



The World Wide Web is in the midst of a boom with many brand new websites appearing each day.

IBM's website is one of the largest and most active in the world with a daily home page hit rate in the millions. When world chess champion Gary Kasparov challenged IBM's chess playing computer, Deep Blue, hundreds of thousands of people followed the progress of the match on our website.

Any internet traveler can see that quality varies widely from one site to another. What makes one website enjoyable and informative while another is difficult or unpleasant to use? We have been exploring this question through research, literature reviews, user feedback and our own experiences as we create our web pages. Our exploration has led to the set of guidelines we offer here. They are suggestions for developing a website which is easy to use, communicates effectively and is aesthetically pleasing. If you are developing for the web, these guidelines will make your life easier.

We will be extending these guidelines to support emerging uses of the web, so watch this space for more information.

Principles, guidelines and conventions

Guidelines are the middle layer of an information structure which includes principles, guidelines, and conventions. Together they provide the foundation for good design.

Principles are the goals which guide design decisions. They reflect knowledge about human perception, learning and behavior. They are stated in general terms such as "build on users' prior experience."

Guidelines are based on principles, but are specific to a particular domain of design. For example, a guideline for web design might state "use consistent layout and visual language throughout your site."

Conventions dictate specific design decisions you have chosen to follow. There may be several equally good ways to design a feature, but you must make a choice that aligns with your site goals and audience requirements. A convention might specify "provide search, help, and home page links at the bottom of each page in the IBM HCI site."

We provide principles and guidelines here. You should develop your own conventions.

Our web design guidelines are organized around phases of the website design process - proposal, plan, design, production and maintenance. We hope that you will find them helpful in designing your site.

For more information:

[UI design principles](#) provide a foundation for good web design.

Proposal

Constructing web pages is getting easier all the time as new web-building products become available. But some aspects of building a high-quality website (e.g. defining its purpose and audience, establishing its goals) can't be automated. These elements are crucial to a successful website.

Definition

Specify a reason for the site

When you decided to build a website, you had a reason in mind. Be sure to communicate the reason for the site's existence to every member of your team, and to let it guide the innumerable decisions you'll make while building the site. Whether you want to inform, enrage, entertain, amaze, or promote, all the site's pages must work in concert to support your intentions.

Define your audience

As much as you may wish it could, your site will not appeal to everyone. So your best bet is to choose a particular segment of the

population and focus your efforts on attracting and engaging these users. Begin by exploring these questions:

- Who is likely to be interested in the content you will provide? For example, if you want to sell exotic fruits through your site, you may list restaurant chefs, immigrants from the fruits' native regions, managers of upscale produce departments, adventurous home cooks, and vegetarians as potential customers. At this initial brainstorming stage, don't try to narrow this list; you want to look at the full range of possible users.
- Which of these user groups are you equipped to serve? In the example above, if you have a small farm with limited production capacity, you may choose to exclude buyers for large produce departments and chefs in the larger restaurants.
- Of the remaining list, which potential users are most likely to have access to the web? This question will become less limiting as web access becomes more universal, but you will need to consider it in your near-term planning.

The audience definition will influence every aspect of your website's design. For example:

- Language - which languages will best serve your audience, the appropriate levels of complexity and formality
- Placement - where to advertise your URL, key words to include so that your site will turn up in your target audiences' search results
- Approach - whether playful or scholarly, pictorial or language-oriented
- Input and output media - voice, visual, or both
- Participants - for user testing, focus groups, task surveys

Determine user requirements

What does your target audience want from your website? The most direct way of finding out is to ask them - or, at least, to ask some representative members of the group. Think about the other likely interests of your audience. In the fresh produce example above, you might send surveys to members of:

- a gourmet recipe exchange
- a Latin-American cultural interest club
- a heart-healthy nutrition reader list
- a waist-watchers' society

If the responses from the various groups point in divergent directions, offer varying content tailored to the needs of each group.

Intranet or Internet

A website may be internal (Intranet) or external (Internet). Consider where you can best reach your intended audience, and then choose the appropriate medium. The Intranet is a vehicle for communication within your company or organization. It is a good place for exchanging ideas and for communicating company-specific information.

The Internet is for broadcasting to the world. The Internet is a good vehicle for projecting your company's image and for positioning and marketing its products.

Set measurable goals

You decided at the beginning what your website's purpose would be. How do you know whether you're achieving what you set out to do? Goals are the measurable criteria that allow you to analyze your accomplishments. For example, you may have started your site as a means to increase your company's profits. A measurable goal supporting that purpose could be a 5% growth in net profit for the first quarter after your website goes "live". Usability and system performance are particularly important areas in which to set goals. Establishing goals and periodically checking your progress against them helps you understand what works and where you could make improvements. Plan to construct prototypes to verify that your goals are being met during development.

Analysis

Look at the existing market

Market analysis is a way to learn about the needs and desires of the audience you are trying to reach. It can run the gamut from formalized statistics-gathering and trend analysis by your company's in-house market research team to your own small-scale informal efforts. Even if you don't have the resources for large-scale research, it's still important to get as much information as possible about your target market. For instance, if you are developing a national sales site for pre-teen rock music, you might go to a local shopping mall to observe and record the CD-buying habits of eight-to-twelve-year-olds. But you should also use any connections you might have, such as contacts in other cities and access to kids' music chatrooms, to obtain less locally biased information.

Check out your competition

Whatever topic your site covers, it is likely there will be other sites with much the same purpose competing for the same users. In the virtually unlimited world of the web, people seldom have the time or inclination to visit multiple sites for similar information. To rate a place on your audience's "bookmark lists", your site will have to beat the competition in at least some of the following areas: usefulness of content, quality of presentation, ease of use, timeliness, and fun factor.

To find out what your competitors are doing, you can use your site's keywords to search for other sites with a similar focus. Bookmark the sites whose themes are most similar to yours, and assess their strengths and weaknesses. The following table is a simplified example of one approach to organizing competitive analysis data:

	Usefulness	Presentation quality	Ease of use	Timeliness	Fun
Company a	4	8	2	6	9
Company b	8	5	6	8	2
Company c	6	4	4	8	1

Knowing what you're up against will help you decide how to design your site, where to concentrate your efforts, what information or services to provide, and which visual and interaction styles to use. Remember -- the web is a dynamic medium, so check these sites frequently to see how your competition is advancing.

Assessment

Know your limits

Every website is subject to constraints of some kind. It's important to recognize the limitations that apply to your project, so that you can make a realistic plan. Some common constraints in creating a website are:

- Time - if your site is related to specific event or release of a product, you may have to work within strict time constraints
- Money - most individuals and companies have a limited amount of money to invest in website development. And, of course, your budget will be a factor in almost every major decision, from the hardware you can use to the number of skilled people you can hire for your project.

Know your site's ongoing requirements

After the initial site-building work is complete, a website will require a support infrastructure. Plan to have some team members obtain the following elements during the proposal, planning, and design phases of construction:

- Hardware - client, server, network lines, phone lines for receiving feedback
- Software - search engines, authoring tools, visual design tools, scripts to support feedback mechanisms, and database software
- Manpower - skills to maintain the server, technicians for phone support, online support

Know your clients

Determine what equipment and software your users will use to view your site. You may need to know what browsers they are using, whether they have Java support, their connection speeds, the capabilities of their workstations, and whether they use special software such as screen readers.

The client environment is typically very heterogeneous; the various hardware and browsers your audience may use can lead to wild variations on your beautiful site design. At present, there is no perfect solution to this conundrum, but you may choose from the following approaches:

- Have the server test the client capabilities and adjust for them. For example, determine which browser each user is running, and send only data appropriate to that browser.
- Provide users with options. For example, let them choose to view your content with or without frames.
- Design to a certain standard and accept the fact that not everyone will be able to use your site. For example, you may decide not to support older versions or less capable browsers.
- Design to the lowest common denominator. This will greatly limit what you can do, but you can be sure everyone will be able to view your content.

Know your server

Understand the capabilities of your server and decide what programs to run on it. For example, Java applications can be distributed with some code running on the server and some on the client. You can split the processing in ways that optimize efficiency and security.

Make sure your server has the capacity to handle the number of requests you expect with acceptable response times.

Some server configurations to consider are:

- Personal computer with local storage, file system, tools, and a network capability for displaying, creating, and maintaining files on the web. This configuration can be costly in terms of the amount of human resource it takes to maintain a website. However, this is probably the best solution for small websites.
- Personal computer connected to a remote server with database technology. This configuration provides the most value to large, interactive websites. Databases such as DB2 provide security, transaction processing, and replication. Lotus' Domino(tm) product goes a step further whereby users participate in the web creation, modification, and the processes through which the information travels. For more information on database technology see the Products section of the IBM website (WWW.IBM.COM).
- Network computers (NC or NetNC) with remote storage, remote application processing. Network computers offer device manufacturers almost complete freedom in choosing processors and operating systems. All network computers must be capable of understanding standard Internet protocols (HTTP, FTP, etc) and run programs written in Java. Network computers provide the inherent manageability of having data and applications stored on a central server. The value in employing an NC strategy is that all development effort is centralized on a server thus driving down hardware costs.

