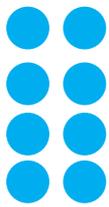


CH 07 DEVELOPMENT

- 7.1 MARK-UP LANGUAGES
- 7.2 CASCADING STYLE SHEETS (CSS)
- 7.3 SCRIPTING LANGUAGES
- 7.4 FILE FORMATS
- 7.5 READY REFERENCE FOR DEVELOPERS
- 7.6 VALIDATION & TESTING
- 7.7 WEB APPLICATION SECURITY



DEVELOPMENT

Use of Open Standard based tools and technologies for the development of websites, software as well as content is very important to interoperability and accessibility of websites. Worldwide web consortium (W3C) is an International body working towards defining standards in web technologies and formats for publishing content on the web. **Department of IT, Government of India is also working towards establishing standards for ICT applications in the Indian Government.** Departments are advised to follow these standards for developing their web applications. However, till this initiative finalises on some standards, Indian Government websites should comply with W3C standards. Most of the browsers, software, companies/communities, also try to comply with W3C standards. Some of the commonly required standards are listed below :

7.1

Mark-up Languages

HTML (Hypertext Markup Language) is at the core of the foundation of World Wide Web. Language has undergone a number of revisions to enable it to be more powerful. HTML 4.01 version has established it as a structural document markup language and is oriented towards the use of Cascading Style Sheets (CSS). XML (Extensible Markup Language) is the means to extend HTML further and make it more generic. XSL (eXtensible Stylesheet Language) is the preferred style sheet language of XML. XHTML 1.0 is an XML based markup language and gives a new dimension to markup languages.

Indian Government websites/web documents/pages/forms should validate to following published grammars:

- HTML 4.01
- XHTML 1.0
- XML 1.0

Web pages should be tested for compliance with validation tools such as W3C mark-up validator. For further details on the above markup languages, visit the website of W3C at <http://www.w3c.org>.



7.2

Cascading Style Sheets (CSS)

Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation of a document written in a markup language. Its most common application is to style web pages written in HTML and XHTML. The CSS specifications are maintained by the World Wide Web Consortium (W3C).

CSS is used by both the authors and readers of web pages to define colours, fonts, layout, and other aspects of document presentation. It is designed primarily to enable the separation of document content from document presentation. This separation can improve content accessibility, provide more flexibility and control in the specification of presentational characteristics, and reduce complexity and repetition in the structural content. CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices.

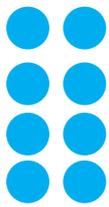
To know more about CSS, visit <http://www.w3.org/Style/CSS/>.

Advantages of using CSS include:

- Presentation of information for an entire website or collection of pages can be held in one CSS file, allowing global changes to be propagated quite conveniently.
- Web browser software can store a copy of the CSS file in the computer's cache, so that it doesn't have to be downloaded each time the user views a web page that references it, hence improving the access time.
- Different users can have different style sheets: for example a large text alternative for visually impaired users or a layout optimised for small displays for mobile phones.
- The document code is reduced in size and complexity, since it does not need to contain any presentational markup.

7.2.1 Therefore Indian Government websites MUST use Cascading Style Sheets as much as possible to control layouts/styles.

7.2.2 Websites that use style sheets should 'degrade' gracefully so that the site remains fully functional even if the style sheet settings are ignored. Therefore the WebPages in Indian Government websites MUST have the same logical order without the style sheets as they have with the style sheets.



7.3

Scripting Languages

Scripting languages are an easy and fast means to enable or include more controls in WebPages. They can be implemented either as server side scripting languages using PHP, JSP, PERL and ASP or as Client side scripting language using JavaScript, Jscript.

- 7.3.1** Server side scripting languages should be preferred over client side since client side scripting may face issues of browser incompatibility, scripts being turned off by browsers, security etc.
- 7.3.2** It **MUST** be ensured that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative accessible page.

7.4

File Formats

Documents form a very important and significant component of Government websites. Indian Government Websites should therefore provide access to documents in appropriate File Formats that are based on open standards and do not impose an unnecessary burden of downloading or acquiring specific software on the intended audience.

