Web Pages for Studying Language and Culture

William J. Poser
Yinka Dene Language Institute and the
University of British Columbia

Yinka Dene Language Institute
Technical Report #3

Yinka Dene Language Institute
RR#1, Site 12, Comp. 41
Vanderhoof, BC V0J 3A0
Canada

1. Introduction

The web pages with which most people are now familiar have two properties that are very useful for studying language and culture: (a) multimedia capability and (b) interactivity. They are also now very widespread. Virtually all schools have computers, and many families now have them too. Here I would like to discuss some simple but useful applications of this technology and to explain how even a novice can go about it.

We increasingly see First Nations communities paying large amounts of money to produce materials about their language and culture. Some of these materials are very nice and have features that require more than simple web page construction by non-programmers using the HTML markup language. However, a surprising amount can be done using nothing but HTML by people with limited computer skills. Doing this where possible has a number of advantages:

- Much of the special purpose software used by commercial outfits runs only on one operating system. Pure HTML web pages are *platform-independent*; material that you write for a Windows system can be used on a Macintosh or Linux system.
- The material remains under local control.
- The material can be modified easily and frequently and added to incrementally. One virtue of this is that students can add material as part of school projects. This teaches them about the subject matter and also teaches them useful computer skills.
- Rather than having to follow the template created by other people, often for a language that is quite different, you can design materials that are appropriate for your own language, culture, and needs.
- Costs are lower since the software needed to create web pages costs nothing and the labour can be done at relatively low cost by people in the community.
- Jobs and money are kept in the community.

Materials in the form of web pages are useful even if you do not have access to the Internet; what is crucial is that the *browser* programs, such as Internet Explorer and Netscape, that are used to surf the internet, provide the interface to the material. If you do not have access to the Internet, you can still use web pages by distributing the necessary files to the computers on which you wish to use them by means of floppy disks, Zip disks, or CD-ROMs. As long as you have a suitable computer with the appropriate network software and a browser, it makes no difference whether you access the files locally or over the network.

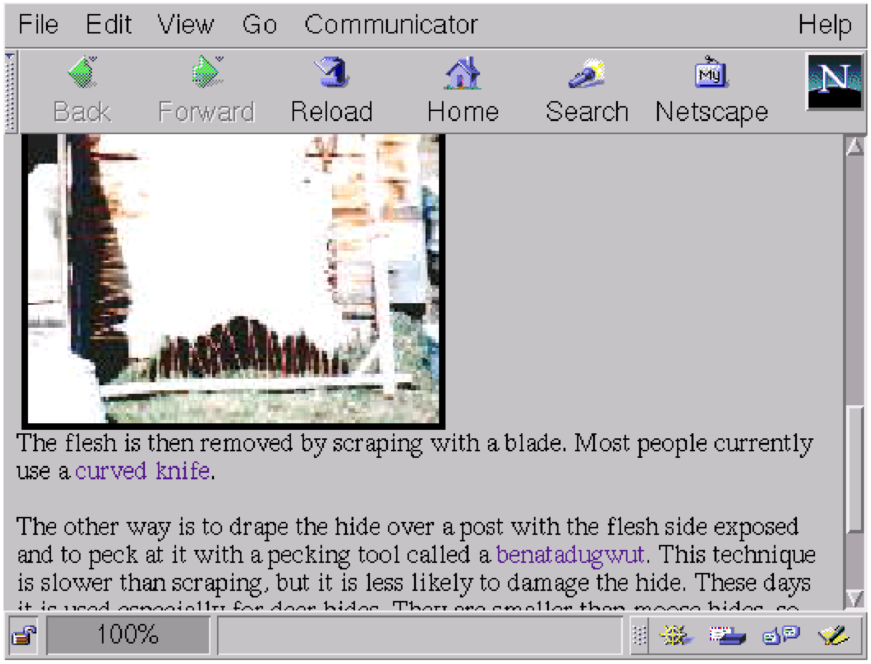
On the other hand, if you do not want your material to be widely distributed, you can still make use of web pages without fear that they will be accessible over the Internet. You can use them on machines that are not connected to the Internet. On machines that are connected to the Internet, you only need to put them in places not accessible to the public. Web pages are useful if you want to make information available over the Internet, but using this technology does not force you to make material publicly available if you do not want to.

