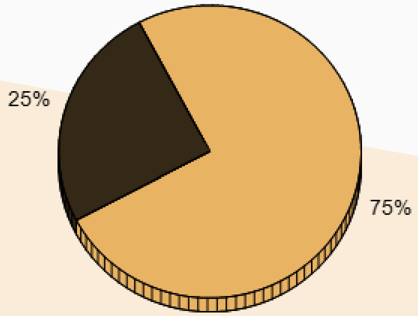
Universal Acceptance – Technical Perspective

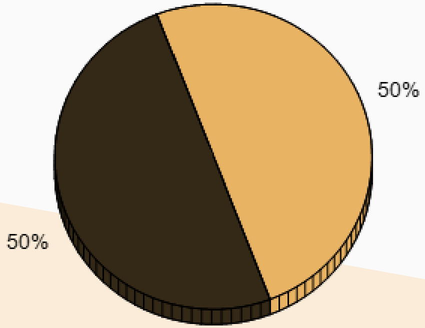
Universal Acceptance

Warm-up Exercise

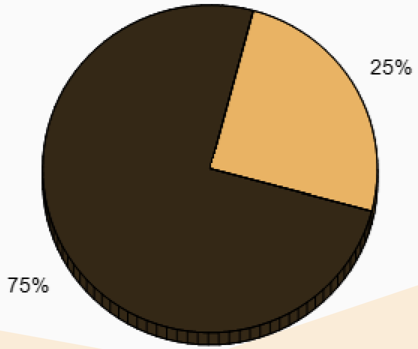
According to w3techs, which of the following pie charts most closely represents the fraction of websites on the Internet that are primarily English language based content?



25% English
75% All other



50% English
50% All other

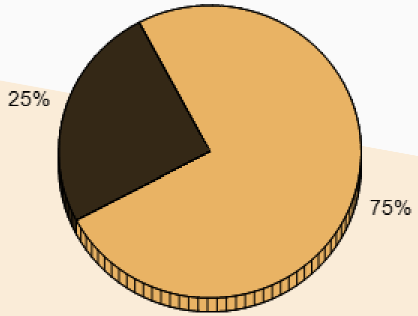


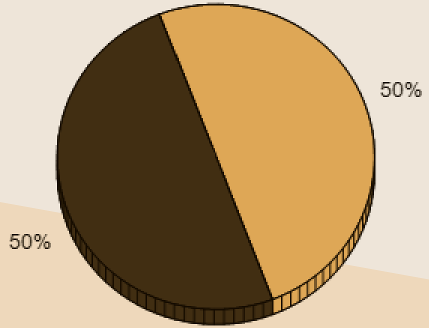
75% English
25% All other

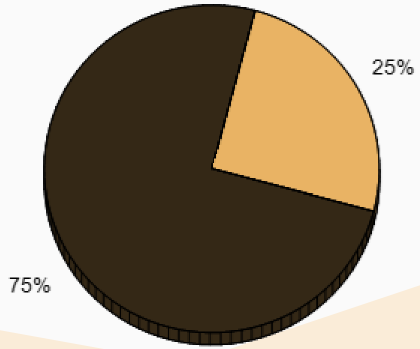
Actual data can be found at https://w3techs.com/technologies/history_overview/content_language/ms/y. Data shown is approximate.

Warm-up Exercise: Solution

According to w3techs, which of the following pie charts most closely represents the fraction of websites on the Internet that are primarily English language based content?







25% English
75% All other

50% English
50% All other

75% English
25% All other

Actual data can be found at https://w3techs.com/technologies/history_overview/content_language/ms/y. Data shown is approximate.

Prepare to do business and communicate with a global user base

Continued expansion of the Internet is allowing access to an increasingly diverse user group

- There is a growing number of languages and scripts present on the Internet, including non-Latin based, language-specific domain names in Arabic, Chinese and many other scripts.
- It is <u>required</u> that your Internet-enabled applications, devices and systems accept, validate, store, process and display all domain names and email address appropriately.

Goals for Today's Lecture

1 Understand and be able to describe the 5 criteria of Universal Acceptance

2 Understand and be able to implement good practices related to each of the 5 criteria

Five Criteria of UA

Accept Validate Store Process Display

Universal Acceptance is the state where

* all valid domain names and email addresses are **accepted, validated, stored, processed** and **displayed** correctly and consistently
* by all Internet-enabled applications, devices and systems.