When choosing file format(s), Departments should consider:

- Intended use of the material
- Frequency of use
- Accessibility of the format and
- Level of effort and time required to convert the material to the specific format.

File formats for different forms of content are discussed below:

7.4.1 Graphics & Multimedia files

- a.** Sites should have image and graphic components in JPEG, PNG and GIF formats and the same should be compressed without losing on visual quality as far as possible, to allow faster downloads.
- b.** Multiple graphic images at the server may be used (such as providing a thumbnail image with a link to a higher resolution graphic) to make the site more usable even for low bandwidth connections.
- c.** Departments may use Web and multimedia technologies to enhance sites, on the condition that all elements are accessible.



7.4.2 Documents

- a. Government websites shall have a lot of information in the form of documents such as Acts, Rules, Schemes, Gazettes, Forms, Circulars, Notifications. Accessibility & usability of these documents by all citizens is as important as that of the entire website. Department of Information Technology, Government of India is working towards establishing standards for ICT applications in the Indian Government. Standard for documents shall also be defined as a part of this initiative. However till it gets finalized departments MUST either use HTML format or any other format that makes the document Accessible. In case documents are published in a format other than HTML format, departments MUST provide a link to the website from where the document reader can be downloaded free of cost.
- b. When the document has been provided in a format other than HTML, websites should include a text description of the document, including the title, file type, file size, and effective date. This will ensure that visitors have a reasonable understanding of what to expect when they view the document.
- c. Large or complex documents (generally, more than 10-15 pages) should be organized into sections or chapters and linked together. Government Websites should also provide a link to download the entire document since some readers may have high-speed Internet access and thus prefer to print the entire document for later reading.

7.5

Ready Reference for Developers

- a. It must be ensure that in content implemented using markup languages, elements have complete start and end tags, elements are nested according to their specifications, elements do not contain duplicate attributes, and any IDs are unique, except where the specifications allow these features. This helps to ensure that user agents, including assistive technologies, can accurately interpret and parse content. If the content cannot be parsed, then different user agents may present it differently. Some user agents use “repair techniques” to render poorly coded content. Since repair techniques vary among user agents, authors cannot assume that content will be rendered correctly by specialized user agents.
- b. Labels or instructions MUST be provided when content requires user input (for example in forms). Text instructions that describe the input must be provided at the beginning of a form or set of fields. Elements



associated with input must be labeled to ensure that information about the input field is spoken by screen readers when the field receives focus.

- c. In situations where web functions are time-dependent, (for example filling out on-line form) it will be difficult for people with disabilities such as blindness, low vision, dexterity impairments, and cognitive limitations to perform the required functions before a time limit occurs. This may render the service inaccessible to them. It must therefore be ensured that such users are given adequate time to interact with Web content whenever possible. For each time limit that is set by the content ,the user **MUST** be allowed to turn off the time limit , adjust the default setting before encountering it or is warned before time expires and given option to extend the time limit with a simple action (for example, “press the space bar”)

Activities that essentially require a time limit (for example an online auction) or the time limit is too long (say 20 hours) are exceptions.

- d. Many users including the visually challenged cannot perceive shape, size or use information about location or orientation. For such users the content that relies on knowledge of the shape or position of objects becomes inaccessible (for example, “round button” or “button to the right”). Hence It **MUST** be ensured that Instructions provided for understanding and operating content do not rely solely on sensory characteristics of components such as shape, size, visual location, orientation, or sound. Additional information needs to be provided to clarify anything that is dependent on this kind of information.
- e. If an input error is automatically detected, the error **MUST** be described to the user in text. The error message should be as specific as possible. This will ensure that users are aware that an error has occurred and can determine what is wrong. Describing the error in text in addition to highlighting the errors will help screen reader users, who cannot distinguish colour and users with cognitive disorders who have difficulty in perceiving the meaning of other visual cues.
- f. All functionality of the content **MUST** be operable through a keyboard interface without requiring specific timings for individual keystrokes, except where input depends on the path of the user’s movement (for example drawing freehand curves or using handwriting to write).
- g. Whenever a web page is rendered using plug-ins or embedded applications it is possible that functionality of the Web page restricts the focus to a subsection of the content, unless the user knows how to leave that state and “untrap” the focus. This situation may affect navigation for people who rely on a keyboard or keyboard interface to use the Web including visually challenged and people



with physical disabilities. Therefore it MUST be ensured that If focus can be moved to a component of the page using a keyboard interface, then focus can be moved away from that component using only a keyboard interface, and, if it is not possible the user is advised of the method for moving focus away.

