

Basics of the operation of the Web

- **Resource:** anything that might be identified by a URI. Information resource: a resource which has the property that all its essential characteristics can be conveyed in a message.
- **Uniform Resource Identifier (URI):** a global identifier in the context of the Web.
- **Representation:** data that encodes information about resource state.
- **Content negotiation:** offering multiple representations for a resource and selecting the one that is the most appropriate when a representation must be served.
- **Dereferencing a URI:** using a URI to access the referenced resource. Access may take many forms, including retrieving, adding, or modifying a representation of the resource, and deleting some or all representations of the resource.
- **Web agent:** a person or a piece of software acting on the Web on behalf of a person, entity, or process.
- **User agent:** one type of Web agent, a piece of software acting on behalf of a person. E.g.: a web browser.

Web standards and standards organizations

A standard is a document that provides requirements, specifications, guidelines or characteristics that can be used consistently to ensure that materials, products, processes and services are fit for their purpose.

There are three types of standards:

- **De facto standards:** arise from common usage or market acceptance. Examples: the QWERTY keyboard layout, TeX, PDF (before 2008).
- **De jure standards:** are mandated by regulators at the local, state, federal, and/or international level. Examples: International System of Units (SI), PDF (from 2008).
- **Voluntary consensus standards:** are specified within a range of private institutions, including engineering societies, trade associations, accredited standards-setting organizations, and industry consortia. Examples: The Internet protocol suite (commonly known as TCP/IP), HTML5.

Organizations responsible for web standards: IANA

- Internet Assigned Numbers Authority (IANA's full form)
- Coordinates the allocation of codes and numbers that form the basis for the operation of the Internet.
 - Manages the DNS root zone, and the .int and .arpa domains.
 - Coordinates the allocation of IP addresses globally.
 - Maintains registries of codes and numbers used in a variety of Internet protocols.
- IANA is a function that is currently performed by the Internet Corporation for Assigned Names and Numbers (ICANN), a not-for-profit corporation.

Organizations responsible for web standards: IETF

- Internet Engineering Tasking Force (IETF's full form)
- An international standards organization developing Internet standards.
 - For example, IETF develops the Internet protocol suite (commonly known as TCP/IP).
 - The IETF has no formal membership, no membership fee, participation is open to anyone.
 - The technical work is done in working groups.
- Formation: January 16, 1986
- Publishes Internet standards-related specifications in the RFC series of documents.

Organizations responsible for web standards: the RFC Series

- Request for Comments (RFC's full form)
- The RFC series contains technical and organizational documents about the Internet.
- By origin, the RFC series is split into four streams:
 - The Internet Engineering Task Force (IETF) Stream
 - The Internet Architecture Board (IAB) Stream
 - The Internet Research Task Force (IRTF) Stream
 - The Independent Submission Stream
- Each RFC is identified by a number, such as RFC 7230.
- Each RFC is available in ASCII text.
- Published RFCs never change.
- Various errors are fixed by errata.
- Amendments can be also made by writing and publishing a revised RFC. An RFC can obsolete or update earlier RFCs.
- The series of IETF RFCs contains the following two important sub-series: Best Current Practice (BCP) and Internet Standard (STD)
- BCPs and STDs are assigned a number in their subseries while retaining their RFC number.
- Several RFCs may share the same BCP or STD number. For example, an STD number identifies a standard not a document.
- Standards Track: the set of maturity levels of RFCs that are intended to become Internet Standards. Originally, three maturity levels were used:
 - Proposed Standard
 - Draft Standard
 - Internet Standard
- Currently, the Proposed Standard and Internet Standard maturity levels are used.
- Internet-Draft: a draft version of a specification made available for informal review and comment during the development.
 - May or may not eventually be published as an RFC.
 - Is subject to change or removal at any time.
 - Is valid for a maximum of six months.

- Should not be cited or quoted in any formal document, except as “work in progress”.

Organizations responsible for web standards: W3C

- The W3C is an international community where member organizations, a full-time staff, and the public work together to develop Web standards.
- W3C publishes documents called Recommendations that define Web technologies and are considered Web standards. W3C standards are developed according to the Open Standards Principles.
- **W3C Design Principles**
 - **Web for All:** The Web must be available to all people, whatever their hardware, software, network infrastructure, native language, culture, geographical location, or physical or mental ability
 - **Web on Everything:** The Web must be accessible from a wide variety of devices. E.g., mobile phones, smart phones, PDA devices, domestic appliances.