These configurations all provide a benefit to you depending on the size of the website you are developing and how you plan to grow your website.

Know your skill requirements

Designing for the web requires multiple skills and activities. No major site can be developed adequately by a single person. Determine the number and types of people needed to design and develop your site, the activities they will perform, and when each is needed. This knowledge will allow you to bring in the right people at the right time and get the work done on schedule. The following table details the typical skills mix needed to create, build, and maintain a website.

Project management	Usability engineering	Visual design	Content management	Website engineering
Initial client contact	User requirements	Art direction	Copyright release	Hardware configuration
Business requirements	Functional specification	Page layout design	Contract negotiation	Software installation
Project plan	Task analysis	Interface design	HTML writing, editing	Custom applications
Contract negotiation	User profile survey	Prototyping	Filtering	Implementing website
Budget	Interface design	Design specification	Documentation	Usage statistics
Schedule	Testing, collect data	Style guide	Online help	Java, LiveScript, etc
Resource allocation	Error handling	Collateral	Resource allocation	Maintenance

Plan

Planning is the preparatory work which should precede the design of a website. A good plan will ensure that website development proceeds smoothly toward the goals you have set for your site.

Schedule

Determine site content

Based on your audience definition and its requirements, create an outline of the content your site will supply to meet those needs. If some of the content is readily available, note within an outline the location of each pre-existing section. Assign areas of missing content to appropriate team members, and remember to include their content-creation time in your schedule.

Site content tends to fall into three categories:

- Initial - what must be there when the site is published
- Core - what information will remain there through refreshes and redesign
- Transient - temporary items

Also, you should determine what is missing and plan for how you will acquire that information.

A website can be a two-way street. While you don't want to post a site full of blank pages, you should expect that visitors to the site may want to reciprocate by providing you with information as well. For example, if you post recipes that use your produce, you might also encourage readers to experiment and send you their original recipes.

Assemble team

After you identify the skills you need, assess any currently available team members and recruit the people to provide the missing skills. Clearly communicate to the team the purpose of the site and the goals you are trying to obtain. Should you be missing some of the critical skills, you may choose to enlist the services of a company that specializes in web design. You can always bring the task in-house at a later date.

Detail scheduling

You probably have some timeframe in mind that you are targeting for publishing your site. Create a schedule that provides time for planning (including the time it may take to get permission to use other company's content), designing, producing, testing & iterating, and validate the schedule with the members of the team. Schedules help a team focus and they can use the target dates to revisit their progress and determine if they will meet or beat their schedule.

The following is a sample schedule detailing specific checkpoints throughout the development process. Your schedule might vary depending on the size of the site you are developing.

Create proposal	July 10
Set schedule	July 24
Information structure defined	August 4
Visual/interaction style defined	August 14
First interim design review	August 21
Production begins	August 23
Interim design review	August 30
Interim design review (final comments)	September 7
Final testing begins	September 14
Site transferred to server	September 21

Rollout

You may choose to publish your site in stages, beginning when all your initial content is ready. You will need to plan this staged rollout ahead of time so that this will be an orderly process where the site is cohesive at each stage and there are no references to information not yet available. Requirements should be set for each stage of the rollout.

Make a contingency plan

If something can go wrong it will. Create a contingency plan for your content. You don't want to have 99% of the site ready to go and be held up waiting for one contributor who didn't come through. Decide which content is critical and manage that closely. Consider alternative content that may be substituted, if necessary, or plan for ways to drop some content without ruining the structure and purpose of the site.

Approach

Formulate presentation approach

So you have a set of information (at least initially). Now you should decide on how you want to present it to your target audience. Is your site going to be used to sell prototyping software? Maybe you would want to use a high tech theme. One careful consideration is that you carry the theme throughout the site and try to avoid clashes between the content and the presentation style. For example, if you are providing medical information to doctors, you probably don't want a comical theme supporting that type of information.

Presentation technology

Standard HTML

The surest method of conveying information on the web is still through standard HTML. HTML-based text is the only information you can be sure that everyone can read.

Use text rather than graphics for most of your content. The HTML 3.0 standard offers many options for formatting your text. You can

change the font **siZe**, the font **color**,

and

the

alignment.

As the previous paragraph illustrates, you can create some pretty abominable designs with standard HTML. It is best to be a little conservative with standard HTML text and let the page layout and graphics establish the "coolness" factor.

Graphic elements

Some of the main graphic elements on the web are GIF, JPEG, PNG, and FlashPix.

The GIF format is good for simple graphics that use low color depths. Large areas of the graphic that use the same colors compress the most. The largest number of colors that a GIF can contain is 256, but using fewer gives you a smaller file size. The example below shows how a flat colored image benefits from being saved as a GIF rather than a JPEG.

**4 Color GIF - Only 2K****Low Quality JPEG - 7K**

GIFs allow you to specify transparency and interlacing. Use transparency when you want a graphic to look like an object rather than a rectangle. This is particularly useful when placing graphics on top of backgrounds. Use interlacing when you want the graphic to fill its area faster and become more focused as it loads. This effect is useful when you want your graphics to fill the page so the user has something to watch and is less likely to become impatient with a slow download.

Use the JPEG format for complex photographic images that demand high colors. While JPEG requires larger file sizes than GIF, you can experiment with different compression levels to get the smallest file possible. Below is an example of a highly compressed JPEG that shows little quality loss but great file size savings.

**JPEG - low quality 5k JPEG - high quality 9k**

Animated GIFs

Animated GIFs allow you to add animation to a web page without the overhead of mechanisms like Shockwave or Java. The bouncing ball below is an animated GIF.



Evaluation

Identify user evaluation opportunities

During development of your website, find ways to let users validate your design and content (they should be a representative sample of your target audience). Some examples of user evaluation opportunities are:

- Early - get in on a discussion group of people who match your target audience. Post a preliminary table of contents and ask for comments, suggestions and areas of interest not covered.
- Middle - circulate sample pages as they are completed and give a brief questionnaire.
- Late - give your readers the opportunity to comment on the site after its released.

User evaluations should be included in each phase of your web design plan. The process we use is called [User-Centered Design](#).

Design

The design of your site should be based on user requirements. The process we use is called User Centered Design.

Structure

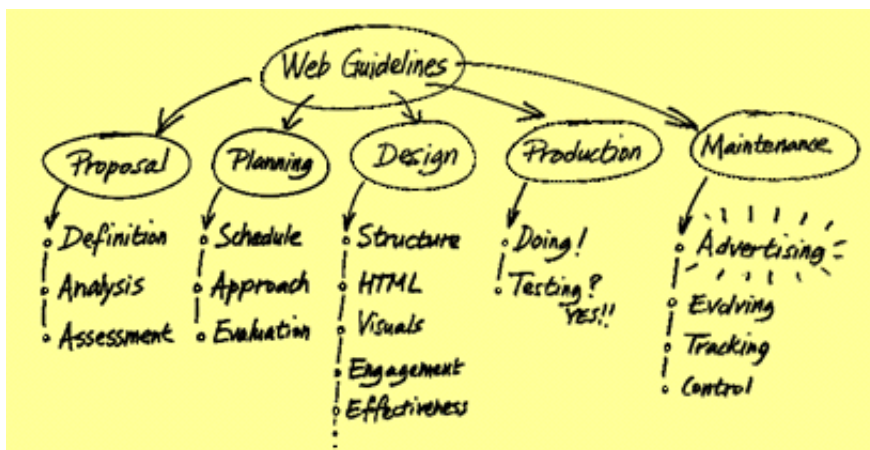
Model/view separation

Model/view separation is the practice of keeping content and presentation independent of each other. This practice is used in object-oriented programming and interface design, but is especially useful in designing websites where contents and views should be updated frequently. Develop your content with few, if any, dependencies on the method(s) of presentation. You can then change the view of the information without requiring changes to the content, and vice-versa. Maintaining model/view separation makes it easier to keep your website's visuals fresh and your information up-to-date.

Navigation structure

Develop a navigation flow that your users can follow easily without getting lost. The best way to get the big picture is to sketch out the flow relationships, either using a flow-charting tool or a simple paper-and-pencil sketch. Define the overall navigational structure of your website early, and put your content into this structure as it is created. This strategy will help you see how and where your content will appear in the user's context, preventing last minute surprises.

The following example shows a flow diagram for these web guidelines.



Structuring the information space

Web users often report being "lost in hyperspace". If users can't visualize the organization of web pages, they will not be able to navigate easily. A good solution to this problem is to use an organizing metaphor to group related pages into a coherent site. For example, the guidelines you are reading use a notebook metaphor to group a set of related pages. The links represented by chapter tabs and section buttons provide the feel of bringing information to the user rather than jumping around through a space as separate pages often do.

