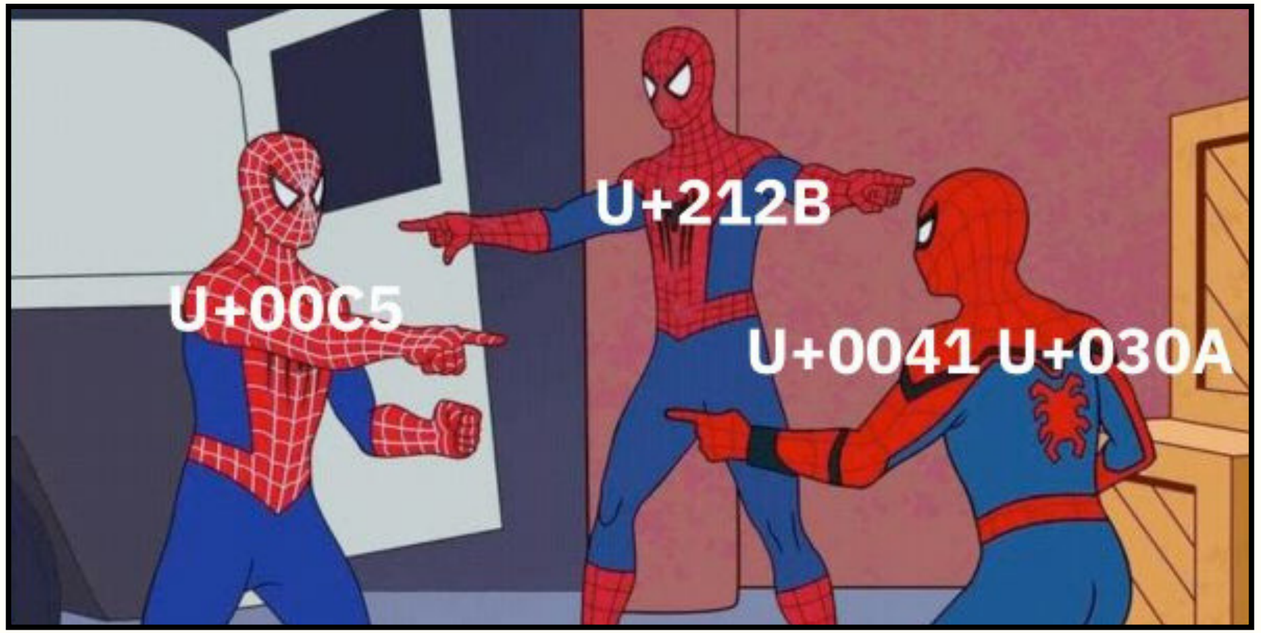

Ashley Blewer
Browse by tags
My homepage
Work with me
Mailing list


compart


Elvia Arroyo-Ramirez Follow
Published in On Archivy 11 min read


FAQ


Noto: A typeface for the world
人類社会のすべての構成員の固有の尊厳と平等で譲る


U+00C5
U+212B
U+0041 U+030A