2. Some Examples

Let us begin with some concrete examples, a fragment of a cultural encyclopaedia, a picture dictionary, a text with interactive annotations, and a children's storybook. In the following section we will discuss how to create such web pages.

2.1. Encyclopaedic Works

Here is a snapshot of a portion of a web page about tanning skins viewed in the Netscape web browser on a laptop computer running the Linux operating system.[1]



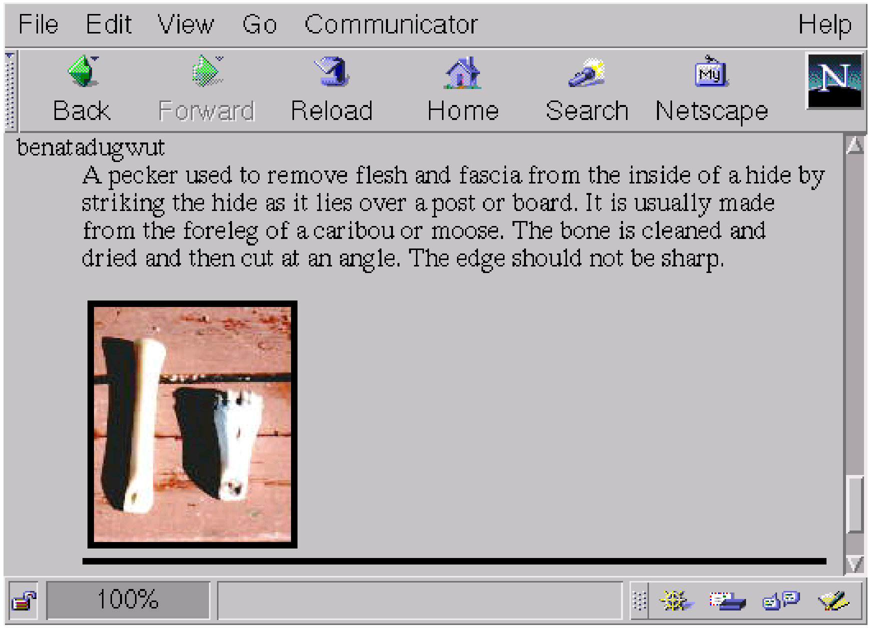
The flesh is then removed by scraping with a blade. Most people currently use a curved knife.

The other way is to drape the hide over a post with the flesh side exposed and to peck at it with a pecking tool called a benatadugwut. This technique is slower than scraping, but it is less likely to damage the hide. These days it is used especially for deer hides. They are smaller than moose hides, so

It consists primarily of text, but contains a picture of a moose hide in a tanning frame. The reader can use the scrollbar at the right side to move around in the text.

[1] The originals of the illustrations in this paper are in colour, as the great majority of computer displays are nowadays. A colour version of this paper is available. However, due to the great cost of colour reproduction, most versions of this paper are not printed in colour.

Several words and phrases, including the Carrier word *benatadugwut* are highlighted. These are links. If the reader selects[2] the word *benatadugwut*, the display changes to a new page displaying information about *benatadugwut*, including a picture of two of them.