Another common structure for a website is to have one or more navigation pages to guide the reader to the content pages they are interested in. This is a hierarchical navigation approach. It works as long as the hierarchy is not too deep (ideally 3 levels or less) and the categories make sense to users. Many other structures can also work depending on the purpose of your site and the users you are creating it for. You should conduct usability testing on the information structure which you think best fits your situation.

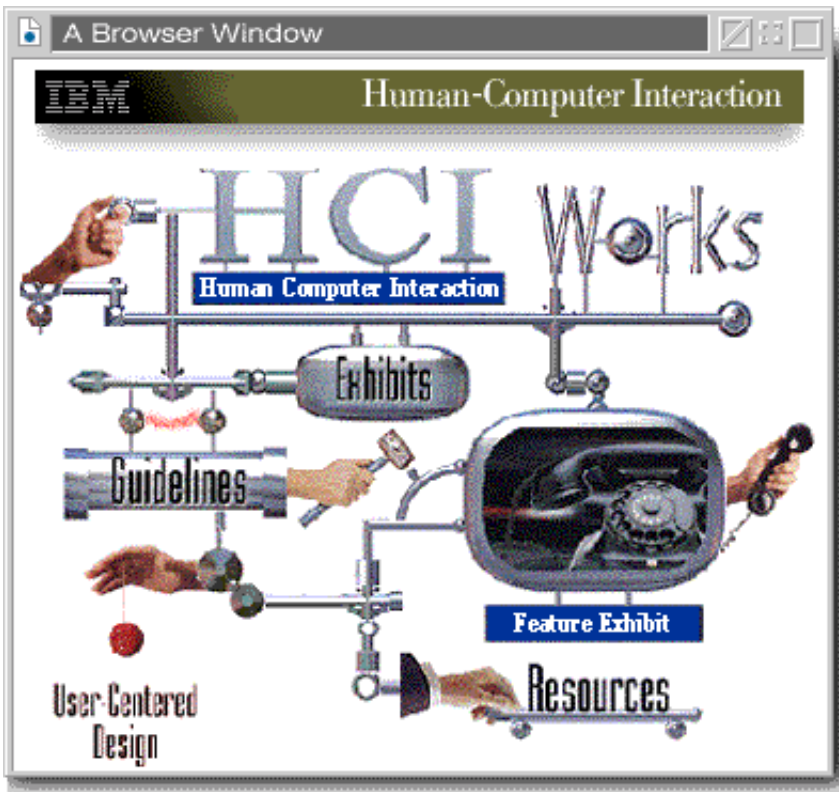
Defining page types

The following information describes the basic types of pages within a website.

Home page

The home page is the central page of your site from which all other pages can be reached (directly or indirectly). It is typically where you want your readers to enter the site. Therefore, it should be the place where you introduce your site and establish your identity.

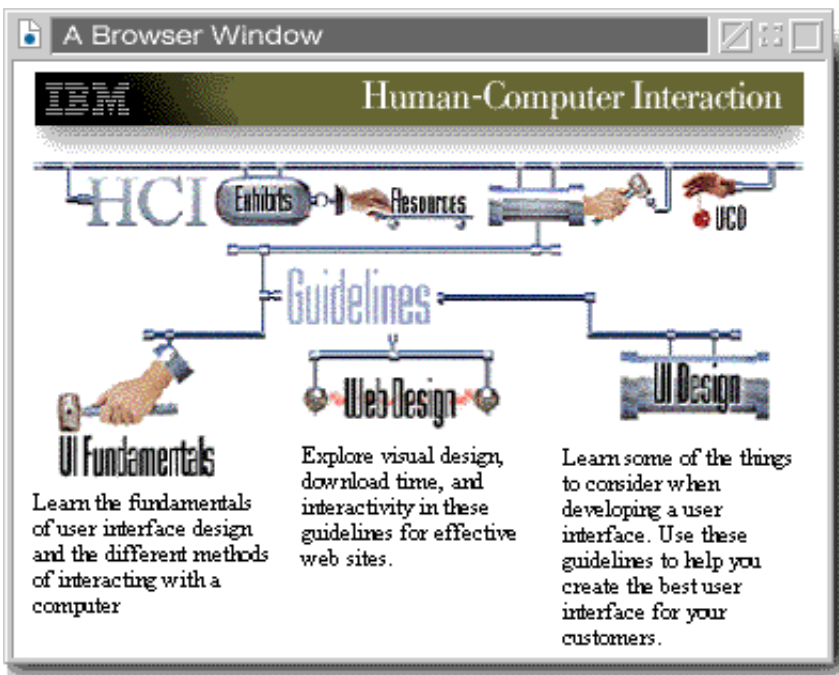
The following example shows the home page for the IBM HCI site. It provides links to the content categories for the site.



Navigation page

Navigation pages have the primary purpose of directing the user to the proper area of your site for the information they are seeking.

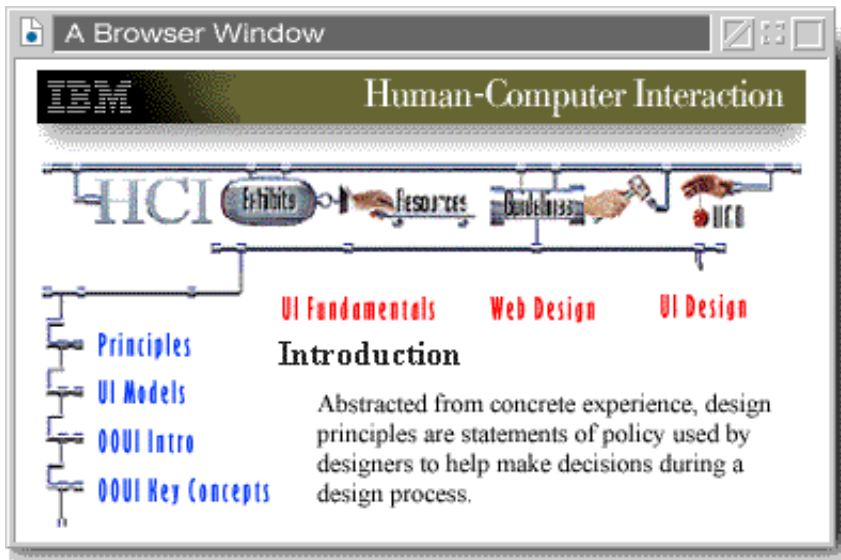
The following example shows a navigation page which guides the user to various types of guidelines on the IBM HCI site. Each link provides a brief description to help the user determine the link(s) they want to follow.



Content page

Content pages provide the information users are seeking when they visit your site. They may also contain navigational information to give users a sense of location within the site and allow them to progress to more information or return to a previous page.

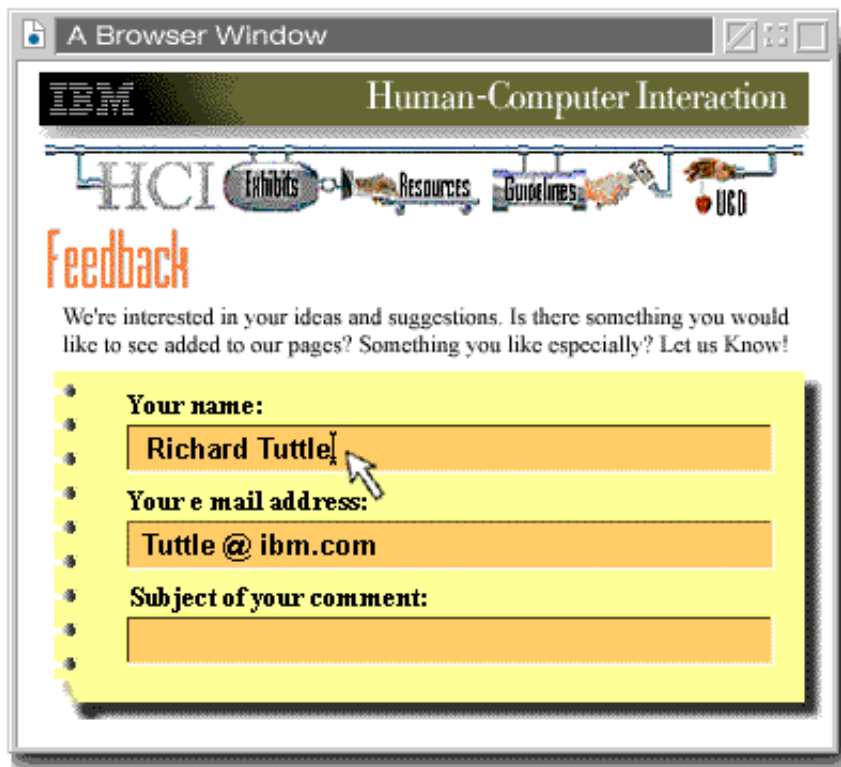
The following example shows a content page from the IBM HCI site. It provides information while showing the reader where the page fits into the site structure as well as providing links to other parts of the site. This demonstrates how a page can serve as a content and navigation page at the same time.



Input page

The purpose of an input page (also called a form) is to collect information from, or establish a dialog with, the user.

The following example shows a page for collecting user feedback.



