Criteria for evaluating programming languages and frameworks with regard to their UA compliance

(Process: A criteria match counts for 1. A criteria mismatch counts for 0. After judging all criteria questions, add up the results.)

Glossary and abbreviations

PL - programming language (please don't confuse with PL/I)
device - an explicitly or implicitly called function, method, subroutine or predicate (depending on the paradigm) used to perform operations on a data set, such as an domain name in string form
data set - a domain name or email address
native - native in this context pertains to the distance of the device from the core of the language
predefined - anything that is for the most part invariable, though maybe not immutable. Devices compiled into the interpreter (for script languages) are predefined, though they may be overridden.
defined - anything variable, such as a device provided by a piece of code as part of a library to the language

Initial observations (for information only)

- What era is the programming language from?
- Has network enabled communication been a focus of the programming language?
- What are the primary domains of the programming language, derived from its real-world use?

Does the PL offer native devices to process a domain name or email address?

- A device is an explicitly or implicitly called function, method, subroutine or predicate (depending on the paradigm) used to perform operations on a data set, such as an domain name in string form.
- Implementations of such devices within the translation units of the interpreter (if present) are native by default.
- Implementations of such devices in the programming language itself are to be considered native unless they're not bundled with the programming languages as a run time environment.
- Hint: If a regular download of the programming language project's distribution files comes with a "standard library" containing such devices for use during run time, this criteria matches.

Does the PL offer devices to inspect a domain name or email address in a structured manner?

- Is the user able to get structured bits of information from a data set?
- Domain name or email address segmentation, such as splitting host and local part.
- Perhaps identifying a TLD, or retrieving the second level domain name.
- Hint: This is not about validating the data set, but about accessing parts of some meaning.

Do the PLs devices offer the ability to assess the validity of a data set in a structured manner?

- Can the user receive detailed information about the validation result?
- If the devices return structured information such as 'error in local part exists', or 'address not RFC compliant', this criteria matches.
- The PL reporting errors in the encoding of the data set DOES NOT match this criteria. Encoding errors are unstructured, hence they are handled elsewhere.

Do the devices handling domain names and email addresses accept input in Unicode?
 - For this criteria, only the common UTF variants count (UTF-8, UTF16, UTF-32).
 - Hint: Accepting either one of the UTFs will result in a criteria match.

Do the devices handling domain names and email addresses accept input in Punycode?
 - Punycode is detailed in RFC3492
 - Converts Unicode to ASCII for transport.

Do the devices handling Punycode identify incompatible Unicode sequences?
 - Punycode is not concerned with 'compliant' Unicode encodings.
 - Punycode to Unicode conversion must thus be integrated into the devices.
 - Rejection or dismissial of incompatible Unicode sequences make this a match.
- Anything else makes this a criteria mismatch.

What is the internal storage format for domain names and email addresses when used in string form?
 - This is for information only; please describe the internal storage format with the name of the closest common encoding.
 - The storage format is most likely a UTF type store.
 - Hint: Please do not count this as a criteria

Is TLD identification supported within the programming language? *(parent to two)*
 - TLD identification is the distinct and exact separation and lookup of the most significant part of the domain name.
 - Splitting a string on a dot boundary is not identification, but segmentation. If a lookup is missing, this is a mismatch.
 - Code splitting a string on a dot boundary and performing a lookup search on a table of some kind makes this a match.
Dependent criteria if parent matches: **Is TLD identification performed using a predefined list or a defined list?**
 - Predefined lists could be compiled in, or otherwise constant. TLDs are on the move, so predefined lists make this a mismatch.
 - Defined lists are able to be updated, perhaps during time of operation. Defined lists make this a match.
Dependent criteria if parent matches: **Are defined lists created from official feeds?**
 - Secondary providers - those without integration into the ICANN processes - make this a mismatch.
- Consequently, primary providers (ICANN/IANA)make this a match.

Does the programming language or framework store domain names and email addresses in an endorsed Unicode representation format in memory during run time?
- Must be either UTF-8/16/32

Does the programming language or framework store domain names and email addresses in an endorsed Unicode representation format in persistent storage?
- Must be either UTF-8/16/32

Does the programming language implement handling of domain names and email addresses according to specifications and best practice?
- Segments have maximum lengths, but Punycode may exceed them.
- TLDs can have more then three letters, and be internationalized

When processing domain names or email addresses for display, is the string representation of the data converted from an internal format to the display format?

Does the programming language or framework account for faulty Unicode encodings stemming from a Punycode conversion device?