CONSTRUCTING USABLE WEB MENUS pdf

1: Design - Material Design

When developing a web site, one of the most important things to consider is the navigation menu, to allow your users to find their way around it. It needs to usable, informative, and well implemented, but this can take time. This book will take all the hassle out of implementing web menus, in.

Chock full of copious amounts of code and screenshots, the book offer a good, albeit incomplete, overview of available menu options for webmasters and guidelines for effective menu design. The book is one of the first in a series of how-tos from Glasshaus, a new imprint from Wrox Press. This new series is designed to teach "web professional to web professional" and is slanted towards the more proficient practioners of the craft. The first part of the book deals primarily with usability and information architecture. Menus must be distinct from content 3. Menus must be clearly readable 4. Menus must be easily scanned for information 5. Menus must be easily operated 6. Menus must behave as your target user would expect 7. Menus must load quickly as possible 8. Menus must be consistent across a site 9. Menus must put a higher premium on usability than branding Menus must be localizable Menus must be accessible to the handicapped Menus must work on multiple browsers All good advice. They make some good points, especially that menus be clearly readable and fast loading. Designers would be well-advised to follow their guidelines. Perhaps this was because they decided to include Netscape 4 among their target browsers. Expandable menus are covered, but hierarchical menus get just one screen shot, from MSDN. While some may question the use of slow-loading or overly complex menus on Web sites, hierarchical menus are in use on many popular sites Overall the book gives developers a good overview of menus on the Web, and how to create them.

CONSTRUCTING USABLE WEB MENUS pdf

2: Building Better Web Sites (Learn in Freedom!)

Just menus? Yes. This book focuses on a single, vitally important task that confronts everyone who builds web sites for use by people: how to code and present menus that are easy to understand and use, that convey what users need to know about the many paths they can take to get to the information.

They are the best source of user interaction. User provides his information and input through forms and system interprets this information to fulfill user requests. There are certain items that make a form usable including input controls, input validation, error handling and user feedback. Two important input controls are radio buttons and drop-down menus. Both of them can be used interchangeably when an input is required from user. However, if we study about the usability of these controls, it becomes apparent that radio buttons and drop-downs should be used in certain scenarios to make it easier for user to select a given input. Below are few rules based on this usability study that will help you to decide about one of these controls while designing a form. There is no clear default or recommended option. You want user to read all options. The options are unfamiliar to user and there is less or no chance that he can predict them. When options need to emphasize In example, using a drop-down menu does not seem good as default option does not give a clue about other options. User scans the option easily and quickly. He provides quick response instead of opening a drop-down menu and select from multiple options. User can see them at a glance and compare them easily. Comparing and deciding about an option from a drop-down menu takes time. User has to open the menu and compare options each time he wants to review the selected option. For longer forms, it becomes easier and quicker to scan the options and mark the required ones. Clicking a drop-down each time to select something takes a lot of time. It is obvious from given example that long forms provide good user experience when all select-able options are visible to user. Use Drop-Down Menus Rule 5: When Default Option Is The Recommended Option Viewing only the recommended option makes easier for user to decide the selection because: It is not encouraged for user to change the default option. Default recommended option In example, there is no need to display all options as here is lesser chance for user to change the default option. User can predict them easily. There is no need for him to see options side by side. It becomes cluttered on UI if a pile of options is placed side by side. User can get confused while looking at them. It takes time to scan a large list of radio options. This makes selection easier and quicker. Since forms can be very long with large number of options, it becomes tedious for user if he has to make extra clicks to fill his information. The given set of rules will help you to decide between two controls, radio buttons and drop-down menus, while designing your forms. You can also read other related essays at uxdworld.

CONSTRUCTING USABLE WEB MENUS pdf

3: Best Website Builders of | Ultimate Comparison Reviews

Constructing Usable Web Menus and millions of other books are available for Amazon Kindle. Learn more Enter your mobile number or email address below and we'll send you a link to download the free Kindle App.