URI and its Syntax

Uniform Resource Identifier (URI):

- A compact sequence of characters that identifies an abstract or physical resource.
 - A resource is not necessarily available on the Web.
 - URIs can be assigned even to objects from the real world or to concepts.
- Each URI begins with a scheme name that is separated by a ':' character from the scheme-specific part of the URI. Scheme specifications can define their scheme-specific syntax within certain limits.
- The organization responsible
- Three types of Schemes: file, http/https and mailto.

URI Syntax

- Syntax is organized hierarchically. Components listed in order of decreasing significance from left to right.
- Generic syntax: scheme ':' hier-part ['?' query] ['#' fragment]
 - The hier-part component may consist of an authority and a path component, its syntax is: '//' authority path or path
 - When authority is present, the path must either be empty or begin with a '/' character.
 - When authority is not present, the path cannot begin with two '/' characters.

- Example:
<https://wordery.com/search?term=scotland#header>
- Example:
<mailto:jeszenszky.peter@inf.unideb.hu?subject=XML>

The Host, the Port

- The name comes from that the name space defined by the remainder of the URI is under its jurisdiction.
- Syntax: [userinfo '@'] host [':' port]
 - A URI scheme may define a default port.
 - For example, the http scheme defines a default port of 80.

Path

- A sequence of path segments separated by a '/' character.
- Terminated by the first '?' or '#', or by the end of the URI.
- The path segments '.' and '..' can be used just as in some operating systems' file directory structures.

Query

- Indicated by the first '?' character and terminated by a '#' character or by the end of the URI.
- Contains non-hierarchical data.
- Often contains name/value pairs of the form name '=' value delimited by an '&' character. In the case of the http and https URI schemes the query component is used for submitting form data

Fragment identifier components

- Indicated by a '#' character and terminated by the end of the URI.
- Allows indirect identification of a secondary resource by reference to a primary resource and additional identifying information.
- The semantics of a fragment identifier are defined by the set of representations that might result from a retrieval action on the primary resource.
- The fragment identifier is separated from the rest of the URI prior to a dereference.
- URI scheme specifications must define their own syntax so that all strings matching their scheme-specific syntax must be an absolute URI without a fragment identifier.

HTTP - Hypertext Transfer Protocol

- A request/response protocol based on the client-server model.

- **Stateless** i.e., subsequent requests are treated independent of each other.
- **Extensible**. E.g., methods, status codes, header fields.
- **General-purpose** - Although mainly used for communication between clients and web servers, in principle, can be used for any other purpose.

The structure of requests and responses

Requests:

- The start-line of requests has the following syntax: *method request-target HTTP-version* CRLF. Components within the line must be separated by a single space.
- The *request-target* identifies the target resource upon which to apply the request.
- The most common form of *request-target* is the following: **path ['?' query]**
 - If the target URI's path component is empty, the client must send '/' as the path.
 - The host and port components of the target URI are sent in the Host header field.
- The request-target can be '*' only for a server-wide OPTIONS request.

Responses:

- The first line of a response message is called the status-line, it has the following syntax: *HTTP-version status_code reason_phrase* CRLF. Components within the line must be separated by a single space.
- The status code consists of three decimal digits and is followed by a short textual description associated with the numeric status code.
- Examples:
 - /1.1 200 OK
 - HTTP/1.1 404 Not Found

HTTP Methods

- Method names are case-sensitive.
- All general-purpose servers must support the methods GET and HEAD, all other methods are optional.
- **GET** – Request's transfer of a current selected representation for the target resource. The response to a GET request is cacheable.
- **HEAD** – Identical to the GET method except that the server must not send a message body in the response. Can be used for obtaining metadata about the selected representation without transferring the representation data. The response to a HEAD request is cacheable.
- **POST** – Requests that the target resource process the representation enclosed in the request according to the resource's own specific semantics. Possible applications include:
 - Submitting data (e.g., form data) to a data-handling process.
 - Posting a message to a newsgroup, mailing list, or blog.

- Creating a new resource.
- Appending data to a resource's existing representation(s).
- **PUT –**
 - Requests that the state of the target resource be created or replaced with the state defined by the representation enclosed in the request message payload.
 - A successful PUT of a given representation would suggest that a subsequent GET on that same target resource will result in an equivalent representation being sent in a 200 (OK) response.
- **DELETE –** Requests that the origin server remove the association between the target resource and its current functionality.