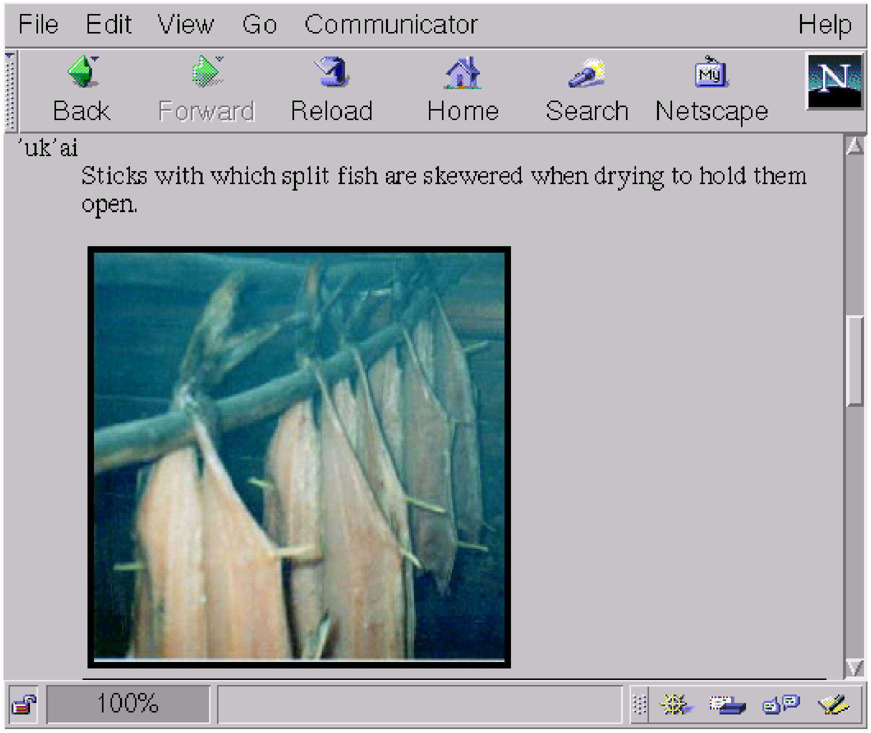


benatadugwut

A pecker used to remove flesh and fascia from the inside of a hide by striking the hide as it lies over a post or board. It is usually made from the foreleg of a caribou or moose. The bone is cleaned and dried and then cut at an angle. The edge should not be sharp.

If the reader is satisifed with this information about *benatadguwut*, he can click on his browser's BACK button, return to the page about skins, and read further. However, the page with the section about *benatadugwut* contains information, often including pictures, about a variety of other tools as well, including other tools used to tan skins, such as scrapers, and tools used for other tasks, such as the poles and skewers used for drying fish.

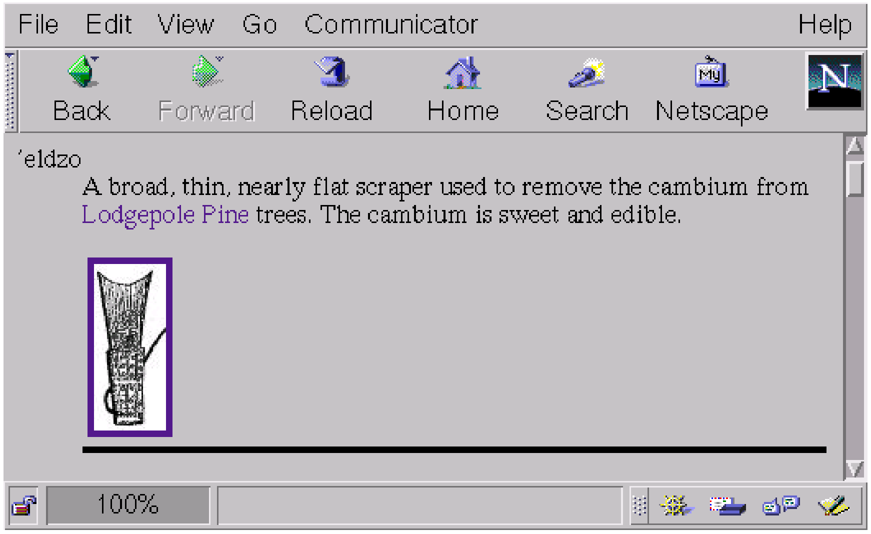
[2] Selecting a link usually means pointing at it with a mouse or other pointing device and clicking a mouse button. For this reason it is often referred to as "clicking on" the link. However, there are other ways of a selecting a link, so it is not entirely accurate to speak of "clicking" on it.



’uk’ai

Sticks with which split fish are skewered when drying to hold them open.