For more information see:

[User Centered Design](#) (UCD) includes surveys and information on UCD topics.



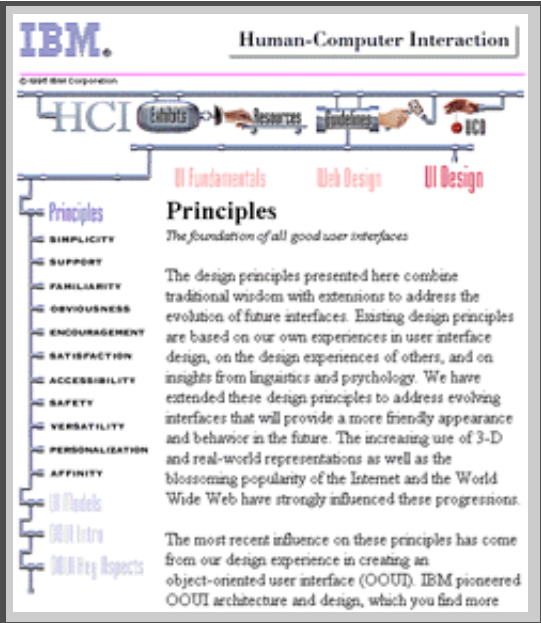
HTML design strategies

HTML is a markup language for the web. It is an evolving language; to increase its capabilities, browser suppliers are making non-standard extensions to HTML. However, using these extensions can make your page unreadable to users of other browsers. You should either use standard HTML or analyze the impact on your audience of using an extension. Estimate the number of readers you will lose and weigh that loss against the benefit that an extension provides.

Tables for layout

Currently there are a limited number of presentation techniques available for the web. Because control over page layout in HTML is crude

at the present time, many people use tables with the borders turned off to force a structured layout. With enough tweaking, a table can emulate nearly any design you can imagine. For some effects, you may want to nest one table inside another -- coding an entire table inside the cell of another table. Our principles page provides an example of this technique. Also, use the SPACER element or transparent GIFs to control the white space in your page layout.

	<pre> <table width="477"> <tr><td colspan="3"> ...banner</td></tr> <tr><td colspan="3"> ...site nav</td></tr> <tr><td colspan="3"> ...titles </td></tr> <tr><td width="131"> <table width="131"> <tr> ...Principles menu</tr> </table></td> <td width="10"> ...white space</td> <td width="335"> ...page content</td></tr> </table> </pre>
<p>Tables for Layout The image on the left shows an example of a page in our HCI site created with nested tables. On the right is a simplified version of the HTML used to implement the desired layout.</p>	

Frames for layout

Frames are an extension to HTML that some of the most popular browsers recognize. Frames can be very useful for displaying information that you want to keep visible at all times, such as navigational elements.

Newer browsers allow for borderless frames, which you can use to make your information look cohesive. The page you are viewing was composed of several frames. As the percentage of browsers that recognize frames increases, they will become a more viable option for laying out information. However, you can design a page using frame tags and include the NOFRAMES attribute to support an alternate view for browsers that do not support frames.

Visuals

Navigation and information content

Websites consist of two categories of elements: information-based content and the navigation and browsing mechanisms. The goal of good visual design for the web is to insure that both categories coexist to deliver a balanced presentation to users. In a successfully designed website, the navigation scheme is visible and easily understandable, yet leaves plenty of visual bandwidth for an attractive content presentation.

The following information explores:

- visual design issues related to navigation elements
- issues relating to the design of editorial contents
- how the layout of a website weaves the two together

Make the navigation scheme visible

As seen in previous sections of these guidelines, two navigation structures exist in a web document. First, the hypertext links that provide references within the site and to other sites. Second, the navigation structure established by the designers of the site that is tightly related to the structure of the information and contributes greatly to the quality of its presentation.

Hypertext links tend to take the audience away from the presentation and foster unstructured browsing. The navigation structure established by the designers of the site provides a framework within which the information can be delivered efficiently. The role of the visual designer is to make this framework clear and visible.

When the navigation framework of a site is clear and visible, users navigate easily through the presentation and resume their structured browsing without confusion. A well-visualized navigation structure provides the backbone to the presentation and contributes to giving it the sense of place that is necessary in the unstructured web.

Visual formula for a navigation scheme

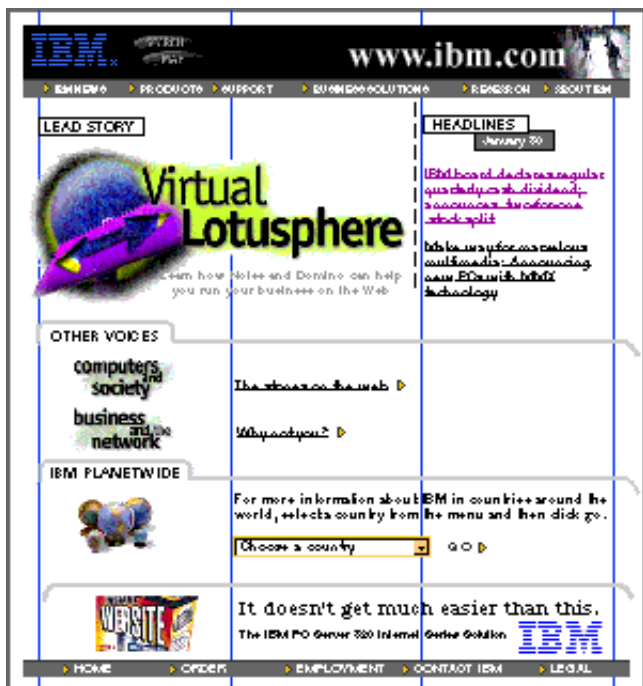
The navigation framework is made visible by representing the navigation elements consistently across the site. A representation formula is

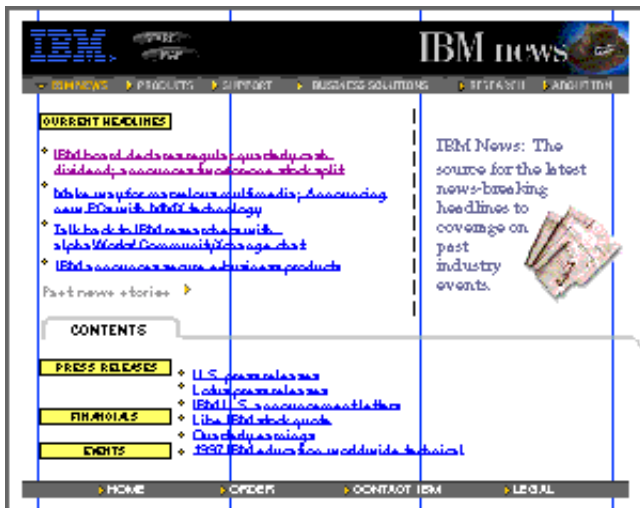
designed and applied to all the navigation elements in the site.

An elementary formula would be to represent all navigation elements the same way. For example, using a blue button with a label indicating the destination of the link. Such a formula would insure that the navigation elements are easily found by users and that their destination is understood. But it wouldn't inform users about the configuration of the site and would not contribute much toward giving the site a recognizable identity.

Formulas that work harder for the benefit of users can be designed. The visual designer can find opportunities for informative visualizations by observing the configuration of the navigation structure, its different levels, and the breakdown of information into topics.

For example, designers of the IBM HCI website (that directed you to these guidelines) created a versatile visual formula for the sites' navigation elements that makes the configuration of the site fully visible from the home page. The home page constitutes a high-level site map informing users about the full contents of the site. It strongly establishes the mode of navigation for the rest of the presentations visual identity. As the following illustrations demonstrate, the solution can be easily applied to the several levels of the sites' navigation structure - which is the sign of a complete solution.





Affordance

An interface control has affordance when a user can infer its function from its appearance. Navigation elements in a website must provide affordance. Their appearance should help users understand that their purpose is for navigating through the presentation.

The mouse pointer change provided by web browsers to indicate that the element pointed at is a link is not sufficient. Relying only on it would force the user to use trial and error to discover which elements are links and which are part of the contents. The convention provided by HTML for hypertext links (colored, underlined text) is reliable, provided that the audience has learned it, but is graphically limiting. The designer can use differences in size to establish a hierarchy between links, but HTML text has poor graphic quality and doesn't allow much visual characterization.

As shown in the previous example, in current state-of-the-art websites, rich imagery is used to implement the navigation elements. Navigation elements can take many different appearances and the visual designer must not lose sight of the affordance requirements. As part of every new design solution, the designer will explain what will help users differentiate links from information contents. Differentiation between navigation elements and information is indeed the main affordance problem to be solved. The audience knows it has to click to navigate, the question is where.

Visual feedback

Visual feedback confirms that an action initiated by a user has been completed. For instance, in a graphical user interface a button is made to look pushed-in when a user clicks on it. In web design, visual feedback on navigation actions has been crude. The fact that a new web page loads slowly is usually the only feedback that occurs, and it is only a side effect of a slow connection.