- h.** It MUST be ensure that the purpose of each link can be determined from the link text alone or from the link text along with its programmatically determined link context.
- i.** When any component receives focus, it MUST not initiate a change of context. Developers must use “activate” rather than “focus” as a trigger for changes of context. This ensures that functionality is predictable as visitors navigate their way through a webpage. (Examples of changing context when a component receives focus include, forms being submitted automatically when a component receives focus or new windows launched when a component receives focus).
- j.** Entering data or selecting a form control must have predictable effects. Changing the setting of any user interface component MUST not automatically cause a change of context unless the user has been advised of the behavior before using the component. Unexpected changes of context can be disorienting for users with visual disabilities or cognitive limitations
- k.** Metadata adds semantic information to pages and sites and provides contextual information for people navigating the site, especially those with screen readers who rely on things such as page titles, structured page headings and lists. Metadata may also be used by some search engines. Indian Government websites MUST provide metadata for page title, keywords, description and language at least on Homepage and all important entry pages.
- l.** Tables help in organising and presenting data on a webpage, however, many designers in the past have been using tables to make the layout of WebPages. This has resulted in the WebPages not being accessible to people using assistive technologies such as screen readers. For this reason
 - Use of Tables for page layout should be avoided.
 - For data tables, proper tags and markup MUST be provided to identify row and column headers and associate data cells and header cells.
- m.** When users navigate sequentially through content, they should encounter information in an order that is consistent with the meaning of the content and can be operated from the keyboard. Hence if a Web page can be navigated sequentially and the navigation sequences affect meaning or operation,



focusable components **MUST** receive focus in an order that preserves meaning and operability.

- n.** For all user interface components, it is a **MUST** that the name and role can be programmatically determined; states, properties, and values can be programmatically set; and notification of changes to these items is available to assistive technologies.

Note : These guidelines are strictly for the developers. Many guidelines in this section have been adopted from W3C's Web Content Accessibility Guidelines (WCAG2). The details for the same are available at

<http://www.w3.org/WAI/>

7.6

Validation & Testing

Websites should be validated and tested with automatic tools and human review. Automated methods are generally rapid and convenient but cannot identify all issues. Human review can help in issues like ensuring clarity of language and ease of navigation.

Following are some important validation methods that may be followed:

- a.** Departments may use automated accessibility tool and browser validation tool.
 - Validate syntax (e.g., HTML, XML, etc.).
 - Validate style sheets (e.g., CSS).
- b.** It is a **MUST** that Indian Government websites are tested for multiple browsers and versions of browsers, operating systems, connection speeds, and screen resolutions to ensure access by all.
- c.** Use of a self-voicing browser, a screen reader, magnification software, small display, etc.
- d.** Use spell and grammar checkers. Eliminating grammar problems increases comprehension.
- e.** Review the document for clarity and simplicity. Readability statistics, such as those generated by some word processors may be useful indicators of clarity and simplicity. Better still, ask an experienced (human) editor to review written content for clarity.

7.7

Web Application Security

Web Application security is of paramount concern to owners as well as consumers of the website. A lot of security threats are handled at data centres and server administrator level where the application is hosted. Application developers should however be sensitive about security aspects, as a lot of security threats arise due to vulnerability of application software code.

These application driven attacks sometimes turn out to be quite fatal. Best Practices to follow while developing web applications using various technologies are available on CERT-IN website ([http:// www.cert-in.org.in](http://www.cert-in.org.in)) as well as in internet space. Developers should read, understand and follow these Best Practices during development. NIC as well as CERT. IN have empaneled a number of agencies to conduct the security audit of applications.

7.7.1 Each website / application **MUST** undergo a security audit from empaneled agencies and clear the same, prior to hosting and after addition of new modules.

7.7.2 Department **MUST** formulate a security policy to address of various security issues related to the website.