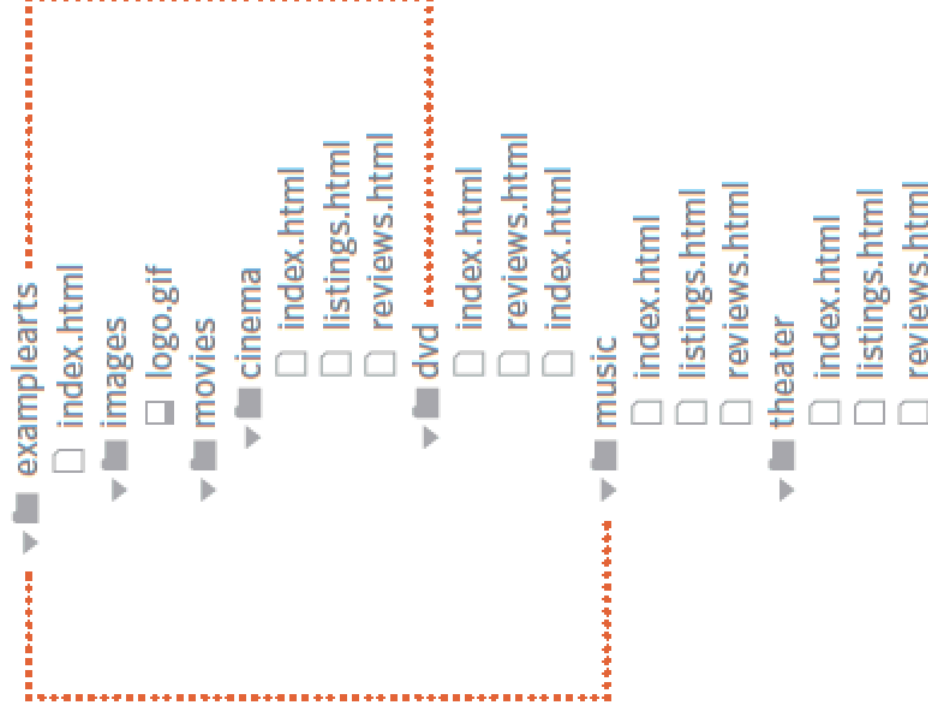


On larger websites it's a good idea to organize your code by placing the pages for each different section of the site into a new folder. Folders on a website are sometimes referred to as directories.

STRUCTURE	RELATIONSHIPS	HOME PAGES
<p>The diagram on the right shows the directory structure for a fictional entertainment listings website called ExampleArts.</p> <p>The top-level folder is known as the root folder. (In this example, the root folder is called examplearts.) The root folder contains all of the other files and folders for a website.</p> <p>Each section of the site is placed in a separate folder; this helps organize the files.</p>	<p>The relationship between files and folders on a website is described using the same terminology as a family tree.</p> <p>In the diagram on the right, you can see some relationships have been drawn in.</p> <p>The examplearts folder is a parent of the movies, music and theater folders. And the movies, music and theater folders are children of the examplearts folder.</p> <p>Instead, these systems often use one template file for each different type of page (such as news articles, blog posts, or products).</p>	<p>The main homepage of a site written in HTML (and the homepages of each section in a child folder) is called index.html.</p> <p>Web servers are usually set up to return the index.html file if no file name is specified.</p> <p>Therefore, if you enter <code>examplearts.com</code> it will return <code>examplearts.com/index.html</code>, and <code>examplearts.com/music</code> will return <code>examplearts.com/music/index.html</code>.</p> <p>Editing the template file would change all of the pages that use that template. Do not change any code that is not HTML or you may break the page.</p>

PARENT

The *examplearts* folder is a parent of the *music* folder.



CHILD

The *music* folder is a child of the *examplearts* folder.

GRANDPARENT

The *examplearts* folder is a grandparent of the *dvd* folder.

GRANDCHILD

The *dvd* folder is a grandchild of the *examplearts* folder.

Every page and every image on a website has a **URL** (or Uniform Resource Locator). The URL is made up of the domain name followed by the **path** to that page or image.

The path to the homepage of this site is `www.examplearts.com/index.html`. The path to the logo for the site is `examplearts.com/images/logo.gif`.

You use URLs when linking to other web pages and when including images in your own site. On the next page, you will meet a shorthand way to link to files on your own site.

The root folder contains:

- A file called *index.html* which is the homepage for the entire site
- Individual folders for the movies, music and theatre sections of the site

Each sub-directory contains:

- A file called *index.html* which is the homepage for that section
- A reviews page called *reviews.html*
- A listings page called *listings.html* (except for the DVD section)

The movies section contains:

- A folder called *cinema*
- A folder called *DVD*.

Relative URLs can be used when linking to pages within your own website. They provide a shorthand way of telling the browser where to find your files.

When you are linking to a page on your own website, you do not need to specify the domain name. You can use **relative URLs** which are a shorthand way to tell the browser where a page is in relation to the current page.

This is especially helpful when creating a new website or learning about HTML because you can create links between pages when they are only on your personal computer (before you have got a domain name and uploaded them to the web).

Because you do not need to repeat the domain name in each link, they are also quicker to write.

If all of the files in your site are in one folder, you simply use the file name for that page.

If your site is organized into separate folders (or directories), you need to tell the browser how to get from the page it is *currently on* to the page that you are *linking to*.

If you link to the same page from two different pages you might, therefore, need to write two different relative URLs.

These links make use of the same terminology (borrowed from that of family trees) you met on the previous page which introduces directory structure.

RELATIVE LINK TYPE

EXAMPLE (from diagram on previous page)

SAME FOLDER

To link to a file in the same folder, just use the file name. (Nothing else is needed.)

To link to music reviews from the music homepage:
`Reviews`

CHILD FOLDER

For a child folder, use the name of the child folder, followed by a forward slash, then the file name.

To link to music listings from the homepage:
`Listings`

GRANDCHILD FOLDER

Use the name of the child folder, followed by a forward slash, then the name of the grandchild folder, followed by another forward slash, then the file name.

To link to DVD reviews from the homepage:
`Reviews`

PARENT FOLDER

Use `../` to indicate the folder above the current one, then follow it with the file name.

To link to the homepage from the music reviews:
`Home`

GRANDPARENT FOLDER

Repeat the `../` to indicate that you want to go up two folders (rather than one), then follow it with the file name.

To link to the homepage from the DVD reviews:
`Home`

When a website is live (that is, uploaded to a web server) you may see a couple of other techniques used that do not work when the files are on your local computer.

For example, you may see the name of a child folder without the name of a file. In this case the web server will usually try to show the homepage for that section.

A forward slash will return the homepage for the entire site, and a forward slash followed by a file name will return that file providing it is in the root directory.