Accept

Accepting occurs whenever an email address or a domain name is received as a string of characters from a user interface, from a file, or from an API used by a software application or online service.

Key points:

Applications and services allow domain names and email addresses to be:

* Entered into user interfaces, AND/OR
* Received from other applications and services via APIs.

Validate

Validation may occur in many places whenever an email address or a domain name is either received or emitted as a string of characters by an application or online service.

Key Points

* Intended to ensure that the entered information is either valid or at least definitely not invalid
* For domain names and email addresses, many programmers have been using heuristics (for example, checking that a TLD has the "correct" number of characters, or that the characters are from the ASCII character set). These heuristics are no longer applicable because the Internet is changing:
    * Domain names and email addresses can now include Unicode characters (and punycode strings)
    * The list of TLDs is growing, changing, and includes left-to-right scripts
    * A TLD can be up to 63 characters long

Store

The Storage process occurs whenever an email address or a domain name is stored as a string of characters in a database or file used by a software application or online service.

Key Points

* Applications and services might require long-term and/or transient storage of domain names and email addresses. Regardless of the lifetime of the data, it must be stored in:
    * RFC-defined formats, OR
    * Alternate formats that can be easily translated to and from RFC-defined formats (this is much less desirable)

Although RFCs require the use of UTF-8 (see RFC 5890), other formats may be encountered in legacy code. See the “Good Practices” section.

Process

Processing occurs whenever an email address or a domain name is used by an application or service to perform an activity (for example, searching or sorting a list), or transformed into an alternate format (such as storing ASCII as Unicode).

Key Points

* Processing means using domain names and email strings in a feature. Additional validation may occur during processing. There is no limit to the number of ways that domain names and email addresses could be processed.
* Examples:
  * "Identify all the people associated with New Zealand because they have a name with a .nz ccTLD";
  * "Identify all the pharmacists because they have a user@example.pharmacy email address";
  * "Identify firewalls that might filter DNS requests that don't apply to their policies".

Display

The display process occurs whenever an email address or a domain name is rendered within a user interface.

Key Point

* Displaying domain names and email addresses is usually straightforward when the scripts used are supported in the underlying operating system and when the strings are stored in Unicode. If these conditions are not met, application-specific transformations may be required. Organizations may also have their own display criteria.

Transition to Good Practices

Accept

Validate

Store

Process

Display

Accept : Good Practices

The process by which an email address or domain name is received as a string of characters from a user interface, file or API.

Good Practices

* User interface elements must support:
  * Unicode
  * Strings up to 256 characters
* ASCII Compatible Encoded text ("Punycoded") in place of Unicode
  * Unicode shown by default
  * Punycoded text shown *only* when it provides a benefit
  * See RFC 3492

Validate: Good Practices

The process by which an email address or domain name – received or emitted – is checked for syntax correctness.

Good Practices

* Validate only if it is required for the operation of the application or service
    * This is the most reliable way to ensure that all valid domain names are accepted into your systems.
* Validate only to the minimum extent necessary
* Recognize that syntactically correct inputs may not represent domain names or email addresses currently in use on the Internet

Validate: Good Practices

The process by which an email address or domain name – received or emitted – is checked for syntax correctness.

Good Practices

If you must validate, consider the following:

* Verify TLD against authoritative table
  * Examples of some authoritative tables that you can use are:
  * http://www.internic.net/domain/root.zone
  * http://data.iana.org/TLD/tlds-alpha-by-domain.txt
  * See also: https://www.icann.org/en/system/files/files/sac-070-en.pdf
* Query domain name against DNS
  * **For example, consider using the GETDNS API (http://getdnsapi.net/)**
* Require repeated entry of email address to preclude typing errors

Validate: Good Practices

The process by which an email address or domain name – received or emitted – is checked for syntax correctness.

Good Practices

If you must validate, consider the following (continued):

* Validate characters - no “disallowed” code points
    * See RFC 5892
* Limit to a few, whole-label rules defined in RFCs
    * See RFC 5894
* If a string resembling a domain name contains the Arabic full stop character “۔” (U+06D4), or the ideographic full stop character “。” (U+3002), convert it to the full stop “.” (U+002E)
* Ensure that the product or feature handles numbers correctly
    * For example: ASCII numerals and Asian ideographic number representations should all be treated as numbers

Store: Good Practices

The long-term and / or transient storage of domain names and email addresses.