Useful visual feedback can be provided by including it in the visual design of the navigation elements. Within a site, visual connections can be established between web pages at both ends of a link. For instance, in the IBM HCI site shown above, links in the source web page have the same appearance as in the target web page, but are shown expanded. This visual feedback conveys that users are following the continuity of the presentation.

Visual design of the information content

This section discusses the various aspects to bringing the content and the design formula together.

Design approach

The contents of a website span between simple text and rich multimedia, combined with various kinds of downloadable applications. But, however sophisticated the contents of a site are, solid editorial design methods will bring about the best solutions. The task begins with putting together a design approach. A design approach brings focus to the design effort by describing in a few sentences the main components of the design solution. A design approach can be formulated by looking at the factors that drive the project:

- intended audience
- purpose of the site
- possibilities offered by the supporting platform
- the nature and structure of the information to be presented

The tone of the presentation can be decided by understanding the audience's cultural make-up and to which age group it belongs. The purpose of the site also influences the tone of the presentation and decides how hard the presentation needs to work at retaining users. By knowing the supporting platform possibilities it can be ascertained that the proposed solution can be successfully implemented.

By examining the nature and structure of the contents, designers and editors of the site can identify opportunities for an original and lively presentation. Opportunities will come from the kind of imagery usually attached to the domain of the presentation; from key concepts that may have to be demonstrated or illustrated; from a special angle that the authors may have chosen to use and that can be reinforced by the visual design.

Layout

Once a design approach is formulated and agreed upon, the designer can proceed to create the site's layout. Designers usually go through a phase of rapid sketching where they try out different ideas without committing too much time and resource. Soon, a layout that has potential is developed further with tighter sketches, then typical pages can be prototyped in HTML.

The task of creating an effective layout necessitates professional skills and a course in these skills is not within the scope of this presentation. In this section we will list and explain some of the attributes that make a good layout for a website.

Visual identity

It is a fact of life in the web world that audiences can enter and exit your presentation with the click of a mouse. The place they have been before and the place they are going to will do their best to compete for the web surfers attention. A strong visual identity will prevent your presentation from vanishing in the blur of the web.

Companies use their brand recognition and include their corporate identity graphics in their website. Visual identity can also be built by making clear choices in the visual elements that compose the site. One or two type families should be chosen (this site uses Bodoni and Futura) and used for all typography that is not simple HTML text (headlines, some level of sub-headings, captions and liftouts). A flexible color scheme should be devised and applied faithfully throughout the site. The use of typography and color in the design should be deliberate and intentional so that a personality can be perceived by the audience.

Continuity and consistency

A website is a constellation of related documents that have to be held together visually. The layout formula that is chosen must be flexible enough so it can be applied consistently to the many situations that occur in the presentation. The formula must be applicable to a strong home page as well as to quieter, information rich, content pages. The visual continuity that results will contribute to the sites' visual identity and convey that site as a unique place on the net.

Rhythm and flow

Like all pieces of editorial design, a website should not be monotonous. Even if the presentation tone elected in the design approach is conservative, visual rhythm must be applied. Opportunities for emphasis are to be found in the information contents, and can be implemented through illustrations, typography, or multimedia.

Good management of space

Screen space constraints greatly influence the layout of websites. The layout formula must insure that the portion of a page that is visible in the image safe area upon load is meaningful. Forcing the audience to scroll left or down at the onset has negative effects. By establishing a grid that takes into account the basic unit of space that is the minimal image safe area, the negative effects of scrolling can be minimized.

Because the web is document based, it is often the case that interactive controls are woven with informational content and can scroll out of site as the document is read. The designer must choose which elements should not be allowed to scroll away, and use the available HTML techniques to keep these elements in the audience's sight.

Economy of visual means

Web pages must download fast. Visual impact must be achieved with as little data as possible. For the layout of each page, the designer must carefully decide where and how to "spend" the small credit of kilobytes that is optimal. With a little know how about graphic file formats and their compression capabilities, the designer should be able to decide on the tradeoffs to be made to execute a design.

Engagement



Vary your content

One way of getting users to return to your site is by improving and adding new content. You should make it clear what has been added so that the reader does not have to search through the site trying to find new content.

One technique for highlighting new content is to feature it on your home page or a dedicated page in the hierarchy. The following example shows a list of feature exhibits recently added to our website.



Provide a way to get user feedback

Users can tell you when something goes wrong with your site, provide suggestions on improvement, or give you a pat on the back for a job well done. As with any user interface, feedback from your users is crucial to designing to meet users needs. A good way to solicit this feedback is to provide a user response form accessible from any page in your site.

Our feedback form is available from all our pages including these guidelines. How are we doing? Use the feedback link at the bottom of this page to tell us.

Effectiveness

Present your message efficiently and effectively

Be brief and to the point. Edit your text down to the minimum required to make your point clearly. A web page should cover a single topic with links branching to related information. Keep your page length short. Long pages keep the user waiting while information transfers and they take too much time to read. If possible, design a page to be viewed without scrolling.

One good technique for keeping your page short is to use the inverted pyramid writing style. Newspaper reporters use this style of writing to let busy people get the information they need quickly. Plant the hook -- the most interesting point you want to make -- in the first paragraph, then use subsequent paragraphs to provide more detail.

Call attention to the most important information

Put the most important information near the top of the page where it can be read without scrolling. Place information in priority order from top left to bottom right (for left-right reading countries). Use position, font size, and color to draw attention to the headings and other important information. Order information according to criteria such as sequence of use, frequency of use, and importance.

Follow a consistent writing style

Use consistent voice, grammar, and notation throughout your site. Spell-check your work and ask others to read and comment on it before putting it out for the world to see.

Use controls consistently

The user interface controls on web pages should be used consistently as with the same controls in other desktop applications. For guidance on this and other design information, consult a user interface style guide such as Common User Access (CUA).

Use graphics sparingly and for effect

Graphics sometimes convey information more efficiently and clearly than words. Use graphics when they are the most effective way to communicate, but do not use them gratuitously. Unnecessary graphics distract the reader from the message and delay transfer of the Web page.

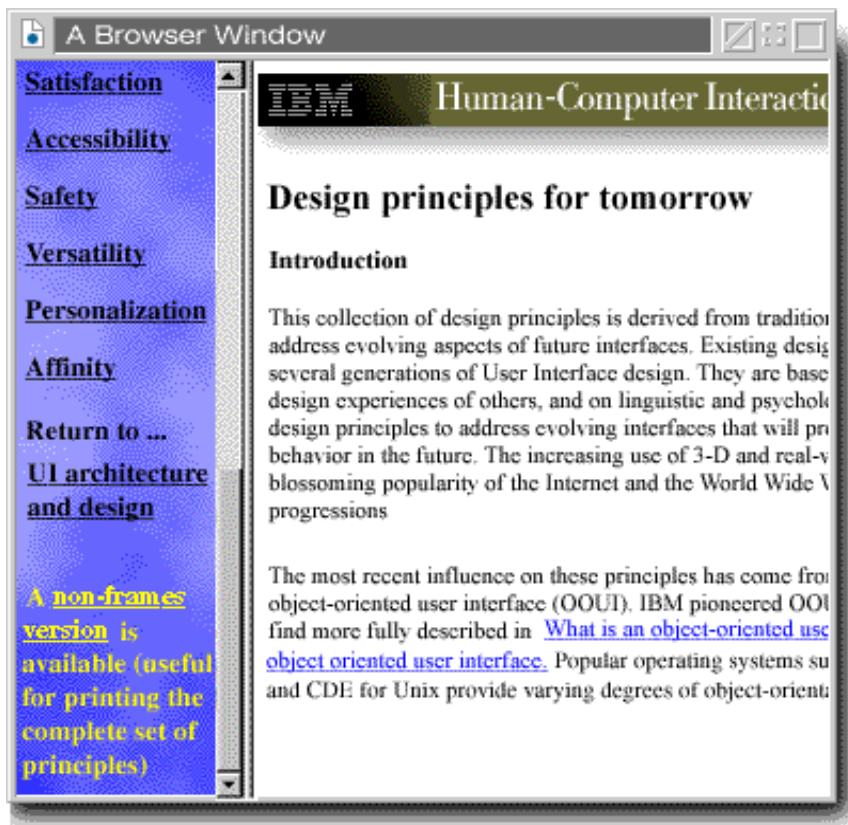
Take advantage of graphics more than once. In this site, we have used graphics multiple times. When a browser comes across a reference to

a graphic that is already loaded, it shows it immediately rather than wasting time downloading it again.

Design for printing

Readers may want to print your web pages. Some information is appropriate for printing and some is not. For example, a page designed to present streaming audio or video can not be printed in a meaningful fashion because the data are not amenable to the print medium. On the other hand, text that users may want to keep as reference material is appropriate for printing. If your users will want to print your Web pages, you should design the pages so that they format well in printed as well as on-screen form. If the reader wants to print multiple pages as a collection, you should provide a means for them to do so.

The following example shows how we provide a link that generates a printable version of the design principles at our IBM HCI site.