Every time a user moves his mouse over a menu item, we need to close all open submenus and check to see if that menu item has any submenus. If it does, we want to display its submenu, which involves computing the correct coordinates for the submenu. The challenge that faces us is being able to determine if a menu has a submenu or not, as well as how to close all submenus. This three-digit number does impose a restriction of at most 1, menu items per sub menu. This process is repeated recursively. To best understand the concatenated id attributes, start by examining table 1, then table 2, and so on. All menus at a given level in the menu hierarchy have an id attribute value with the same number of characters, while all menus closer to the top-level menu have fewer characters in their id attribute value, and while all menus at a deeper level have more characters in their id attribute value. These facts allow for determining if a submenu needs to be display, and for correctly closing submenus when a menu item has the mouse moved over it. The onmouseover event calls the mousedOverMenu JavaScript function, which expects three input parameters: An abbreviated form of the mousedOverMenu function is shown below, with the germane parts in bold: Next, a check is done to see if the menu item has a submenu. This process starts by setting the display and position CSS properties of the submenu to block and absolute, respectively. Next, the coordinates are calculated depending on if the menu items are arranged vertically or not. When the user moves the mouse out of a menu item, eventually the closeSubMenus function is called. A JavaScript timer is used to leave a submenu displayed for two seconds after the mouse leaves the submenu. As we saw, the closeSubMenus function is also called upon immediately entering the mousedOverMenu function. The closeSubMenus function closes all submenus that are at a deeper level in the menu hierarchy than the menu item passed into the function. To see why we only want to close the menus that are in a deeper level in the menu hierarchy assume, for a moment, that the closeSubMenus function closed all submenus. This would be disastrous when moving the mouse onto a submenu, since doing so would cause that submenu to disappear! The code for the closeSubMenus function can be seen below. Note that it takes advantage of the fact that all menu items with more characters in their id attribute must be at a deeper level in the hierarchy. For now, just realize that subMenuIDs contains a list of all submenus. Examining the Plumbing of the skmMenu Control There are three main classes used to generate the menus. First, there is the MenuItem class. Each instance of a MenuItem class represents a menu item in some menu. This string property is automatically set by skmMenu during the databinding process. NET page developer will likely never need to read or manipulate this property. Text This string property indicates the text that is displayed in the menu item. Url If provided, indicates the URL that the end user is sent to upon clicking the menu item. CommandName If provided, the menu item, when clicked, causes the Web Form to be posted back. The relationships among the classes As Figure 5 shows, the Menu class has precisely one MenuItemCollection instance, which corresponds to the top-level menu. Each MenuItem instance can contain an optional MenuItemCollection, which, in turn, contains a set of MenuItem instances, which, in turn, can contain a MenuItemCollection instance, and so on. This object model accurately describes the recursive nature of menu systems that we talked about earlier in this article. NET page developer will only need to concern himself with this underlying object model if he decided to explicitly add or access MenuItems from a Menu. These objects can be automatically built up for the ASP. NET page developer through databinding. An example valid XML file can be seen below. This XML file creates a menu with the following structure: The XML structure allows for an arbitrary submenu depth. This concludes our introductory examination of skmMenu. In the second upcoming part of this article series, the skmMenu code will be examined in depth. NET Web page that utilizes skmMenu. Using skmMenu in an ASP. NET Web page you must first download the code from the end of this article. The code contains a pre-built assembly, skmMenu. You can optionally compile the source code to generate the assembly yourself. If you are using Visual Studio. NET, you can add the assembly to your Web application by simply right-clicking on the References folder in the Solution Explorer and then browsing to

the skmMenu. If you are not using Visual Studio. For those using Visual Studio. NET, you will also want to add skmMenu to the Toolbox. NET Web page, simply use: Figure 6 shows skmMenu immediately after being added to the Designer, while Figure 7 shows skmMenu after the Layout property has been changed from Vertical to Horizontal and some stylistic properties have been set. NET designer after skmMenu has been added Figure 7. The designer after some skmMenu properties have been set To bind skmMenu to an XML file containing information for the menu hierarchy, first create the XML file and place it within your Web application. Customizing the Appearance of skmMenu skmMenu contains four properties that can be used to customize appearance:

CONSTRUCTING USABLE WEB MENUS pdf

Theory of political decision modes How investigating wicca strengthened our faith (and other paradoxes along the way). Categorical legitimacy Bringing up moral children in an immoral world Interpretation of classical electromagnetism Teaching techniques and insights for instrumental music educators Seasons of the Pines A tale of love and darkness book Sri suktam sanskrit 101 commonsense rules for making things happen In the favour of the king Soils and Land-Use Issues in the MacKenzie Hill Country (Landcare Research Science Series,) The First Five (Rollins, Henry) Singapore salary guide 2018 Communicating well Black Jesus and other stories The Kingfisher young peoples book of living worlds The nursing assistant 5th edition Shepherd the flock book Infinite-Dimensional Lie Algebras (Translations of Mathematical Monographs) The tuatara recovery plan Cigar Companion (Connoisseurs Guides) Crafting a Business Preaching, practice and participation East Pakistan the end game 10 easy lessons learn to play blues harmonica Integrated chinese level 1 part 1 third edition workbook Local bonds, global ties : updating the concept of diaspora Frontier historian Mental Distress among Winter-over Personnel in Antarctica Cardinal Newmans Dream of Gerontius. OneKey Student Access Kit How in the World Do We Make a Difference? IV. The literary style. 1930. Breeding British birds in aviaries and cages. Reconsidering Malory Caroline D. Eckhardt Natural history and human evolution [by Olive T. Miller. Molecular biology lodish 8th edition Guide to boundary water routes What makes a great clinical instructor