Good Practices

* Apps / services should support Unicode
* Information stored in UTF-8 whenever possible
  * Some systems may require support for UTF-16 as well, but generally UTF-8 is preferred. UTF-7 and UTF-32 should be avoided.
  * In UTF-8, UTF-16, and UTF-32, each character is represented either as a sequence of one to four 8-bit bytes, one or two 16-bit code units, or a single 32-bit code unit. UTF-7 was intended to provide a means of encoding Unicode text for use in Internet E-mail messages (see RFC 2152).
* Consider end-to-end scenarios before converting between A-Labels & U-Labels
  * It may be desirable to maintain only U-Labels in a file or database, because it simplifies searching and sorting. However, conversion may have implications when interoperating with older, non-Unicode-enabled applications and services. Consider storing and indexing both formats.

Store: Good Practices

The long-term and / or transient storage of domain names and email addresses.

Good Practices

* Clearly mark email addresses and domain names during storage
    * Instances where email addresses and domain names have been filed under the “author” field of a document or “contact info” in a log file have led to the loss of the original address
* If you don’t store in Unicode, you must be able to match strings in multiple formats
    * For example, a search for example.みんな should also find example.xn--q9jyb4c

Process: Good Practices

Occurs whenever an email address or domain name is used by an application or service to perform an activity, or is transformed into an alternate format.

Good Practices

- Since the Unicode standard is continually expanding, check code points not defined when application / service was created – shouldn’t “break” user experience.
  - Missing fonts in the underlying operating system may result in non-displayable characters (frequently the “▯”character is used to represent these), but this situation should not result in a fatal crash
- Use supported Unicode-enabled APIs
- Use latest Internationalized Domain Names in Applications (IDNA) Protocol & Tables documents for Internationalized Domain Names (IDNs).
  - RFC 5891
  - RFC 5892

Process: Good Practices

Occurs whenever an email address or domain name is used by an application or service to perform an activity, or is transformed into an alternate format.

Good Practices

- Process in UTF-8 wherever possible
- Ensure numbers are handled as expected
- Treat ASCII numerals, Asian, Arabic, and Persian ideographic number representations as numbers
- Upgrade apps & servers/services together
  - If the server is Unicode and client is non-Unicode, or vice versa, the data will need to be converted to each code page every time the data travels between server and client.
- Perform code reviews to avoid buffer overflow attacks
  - When doing character transformation, text strings may grow or shrink substantially.

Display: Good Practices

Display occurs whenever an email address or a domain name is rendered within a user interface.

Good Practices

* Display all Unicode code points supported by underlying operating system.
    * If an application maintains its own font sets, comprehensive Unicode support should be offered to the collection of fonts available from the operating system.
* When developing app/service, or operating a registry, consider languages supported and make sure operating systems and applications cover those languages.
* Convert non-Unicode data to Unicode before display.
    * End user should see "everyone.みんな" vs. "everyone.xn--q9jyb4c." (This conversion is an example of UA-ready processing.)

Display: Good Practices

Display occurs whenever an email address or a domain name is rendered within a user interface.

Good Practices

* Use Unicode IDNA Compatibility Processing to match user expectations
    * To learn more, go to: http://unicode.org/reports/tr46
* Be aware of unassigned & disallowed characters
    * See RFC 5892

Summary

Accept Validate Store Process Display

It is required that your Internet-enabled applications, devices and systems accept, validate, store, process and display all domain names and email address appropriately.

Tools & Resources for Developers

Authoritative Tables:

* http://www.internic.net/domain/root.zone
* http://www.dns.icann.org/services/authoritative-dns/index.html
* http://data.iana.org/TLD/tlds-alpha-by-domain.txt
* See also SAC070: https://tinyurl.com/sac070

Internationalized Domain Names for Applications:

* Tables: https://tools.ietf.org/html/rfc5892
* Rationale: https://tools.ietf.org/html/rfc5894
* Protocol: https://tools.ietf.org/html/rfc5891

Unicode:

* Security Considerations: http://unicode.org/reports/tr36/
* IDNA Compatibility Processing: http://unicode.org/reports/tr46/

Universal Acceptance Steering Group info & recent developments: www.uasg.tech

Glossary, partial

A-label - The ASCII-compatible encoded (ACE) representation of an internationalized domain name, e.g. how it is transmitted internally within the DNS protocol. A-labels always commence with the prefix “xn--”.

ACE prefix -ASCII Compatible Encoding Prefix.