For more information see:

[Common User Access \(CUA\) Style Guide](#)

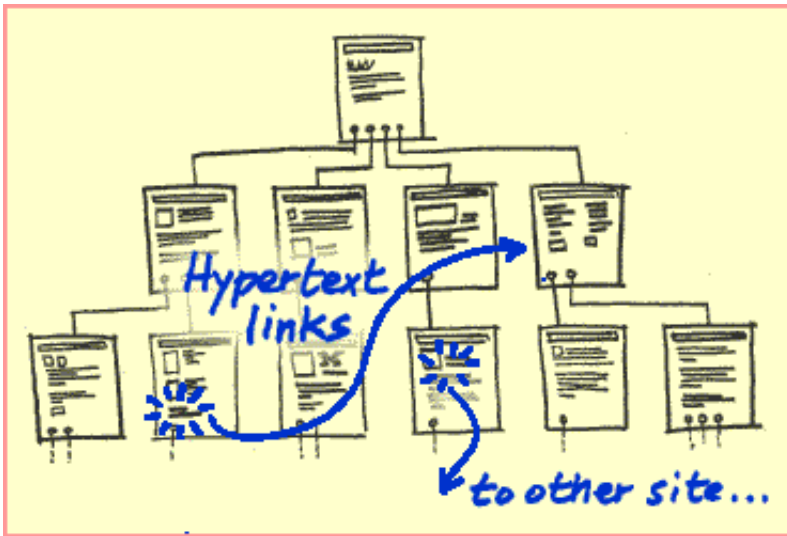
Navigation

Navigation cues

Your users need to know where they are and how to get where they are going next. Show users where they are by clearly labeling the current location within the site.

Provide clear, simple ways to get to other parts of the site. Some good ways to provide navigation to other parts of your website are through navigation links, site maps, and search mechanisms. Traditional publishing techniques for organizing, such as an index or table of contents, are also useful.

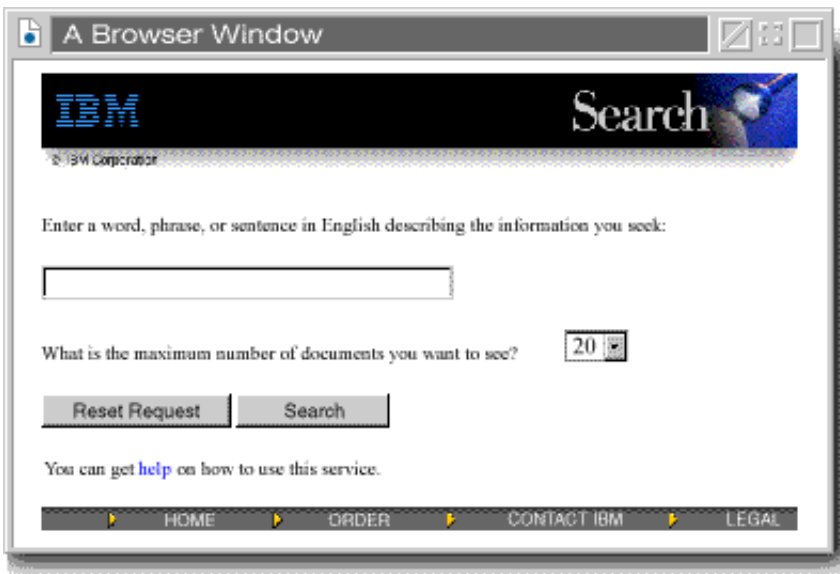
The following example shows a hierarchical organization of web pages with links to navigate the hierarchy.



The meta information in your HTML document defines how your pages will be found. Include the attributes TITLE, DESCRIPTION and KEYWORDS in your meta information so that search engines can pick up your pages and display them to the appropriate audience. You can see examples of meta information by viewing the document source from your browser.

Avoid overloading the KEYWORDS attribute with one particular word to insure the site is picked up by search engine. Use keywords that identify the page.

The following example shows a search page that can be used to find all the pages related to a given topic on the IBM website. The value of a site-specific search engine is that users can get to the information quickly.

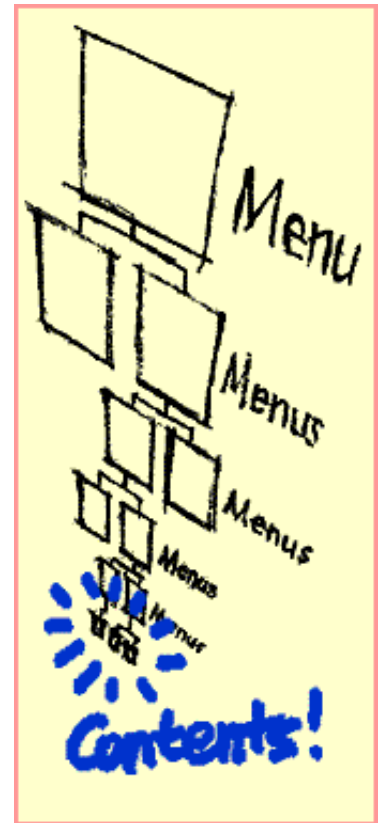
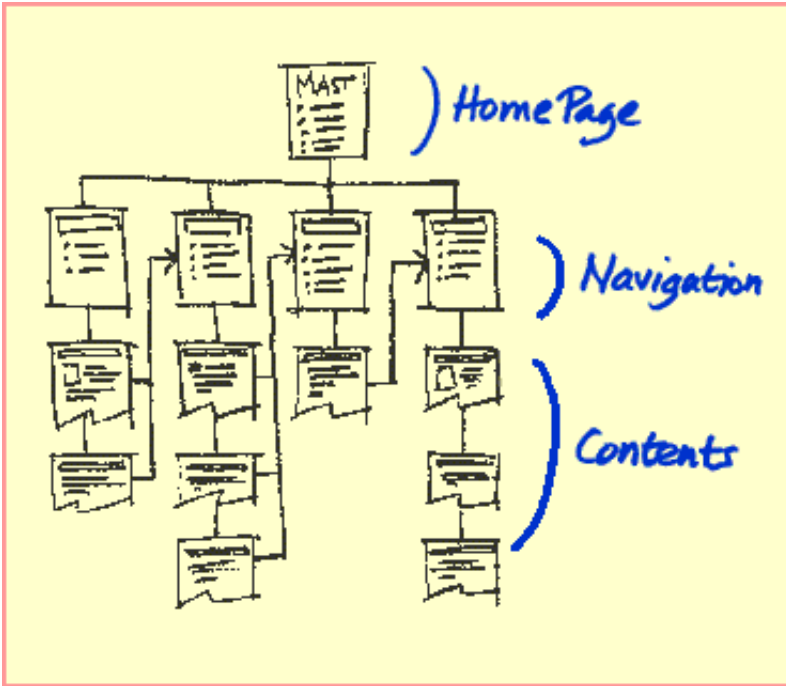


You can minimize the need to navigate by presenting information concisely and by matching your web page to the likely window size (so that scrolling is minimized). The example shows how it may be necessary to scroll in two directions if your page is larger than the browser window.

When navigation is necessary, minimize the number of steps required. For example, provide links within long pages and keep the navigation hierarchy as flat as practical. There should be no more than five levels and preferably fewer than three.

Group related pages into logical categories. Provide appropriate navigation structures for accessing those categories. Your users should be able to build a mental model of your site.

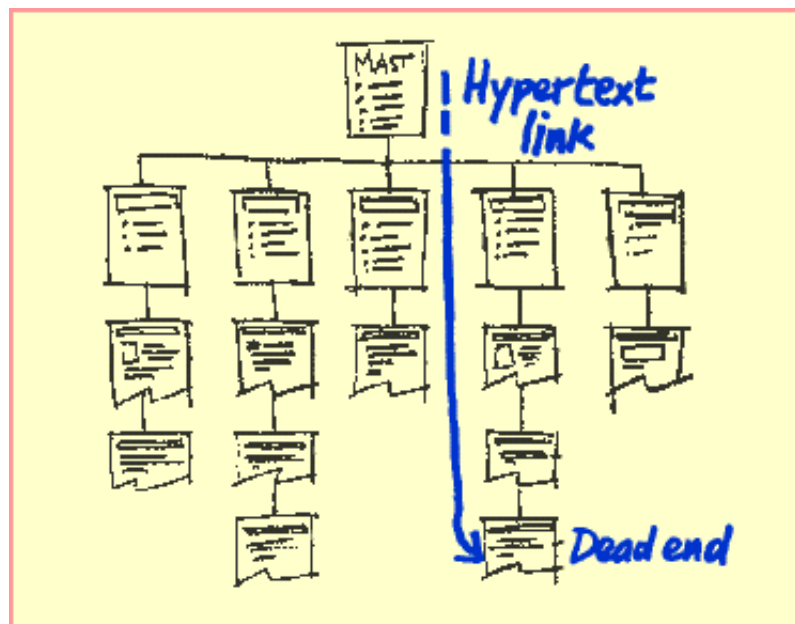
The following examples contrast a structured and unstructured set of web pages.



