

Nevada Department of Transportation Internet Website Standards

1. **Pages must comply with the [State of Nevada Web Style Guide](#).**
2. **Must comply with Priority 1 checkpoints of the [W3C Web Content Accessibility Guidelines](#) and the federal government's [Section 508](#) requirements. It is strongly recommended that compliance also be met for W3C Priority 2 checkpoints whenever possible.**
3. **All pages must use valid, standard programming code.** The W3C sets and publishes standards for web programming languages, including HTML 4.01, XHTML 1.1, CSS Level 1 & 2, XML, DOM, and SMIL. Programming codes is considered **valid** when it follows all of the rules and conventions specified in the published standards.
4. **Indicate the programming language in your pages by including a document type declaration.** Netscape 6 and above, Internet Explorer 6 on Windows, and Internet Explorer 5 for Macintosh all support DOCTYPE switching. This is a technique where these browsers can be switched from Quirks mode (which emulates buggy implementations in earlier generation browsers) to Standards mode (which more strictly adheres to the Standards). For new content, DOCTYPE use will invoke Standards mode in Netscape Gecko and Internet Explorer 6 **AND ASSIST TESTING FOR COMPLIANCE!!!** This will ensure that your designs work similarly in these browsers as well as in any other browsers which support the Standards.

Example: for HTML 4.01 Strict

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

or for XHTML 1.0 Transitional

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

5. **Test all pages for accessibility, coding compliance, etc.** Use compliance checkers supplied by the W3C, Dreamweaver, etc. to check compliance with HTML standards, CSS standards, accessibility standards, etc. Use browser and operating system accessibility features and leading assistive technology such as screen readers and magnifiers to check for functional accessibility.

Check your pages for validity at sites such as <http://validator.w3.org/> for html and <http://www.w3.org/TR/REC-CSS1> for CSS1. Other check for validity sites are also listed on [NDOT's webstandard links page](#).

Reference: WCAG 3.2, 11.1.

- 6. Tags and attributes must be in lower case.** This is because XHTML documents are XML applications. XML is case-sensitive. Tags like <p> and <P> are interpreted as different tags.

This is incorrect:

```
<BODY>
<P><A HREF="My Link">This is not correct</A></P>
</BODY>
```

This is correct:

```
<body>
<p><a href="My Link">This is correct</a></p>
</body>
```

- 7. Elements must be nested correctly.**

This is incorrect:

```
<p><b>This is not correct</p></b>
```

This is correct:

```
<p><b>This is correct</b></p>
```

- 8. Attribute values must always be quoted.**

This is incorrect:

```
<table width=640><tr><td>This is not correct</td></tr></table>
```

This is correct:

```
<table width="640"><tr><td>This is correct</td></tr></table>
```

- 9. All pages must have a page title.** This title will match the heading title at the top of the content section.

Example:

```
<title>Nevada Department of Transportation Homepage</title>
```

The template for the internet site automatically meets this requirement by including a PageTitle variable which is used in both the <title> tag and using the variable for the heading.

Note the following code excerpts:

```
<% PageTitle = "Put page title here" %>
<title><%= PageTitle %></title>
<h1><%= PageTitle %></h1>
```

10. All pages must have the following "meta" tags:

```
<meta name="author" content="Author's Name">  
<meta name="description" content="Page Description">  
<meta name="keywords" content="Search Keywords">  
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1">  
<meta http-equiv="content-language" content="en-us">
```

11. All code should be written so that it is developer tool or IDE independent.

Here are some examples:

Do not use "Frontpage Extensions" elements in Visual Interdev.

Do not require Macromedia Dreamweaver templates for maintaining consistency across pages.

NDOT doesn't want to be locked into using only one particular IDE for the maintenance-life of a program. This has caused problems in the past. In other words, all web applications should be maintainable using a simple text editor.

Please remember that tools change and that not all developers use the same tools. Have courtesy for the person who must maintain your code using a different tool than you!

12. Modularize code that is used in more than one program and use include or server.execute statements so that changes need to be made in only one place. Pieces of the web page that are used more than once or on more than one page are easier to maintain if they are written in a separate module and then "included" in all the pages using an "Include File" or "server.execute" statement. There is nothing special about an include file; its just a piece of asp or html code. But if you write it generically enough, then you can include it on any page you want as many times as needed.

```
<!-- #include file="includes/MyDataConnection.asp" -->
```

for includes within directory structure or

```
<!-- #include virtual="/includes/MyDataConnection.asp" -->
```

for includes from root structure of web site

Things that work well as includes:

- common functions
- constants
- data connections
- copyright statements
- application or subdirectory navigation
- anything else you need on more than one page.