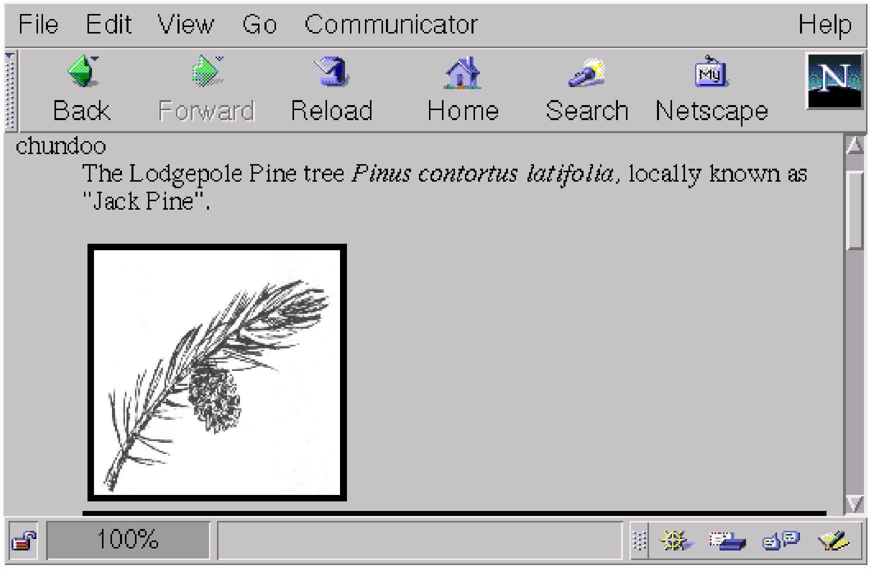
By scrolling around using the browser’s scrollbar, the reader can learn about other tools. One of the other tools described is the cambium scraper. This is described briefly and illustrated by an old drawing.



’eldzo

A broad, thin, nearly flat scraper used to remove the cambium from Lodgepole Pine trees. The cambium is sweet and edible.

The description of the cambium scraper mentions that it is used to scrape cambium from Lodgepole Pine trees. The reader who wants to know more about these trees can select this link, which leads to a page about trees, including a bit of information about the Lodgepole Pine and a drawing of a bough.



chundoo

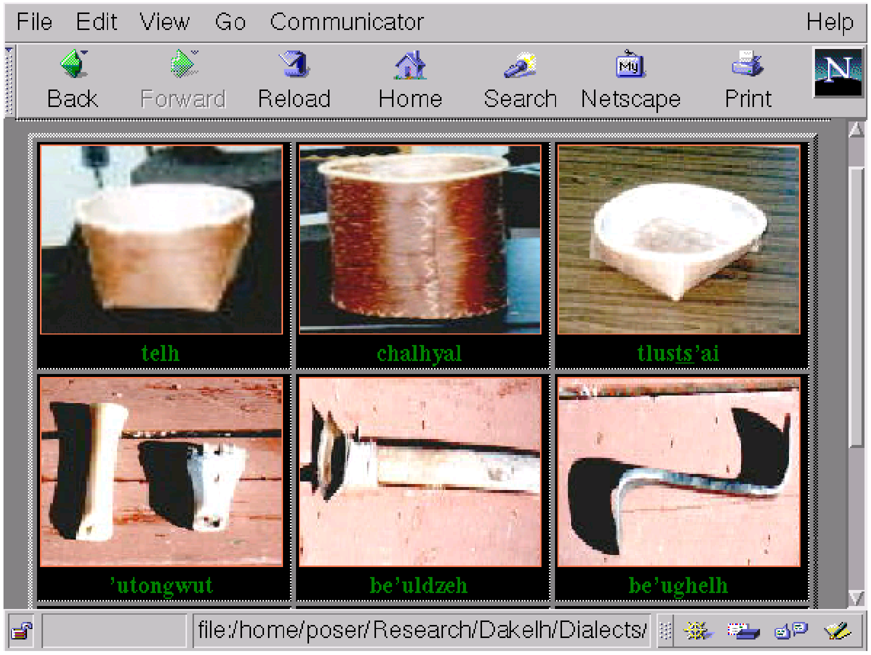
The Lodgepole Pine tree *Pinus contortus latifolia*, locally known as "Jack Pine".