Provide navigation control on every page. Avoid dead-ends. Provide links for the user to continue through your site in a logical sequence or branch to other sections. Don't rely solely on the navigation controls in the browser.

Provide a consistent set of navigation links on each page. For example, see the links at the bottom of this page.

The following example illustrates a dead-end page. If users entered the site on the page illustrated, they would have no way to link to other pages in the site. The way to avoid a dead-end page is to supply a link to your home page on every page.

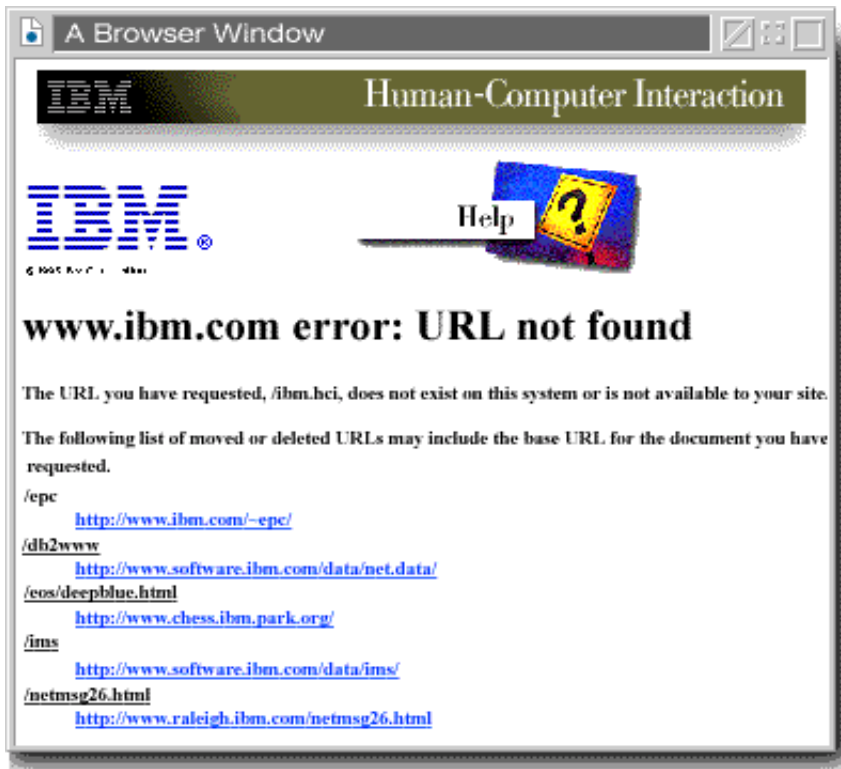


Messages

Provide clear error messages

If an error occurs, tell users what the error is, why it occurred and what they can do to fix it. Communicate in the user's vocabulary. It is a good idea to buy server software which allows you to customize the error messages. Strive to avoid users making errors.

The following example tells the user what the error is and provides alternative links so that the user can recover from the error quickly and easily.

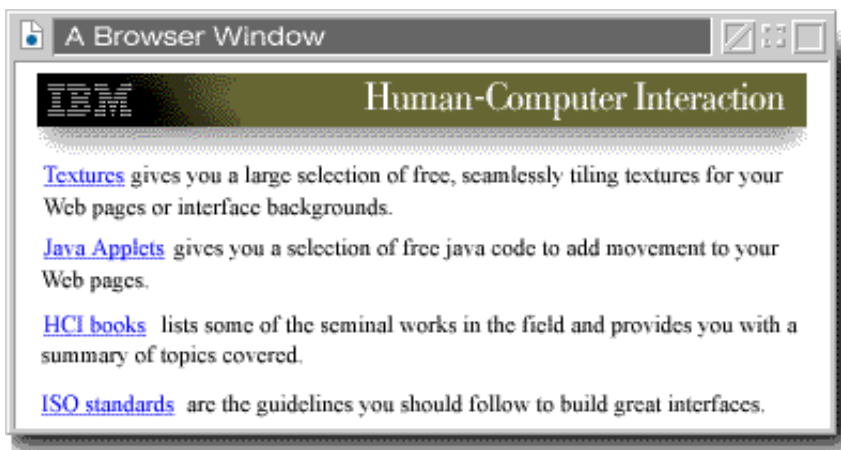


Keep users informed

Users should understand how the interface works, what is currently happening, and what will happen if they perform an action.

Use the surrounding text to tell users where a link goes and what it will show them. Choose meaningful words, short phrases, or graphics to represent the link. Add explanatory text when additional information is desirable. Don't make the reader waste time following links just to find out what they are. For all transfer animations, Java applets, downloads, include file size information.

In the following example, explanatory text is added next to the link where needed.



Tell users when a link will be slow and show progress. Although you can't be sure of the speed of the network, you do know that large transfers are likely to be slow. Users should be told in advance when a link may be slow since this information may affect their decision whether to take a link. Progress indicators should be provided to show the status of lengthy operations such as downloads. Browsers typically provide a progress indicator for file downloads, but not for other transfers such as Java or Shockwave, there you could create your

own.

The following examples show a notice of a file size on a link and a progress indicator.



Using Java applets

Java allows for the creation of applications which are interactive, network aware, and cross-platform.

Use Java where it provides value add

Java can create rich, interactive applications not possible with HTML or other mechanisms. You should be aware that it takes time to load Java applets, and therefore, design accordingly. The following multi-state, animated Java buttons give users helpful feedback by linking visuals to specific button behaviors. Separating the object's behavior (model) from the object's visual representation (view) is a common object-oriented design technique.

The following example shows a set of buttons created in Java that highlight as the pointer passes over them.

Do not use Java when efficient alternatives will do. For example, do not use Java for an animation when an animated GIF would do the same thing faster.

Use Java when performance meets users requirements

Java code must be downloaded before it can run. Depending on the size of the Java code and the speed of the user's connection, there can be a lengthy delay before the applet starts running. Use Java when the tradeoff between load time and value delivered is acceptable.

JavaScript

JavaScript is often a better alternative for comparative Java functions. It can add life and meaning to your site with minimal impact to page performance and programming resources. For instance, JavaScript allows the designer to present user feedback on mouse-over events and can control and direct form input. For example, when Netscape users point to one of the chapter tabs on this notebook, the status area provides a description on the chapter's contents. JavaScript also gave us the function we needed to provide this notebook metaphor. JavaScript enables one click on a link to do several actions: selects a tab, tab pops up, the section button appears, and the section content fills. This action would not be possible with standard HTML.

For more information see:

[Java resources](#) provide Java examples and applets like the animated buttons shown above.

Testing

Usability testing

Designing for the web is really designing for people, so every aspect of the design is also a usability consideration. The time spent doing usability testing and practicing User Centered Design (UCD) can save a project significant resources by allowing you to correct user problems before the general public encounters them. The promise of usability has become a very commonly used selling point for consumer products.

Usability testing can be facilitated by some of the characteristics unique to designing on the web:

- Easy access to users - you can get user input remotely by supplying people with the URL of the site (or page) that needs to be tested.
- The ability to build an automatic test facility - you can program tests into the website by incorporating automated recording mechanisms.
- The speed at which a high fidelity "prototype" can become the final product - last minute improvements to the website are possible based on the last user test.
- Ease of feedback responding doesn't require undue effort on the user's part.

Usability testing should be done in all phases of website development. It is particularly useful early in the design cycle because that is when the big decisions are made and it is easier to make changes in response to user feedback. User data can help you to determine user needs, optimize structure of your site, and develop a usable interface.

Benchmark testing

You should set benchmark design and performance criteria for your website. For example, you may decide that all pages must load within 30 seconds via a 28.8 modem. If you find a page takes longer than this to load, you can then redesign it to decrease load time. You might reduce the number and size of the graphics, or break a long page into multiple, linked pages.

Iterative design

You can't expect to get everything right the first time. Allow time in your schedule to iterate on your design until your site has the qualities you are aiming for. You should continue looking for and fixing problems even after you have published your site. An advantage of the Web over traditional forms of publishing is that changes are relatively easy. It is difficult and expensive to recall a book to fix an error, but very easy to update a website. It is expected of any quality website.

Low-fidelity prototypes

Low-fidelity prototypes are non-interactive representations of your Web pages. They are used to design the presentation of your content. Seeing what your pages will look like helps you as a designer to refine your design and serves as a good communication vehicle. You can show it to other members of the design team and to end users to get feedback early in the design cycle when changes are made most easily.

Paper and pencil or white board techniques are good for constructing low fidelity prototypes. The following example shows a low fidelity prototype of the cover page for our web guidelines.



High fidelity prototypes

High fidelity prototypes capture both the look and the interaction of the website. They can be so realistic that no one can tell them from the final product. You can do your prototyping in HTML or use tools such as HyperCard or Visual Basic. You can choose to make your prototype high fidelity in just a single aspect such as layout or appearance. Paint and image tools or HTML prototypes are good for this.

These guidelines were developed as a high fidelity prototype that evolved into what you see before you.