ASCII Characters - American Standard Code for Information Interchange. These are characters from the basic Latin alphabet together with the European-Arabic digits. These are also included in the broader range of "Unicode characters" that provides the basis for IDNs.

API - An Application Programming Interface (API) is a set of routines, protocols, and tools for building software and applications. An API may be for a web based system, operating system, or database system, and it provides facilities to develop applications for that system using a given programming language.

Codespace - Range that define the lower and upper bounds for an encoding.

Code Points - A code point or code position is any of the numerical values that make up the code space. They are used to distinguish both, the number from an encoding as a sequence of bits, and the abstract character from a particular graphical representation (glyph).

DNS Root Zone - The root zone is the central directory for the DNS, which is a key component in translating readable host names into numeric IP addresses.

Glossary, partial

EAI - Email Address Internationalization is an email address that requires the use of Unicode in all parts of the email address.

IANA - Internet Assigned Numbers Authority. Its functions include: (1) Maintenance of the registry of technical Internet protocol parameters, (2) Administration of certain responsibilities associated with Internet DNS root zone, and (3) Allocation of Internet numbering resources.

ICANN - The Internet Corporation for Assigned Names and Numbers (ICANN) is an internationally organized, non-profit corporation that has responsibility for Internet Protocol (IP) address space allocation, protocol identifier assignment, generic (gTLD) and country code (ccTLD) Top-Level Domain name system management, and root server system management functions.

IDN - Internationalized Domain Names. IDNs are domain names that include characters used in the local representation of languages that are not written with the twenty-six letters of the basic Latin alphabet “a-z”, the numbers 0-9, and the hyphen “-“.

IDNA - Internationalized Domain Names in Applications.

IDN ccTLD - Country Code Top-level Domain that includes characters used in the local representation of languages that are nor written with the twenty-six letters of the basic Latin alphabet “a-z”. Examples: .рф (Russia), . مصر Egypt, and . السعودية Saudi Arabia.

Glossary, partial

IETF - The Internet Engineering Task Force (IETF) is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. It is open to any interested individual. The IETF develops Internet Standards and in particular the standards related to the Internet Protocol Suite (TCP/IP).

Language - The method of human communication, either spoken or written, consisting of the use of words in a structured and conventional way.

Punycode - It is an algorithm to represent Unicode with the limited character subset of ASCII supported by the Domain Name System. Punycode is intended for the encoding of labels in the Internationalized Domain Names in Applications (IDNA) framework.

Registrar - An organization where domain names are registered by users. The registrar keeps records of the contact information and submits the technical information to a central directory known as the "registry".

Registry - The authoritative, master database of all domain names registered in each Top Level Domain.

RFC - A Request for Comments (RFC) is a formal document from the Internet Engineering Task Force (IETF) that is the result of committee drafting and subsequent review by interested parties.

Glossary, partial

Script - The collection of letters or characters used in writing, representing the sounds of a language.

Second-level domain name - In the Domain Name System (DNS) hierarchy, a second-level domain (SLD or 2LD) is a domain that is directly below a top-level domain (TLD). For example, in example.com, example is the second-level domain of the .com TLD.

U-label - A "U-label" is an IDNA-valid string of Unicode characters including at least one non-ASCII character. Conversions between U-labels and A-labels are performed according to the Punycode specification [RFC 3492].

UA-ready Software or UA-Readiness - Universal Acceptance Ready Software. It is a software that has the ability to Accept, Store, Process, Validate and Display all Top Level Domains equally and all IDNs, hyperlink and email addresses equally.

Unicode - A universal character encoding standard. It defines the way individual characters are represented in text files, web pages, and other types of documents. Unicode was designed to support characters from all languages around the world.

Some Relevant RFCs to Universal Acceptance

Internationalized Domain Name Acceptance	
Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework	https://tools.ietf.org/html/rfc5890
Internationalized Domain Names in Applications (IDNA): Protocol	https://tools.ietf.org/html/rfc5891
The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)	https://tools.ietf.org/html/rfc5892
Internationalized Domain Names for Applications (IDNA): Background, Explanation, and Rationale	https://tools.ietf.org/html/rfc5894

Additional RFCs to Referenced

Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)	https://tools.ietf.org/html/rfc3492
UTF-7: A Mail-Safe Transformation Format of Unicode	https://tools.ietf.org/html/rfc2152

Additional RFC relevant to Universal Acceptance:
https://uasg.tech/wp-content/uploads/2017/06/UA006-Relevant-RFCs.pdf