Status Codes

- Three-digit decimal integer numbers.
- The first digit defines the class of status code (response).
- Clients are not required to understand the meaning of all registered status codes.
 - However, a client must understand the class of any status code, as indicated by the first digit.
 - An unrecognized status code must be treated as being equivalent to the x00 status code, where x is the first digit of the unrecognized status code.
- Status codes are extensible.
- IANA maintains the registry of status codes.
- Classes of Status Codes
 - **1xx: Informational** - Indicates an interim response prior to sending a final response.
 - **2xx: Success** - The request was successfully received, understood, and accepted by the server.
 - **3xx: Redirection** - Further action needs to be taken by the user agent in order to fulfil the request, this can happen automatically.
 - **4xx: Client Error**
 - **5xx: Server Error**

Content Negotiation

- Content negotiation is a mechanism for selecting the most appropriate representation to a given request.
- HTTP/1.1 defines the following two patterns of content negotiation:
 - **Proactive:** the server selects the representation based upon the user agent's stated preferences. This is also known as server-driven negotiation.
 - **Reactive:** the server provides a list of representations for the user agent to choose from. This is also known as agent-driven negotiation.
- There are further patterns, different patterns are not mutually exclusive.

Cookies

- A name/value pair and associated metadata (attributes) sent by an origin server to a user agent using the Set-Cookie header field in a response. An origin server can specify a scope for the cookie using the attributes.
- In subsequent requests, the user agent returns the name/value pair to the origin server in the Cookie header field.
- Uses:
 - Session management
 - Personalization
 - Web tracking
- When the user agent receives a Set-Cookie header, it stores the cookie together with its attributes.
- Subsequently, when the user agent makes an HTTP request, it includes the applicable, non-expired cookies in the Cookie header field. It includes only the name/value pairs without the attributes!
- If the user agent receives a new cookie with the same name, Domain attribute, and Path attribute as a cookie that it has already stored, the existing cookie is replaced with the new cookie.
- When the current session is over, must remove all nonpersistent cookies from the cookie store. (Cookie Management)

Markup languages of the Web

Markup languages are computer languages for annotating text. They allow the association of metadata with parts of text in a clearly distinguishable way. Markdown is a lightweight markup language with plain text formatting syntax.

XML Documents

- Textual objects that are well-formed according to the standard.
- Each XML document has both a logical and a physical structure.
 - Physically, it is comprised of storage units called entities, each of which may refer to other entities.
 - Logically, an XML document consists of declarations, elements, comments, character references, and processing instructions, collectively known as the markup.
- Each element is delimited by a start-tag and an end-tag or is made up from a single empty-element tag.
- XML documents should begin with an XML declaration which specifies the version of XML being used.
- A special attribute named xml:lang may be inserted in documents to specify the language used in the contents and attribute values of any element
- Documents Type Declaration allows to define constraints on the logical structure of the document and supports the use of the storage units.
- An XML document can have following elements:
 - Declaration,

- Root,
- Comments,
- DocType,
- Elements which comprise of element name, value, namespace, comment, CDATA, entity references.
- Refer to this link for basic idea - <http://www.brenkoweb.com/tutorials/xml/xml-structure/structure-of-xml-document>

HTML Documents

- Elements, attributes, and attribute values in HTML are defined to have certain meanings (semantics). For example, the ol element represents an ordered list, and the lang attribute represents the language of the content.
- Authors must not use elements, attributes, or attribute values for purposes other than their appropriate intended semantic purpose.
- The majority of presentational features from previous versions of HTML are no longer allowed.
- The problems of presentational markup:
 - The use of presentational elements leads to poorer accessibility.
 - Higher cost of maintenance. – Larger document sizes.
- The only remaining presentational markup features in HTML are the style attribute and the style element.
- The following elements that were previously presentational have been redefined to be media-independent:

Element	Description	
	HTML 4.01, XHTML 1.0	HTML5
b	Bold font	Keywords
i	Italic font	Alternate voice
hr	Horizontal rule	Thematic break
small	Smaller font	Side comment
s	Strike-through	Inaccurate text
u	Underline	Keywords

- Each element defined in this specification has a content model: a description of the element's expected contents.
- In the HTML syntax the document type declaration is required, whose only purpose is to ensure that the document is rendered in standards mode.