For more information see:

[The Art of Rapid Prototyping](#) is a book that goes into more detail on prototyping.

Production

The production phase is when you create your pages and implement your website based on your finalized design.

Doing

Now that most of the tough decisions are made, the fun begins. Because the team has spent the up-front time on planning, testing, and designing, they can now work in unison on the production.

Define the site file structure

Considering your web server's conventions, define a file structure for your site. To help your team manage content and generate relative links, match the directory structure to your design structure and use a lowercase filenaming convention. There are several reasons for carefully defining the file structure:

- Sites grow and evolve quickly. Make the structure easily extensible.
- Scripting and file maintenance. A sensible design and a consistent naming scheme makes it easier to find, maintain, and build files. The careful naming scheme used for these pages enabled us to easily implement JavaScript for building multiple links.
- Some servers are case-sensitive. Problems often occur when the production environment is different than the final environment. For example, on a case-sensitive server, home.html is not the same as Home.html, and the result will be a broken link. Make all files lowercase unless a program, such as a Java applet, specifies otherwise.
- Relative links are more flexible than absolute links within a site. A relative link (images/example.gif) is easier to create and move than an absolute link (<http://www.ibm.com/ibm/hci/guidelines/web/images/example.gif>). If you want to stage your files before moving them onto the web server, use relative links.

Generate page templates

To ensure consistency throughout your site, generate page templates for your development environment. Your templates should include any conventions dictated by your company or organization, and they should exemplify your design decisions. For example, users can quickly recognize this as an IBM site by the consistent use of mastheads and footers on every page. Other aids for automating content generation based on a design include; style sheets, server side includes, SGML DTDs, and database templates.

Generate content

Content can vary from text and graphics to multimedia presentations and intricate programs. Each type of content requires different skills and different tools. Find the tools that work best for the individual, the task, and the site. For intensely graphical sites, you might invest in graphic and multimedia programs, photo disks, clip art, and image compression tools. For larger sites, consider a database program like Domino to generate dynamic pages and visual builders to create web applications. Simpler sites can be created in an ASCII or WYSIWYG editor.

Many web pages are created without ever directly editing the HTML; however, working directly with code can provide more control than a WYSIWYG editor. This is especially true when working with the very latest HTML extensions such as LAYERS. The right tool can save time and allow focus to be directed towards the really important work, so consider trying some of the tool downloads available on the web today. These limited offers allow you to try a product before committing to a purchase.

Do the right thing

As you create your pages, don't overlook the little things that make your site easier to use and find. Include the following HTML elements and attributes:

TITLE

Create a short, descriptive title for your page. TITLE is used by search engines and appears in the search output as well as in the browser's title bar, history, and bookmark list.

META NAME="keywords" CONTENT=" "

Enter index terms for your page within the CONTENT quotes. Terms can be used by search engines to help users find your page.

META NAME="description" CONTENT=" "

Enter an abstract of your page within the CONTENT quotes. Some search engines search the text and display it in the search output.

ALT attribute for IMG SRC

Enter a descriptive title for the graphic. Alternative text explains graphics for those using screen readers, text only browsers, and browsers with image loading set to off. This is especially important for graphics that serve as links.

WIDTH and HEIGHT attributes for IMG SRC

Enter the correct image size. Sizings can speed page download time and set the page format before all the content loads. The LOWSRC attribute can also enhance image download time.

Stage the site

Throughout content generation, move the files onto a shared web server that is protected from public view. This enables the team to iteratively test their pages and track their progress. As pages and sections of the site develop, spell check, edit, test, and review. Be sure to stay true to your design decisions and continue to check against your user requirements and competitive benchmarks.

Testing

Test in the client environment

Web pages can look and behave very differently depending on the browser, operating system, screen resolution, and internet connection. During the proposal phase, you determined the client environment and you designed accordingly. Now is the time to verify that you can access and use your pages in that environment. If you based your design on a future browser release, this is especially important as function can drop out before a final release. Some of the problem areas to check:

- JavaScript, Visual Basic Script, ActiveX, and Java applets. Results can vary based on the browser and the operating system. Use comments to hide script language from older browsers.
- Page layout. Browsers interpret and present HTML differently.
- HTML extensions. Extensions to HTML3.2 are browser and version specific.
- Horizontal scroll bars. There is nothing more annoying than having to scroll, especially horizontally. Test your pages in the resolution you designed for.
- Colors. The appearance of colors can vary based on resolution, color depth, and video drivers.

There are several good HTML validators and link checkers on the web; some even evaluate the download time for a page given different connection rates. However, there is nothing better than a hands-on review of all your pages in the defined client environment.

Usability verification

There are some unique considerations for usability on the web. Some things to address when verifying your site:

- Can users distinguish between static graphics and graphical links?
- Are users able to see the most important information without scrolling?
- Can users navigate easily within your site without getting lost?
- Can users find the information they are looking for easily?
- Do the user's reactions to your site support your goals?

Quality assurance testing

Before you publish your site to the web, ensure that:

- Content is technically accurate, spelled correctly, and well written.
- TITLE and META elements are complete and accurate.
- ALT, WIDTH and HEIGHT attributes are set for all graphics.
- Links go to the correct location.
- Pages, forms, graphics, applets, scripts, and plugins display and behave correctly.

- Feedback and form scripts work. Is the script sending the data to the specified person, and in a usable format? If you are using an automatic feedback response system, is it working?
- Image maps function correctly for both server and client-side maps, the boundaries are defined correctly, the ALT text is accurate, and an alternate list of links is provided.
- The file size is specified for all downloadable files.

Backup plan

Keep a safe copy of your data. For example, a hard drive crash, a hacker attack, or a virus, can wipe out all your hard work.

Publish the site

Once you have verified that your site meets all your stated requirements, publish all the files to the web server and announce the site. If publishing requires moving the site from another server, you should test your pages one more time on the new server.

Maintenance

Maintenance is the continuing process of keeping your website running and up-to-date.

Advertising

Reaching your audience

Usually users do not just happen across your site unless you have communicated the universal resource locator (URL) to them. There are many ways to get your site indexed so that if a user does a search on a particular topic, your site will be included in the search results.

Place keywords in the meta-information. Search engines can use this to identify the topics of your page even when those specific words don't appear in the text.

Other ways to advertise your website are:

- Announce your site on relevant news groups
- Advertise on major websites like Yahoo and Netscape
- Advertise via other media like magazines, radio, and television

For more information:

www.submit-it.com is a service for indexing your web pages.

Evolving

Planning for evolution

Your site should be designed with extensibility in mind. Once the site is published and you begin receiving feedback, ideas for new content become apparent. Can the new content fit easily into the overall design without causing a tidal wave of changes to the visuals and content?

How often will you update your site? Updates keep users interested so update as frequently as you like. However, give careful consideration to major revisions, like a change in visual style. If you designed for model/view separation, the visual style can be changed while leaving the web structure intact and not disrupting anyone who has bookmarked your site. Otherwise, major revisions take time and resource so when planning a major revision you should revisit the process defined in these guidelines.

When you update your site, it is helpful to indicate what is new to users. For instance, if you allocate a certain amount of space on your home page for a feature or news section, users can see what is new right away and don't have to search through the whole site trying to determine if it has been updated and what has been added. You should date and/or version both the site and specific contents. Remove or preferably archive information that is no longer current.

Keeping it alive

There are several things to consider in keeping your website alive:

- Frequent updates - keep users interested.
- Link maintenance - dead links have a negative connotation with users.
- Respond to feedback - respond quickly to the feedback you received. If the feedback is in the form of a request and you cannot answer the request immediately, send a message letting users know you received their feedback and that you will be getting back to them soon.

Tracking

Tracking usage

Usage information helps you determine if you are reaching people as you intended. Hit rates tell you how many people are visiting and how often. The addresses where requests originate combined with user comments will help you judge whether you are reaching the intended audience(s).

Other forms of usage tracking should also be considered. For example, produce registration questionnaires that include a question like "where did you learn about this product?" and find out which websites users visited before finding yours by asking them to provide the website address they were at when they discovered yours.

Responding to feedback

User feedback is valuable to you. Be responsive to your readers when they communicate with you. At a minimum, acknowledge that you have received their communication. Better yet, tell them what you are going to do based on their feedback.

Control

Version control

You must know which version of your web information is the most current. You may want to keep previous versions if the information is still useful or you would like to have a history of how the information has evolved. You may want to maintain alternative versions of the information. A method for version control becomes very important when you are maintaining intranet and internet sites.

One method for version control is placing text at the bottom of your HTML pages stating the date the page was created. This text is viewable on the web and serves as a reminder to you and others that the page is new or might be ready for updating. The following is an example of the tagging to use:

This document generated at Thu, 23 Jan 1997 19:26:41 on
(place reference to website here).
[[HCI Home](#) | [Exhibits](#) | [Guidelines](#) | [Resources](#) | [UCD](#) | [Feedback](#)]
[[IBM Home](#) | [Order](#) | [Search](#) | [Contact IBM](#) | [Legal](#)]