- 13. Use appropriate markup to convey document structure.** HTML includes markup (programming code) to identify the structural elements of a document. For example, the <p> element identifies a paragraph and <h1> identifies the page heading. Identify section heading, paragraphs, lists, quotes, etc. using the appropriate tags instead of relying on formatting commands to distinguish these elements. For example, use <h1> tags to identify the page title <h2> tags are used to define subheadings within the page rather than simply applying font-size or bold formatting commands. Do not misuse structural elements for formatting effects, such as using <h1> to make text bold or <blockquote> to indent a paragraph that is not actually a quotation.
Reference: WCAG 3.5,3.6,3.7,5.4
- 14. Use style sheets for formatting whenever possible.** Separate content from presentation! Cascading Style Sheets (CSS) is a formatting language designed to complement HTML. While HTML is designed to identify a document's structure, CSS is intended for formatting and presentation.
Reference: WCAG 3.3.
- 15. Avoid using images to display text.** Web developers often use images of text to achieve a specific style, size, or special effect. Users with limited vision rely on the ability to enlarge text or choose enhanced color combinations. When text is used instead of images, style sheets can be used to achieve specific sizes, colors, or effects.
- 16. Avoid using absolute sizes for fonts.** Font sizes can be set using "absolute" or "relative" units of measurement. Absolute units, notably pixels, points, and inches, are based on fixed physical measurements; "relative" units, such as percentages or "small," "medium," or "large" are based on the user's default font size. Users with limited vision often rely on the ability to enlarge text. Most web browsers allow users to easily change the size of text that has been set with relative units (or not at all).
- 17. Avoid using "ASCII art."** "ASCII art" (aka "emoticons") such as ":-)" can be extremely confusing to screen readers. For example, ":-)" reads as "colon dash right parenthesis," and "->" reads as "dash greater than."
- 18. Make sure that links are understandable out of context.** A link is understandable out of context when it clearly indicates its destination or function without requiring additional information. Screen reader users often tab through links (skip from link to link by pressing the TAB key) in order to "scan" a page. Most screen readers offer a "links list" feature to help speed the process of navigating to specific links. Links that are not understandable out of context, such as "Click Here" or "more," make these techniques much less efficient.
Reference: WCAG 13.1
- 19. Frames are not to be used on NDOT's internet web page.** The current site structure does not use frames. It is forbidden to frame the State of Nevada homepage into a frame. Some other reasons not to use frames are: Many browsers cannot print framed pages appropriately. Search engines have trouble with frames because they don't know what

composites of frames to include as navigation units in their index. Frames are difficult to maintain. Frames don't bookmark well.

- 20. Make sure that significant interactions can be performed with both keyboard and mouse.** Scripting languages, such as JavaScript, are programming languages that can be used to automate certain tasks and enable pages to change and respond to user input. Scripts can trigger changes when the user performs specific actions (“events”). Some events are triggered by either mouse or keyboard actions. For example, an image can change color when the mouse pointer hovers over it. Users with physical impairments may be able to use the keyboard but not the mouse. Individuals who cannot see the mouse pointer on the screen also use the keyboard for all interactions. Scripts that only can be triggered by the mouse are not usable by these individuals. Whenever using a mouse-only event (e.g. onmouseover, onmouseout) to trigger a significant script action, also use the corresponding keyboard event (e.g., onfocus, onblur). Also make sure that keyboard events do not unintentionally trigger script actions. For example, keyboard users should be able to arrow through choices or use an access key in a <select> list without triggering each choice (e.g. onchange).
WCAG 6.4, 9.2, 9.3
- 21. Do not automatically refresh the current page.** It is possible to cause web pages to automatically re-load their content on a certain interval. When a page automatically refreshes, it can cause a screen reader to re-start reading from the beginning of the page. Do not use **HTTP-EQUIV="refresh"**. If necessary, provide a link or control to allow the user to refresh a page at his or her discretion.
Reference: WCAG 7.4
- 22. Notify users of time limits and provide a means to extend time if needed.** Some web pages, frequently those that require a user to log in with an ID and password, “reset” themselves after a certain period of inactivity. Typically, any form entries that have been partially completed are erased and the user must start over. Users with visual, physical, or cognitive disabilities may require more time than average to read and interact with a web page. Provide a clear explanation of any time limits and offer the user a way to extend or remove the limits, if necessary. Avoid using time limits unnecessarily.
Reference: WCAG 7.5, 508p.
- 23. Use a liquid layout that lets users adjust the homepage size and minimize the need for horizontal scrolling. Different users have different monitor sizes.** People with big monitors want to be able to resize their browsers to view multiple windows simultaneously. You can't assume that everyone's window width is 800 pixels: it's too much for some users and too little for others. If a web page is wider than the window or screen in which it is viewed, most browsers will display a horizontal scroll bar and require the user to manually scroll to see the entire page. Users may also find repetitive scrolling fatiguing and inefficient. When reading a long page, users read from left to right and must repetitively scroll for each line. When vertical scrolling is used, users only need to scroll AFTER reading the entire screen. Page Formatting: At a minimum, web pages should be viewable without horizontal scrolling on an 800 X 600 pixel screen display.

24. **Use NDOT templates for internet pages. *DO NOT CHANGE THE TEMPLATES OR CSS STYLE SHEETS!*** Use of NDOT templates help assure consistency and a common look-and-feel throughout the site. In addition, the templates include implementation of these standards where appropriate such as meta tags, doctype declaration, standard headings and footings, navigation, etc.
25. **Use color to distinguish visited and unvisited links.** Tell users at a glance which site areas they've already seen. Navigational confusion results when designers disable one of the few useful features of a standard Web browser: having visited and unvisited links appear in different colors. Testing has shown that violating this guideline is particularly harmful for elderly users. NDOT CSS style sheets distinguish visited and unvisited links by default.
26. **Do not use parent paths.** Parents paths use relative addressing such as **../links**. Because of security concerns, NDOT servers have parent paths disabled by default so these types of paths will not work.
27. **Web pages that require browser plug-ins such as Real Video™, Adobe Acrobat Viewer™, etc. should identify the plug-in required on their page and provide a link to where the user can download the plug-in.**
28. **Staging: Staging sites shall be used as a part of the site development and approval process.** Before launching a new web site or redesign of an old site, it is recommended that the site and all its features will be staged and have a review period of at least one week.
29. **Consultants and division personnel should involve appropriate Information Services staff throughout the design, coding and implementation phases of web site development.** Doing so assures that the web application is placed in appropriate places in the site structure, inclusion in site indexes and searches, assistance in testing and meeting standards compliance, etc.
30. **All new content to the internet web site must be approved by the Assistant Director of Administration.**