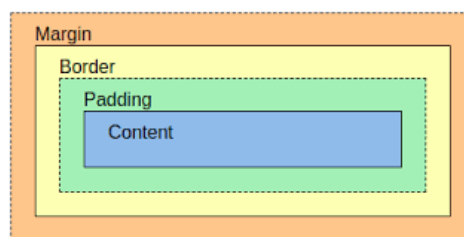
Comparison of XML and HTML

- XML:
 - No predefined set of tags
 - Its purpose is to describe data. Is used as a data exchange format
- HTML:
 - Uses a predefined set of tags
 - Its purpose is to present information. Is a presentation language

- Can be considered as a special application of XML (XHTML)

Stylesheet languages (CSS)

- A style sheet language for describing the rendering of structured documents (such as HTML and XML). Supports rendering on different devices, such as screens, printers, and Braille devices.
- Separates the presentation style of documents from the content of documents. Thus, simplifies web authoring and site maintenance.
- **Box Model:**
 - CSS takes a source document, organized as a tree, and renders it onto a canvas (such as the screen) generating an intermediary structure, the box tree, which represents the formatting structure of the rendered document.
 - Each box in the box tree represents a corresponding element (or pseudo-element) from the document in space and/or time on the canvas.
 - For each element, CSS generates zero or more boxes as specified by that element's display property. Typically, an element generates a single box.



- **Rule Ordering:**
 - The ordering of style rules may be significant. The ordering is significant when more than one style rule with the same specificity applies to an element.
 - In this case, the last declaration in document order wins.
 - For example, the first two rules will never be taken into account here:

```

- a:active { color: red }    /* (a = 0, b = 1, c = 1) */
a:hover { color: green }    /* (a = 0, b = 1, c = 1) */
a:visited { color: black }  /* (a = 0, b = 1, c = 1) */
a:link { color: blue }      /* (a = 0, b = 1, c = 1) */

```
 - The correct order of these rules is the following (the first two rules are interchangeable):

```

- a:visited { color: black }
a:link { color: blue }
a:hover { color: green }
a:active { color: red }

```

JSON

- Lightweight, textual, and platform independent data exchange format.
 - Used for representing structured data.
 - Can be read and written easily by humans.
 - Can be generated and processed easily by computer programs.
- Originates from the ECMAScript programming language.

Comparison of JSON and XML

- JSON can be used as an alternative to XML for data exchange. Provides the same advantages as XML but without its disadvantages.
- The main difference is that JSON is data-oriented, while XML is document-oriented.
 - JSON is the perfect choice for representing data structures. JSON is less verbose compared to XML.
 - Use XML for document-oriented applications. XML is extensible and has a more mature infrastructure.

Convert XML to JSON and Vice-Versa

```
- XML:
<file>
  <uri>http://www.w3.org/Icons/w3c_home.png</uri>
  <size>1936</size>
  <contentType>image/png</contentType>
  <lastModified>2006-07-24T14:58:33Z</lastModified>
</file>

- JSON:
{
  "file": {
    "uri": "http://www.w3.org/Icons/w3c_home.png",
    "size": 1936,
    "contentType": "image/png",
    "lastModified": "2006-07-24T14:58:33Z"
  }
}
```

```
- XML:
<properties>
  <property name="user.home">/home/jeszy</property>
  <property name="user.name">jeszy</property>
</properties>

- JSON:
{
  "properties": {
    "property": [
      {
        "name": "user.home",
        "content": "/home/jeszy"
      },
      {
        "name": "user.name",
        "content": "jeszy"
      }
    ]
  }
}
```

Applications of JSON

- Ajax(Asynchronous JavaScript and XML), Ajaj(Asynchronous JavaScript and JSON)
- GeoJSON
 - Is a format for encoding geographic data structures.
- JSON-LD
 - A lightweight syntax to serialize Linked Data in JSON
- NoSQL databases: a number of document-oriented databases use JSON for storing data.
 - Example:
 - Apache CouchDB
 - RethinkDB
- Web APIs:
 - Facebook APIs
 - Flickr API
 - Reddit API
- Storing configuration data:
 - Package.json
 - Uses:
 - Grunt
 - Visual Studio Code