There are several virtues to this approach. First, it is easy to include images, sound files, and other multimedia information. Second, you are not restricted to a linear structure. The user can move around, following links to his or her interests or areas of ignorance, then returning to a previous portion of the text when he has satisified his curiosity. Third, such projects can be created incrementally. Whenever someone brings in some text, a new illustration, or some relevant sound, it can easily be added to the existing material. This makes it ideal for projects involving the collaboration of many people, such as student projects.

2.2. Picture Dictionaries

A common language teaching tool, especially for children, is a picture dictionary, in which pictures, usually arranged by topic, are used to illustrate words. Such dictionaries are useful even for adults since they not only make the objects more memorable but they can help to make clear what a word refers to when the object is unfamiliar. They can also help to present contrasts with which the learner may not be very familiar. For example, some people do not know the difference between a black bear and a grizzly bear, for which, in some languages, such as Carrier, there are entirely different words. Presenting pictures of the two side-by-side can help to bring out the contrast.

One way to organize a picture dictionary is to start with an index page that lists categories of things. Selecting a category brings up a page of things of that type. Here is a page in the Nak'albun/Dzinghubun dialect of Carrier for tools and related objects.



file:/home/poser/Research/Dakelh/Dialects/

Although it is not obvious from the illustration, selecting one of the pictures triggers the playback of an audio file containing the Carrier word. Such a dictionary may also have an English index so that the user can easily get to the page containing the item he or she is interested in.

2.3. Annotated Texts

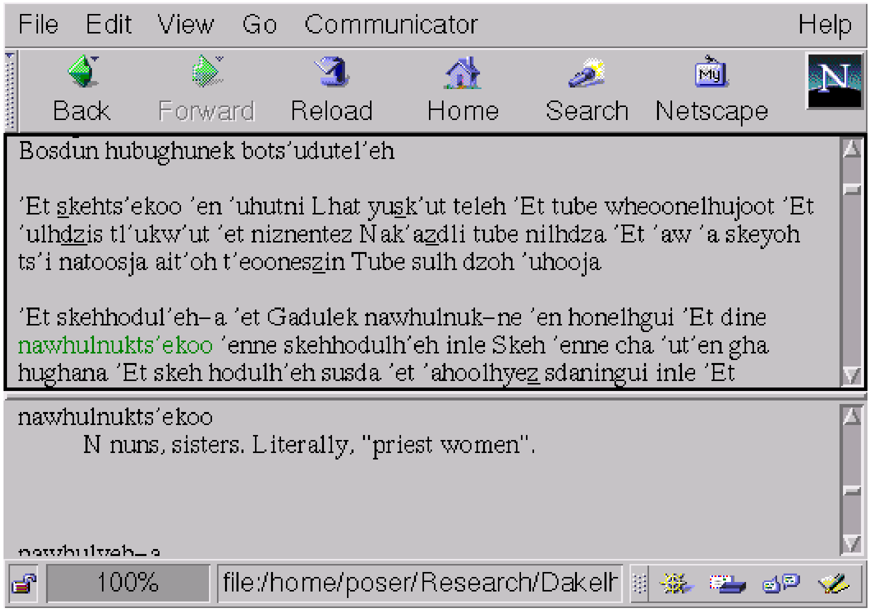
One very useful application of web pages is as a cheap, easy, and platform-independant way of presenting annotated texts. This allows a language learner to read through a text and to obtain information about it, say the meaning of a unfamiliar word, quickly and easily. This is especially useful for intermediate to advanced level language learners. You can buy commercial programs that do this, but the ones with which I am familiar require you to use their material. There is no way to prepare additional texts or to modify what they provide.

In fact, there is a simple way to do this yourself by taking advantage of the technology that underlies the World Wide Web. Not only does this approach allow you to produce exactly the materials that you want, but it saves money, since the software that you need is free.

A third, and major, virtue of this approach is that it is platform-independent. With a modicum of care, the same files will work on any computer that has a web browser, such as Netscape or Internet Explorer. That means that what you do can be used on PCs running the various versions of Windows, on Macintosh machines, and under the various versions of UNIX, including LINUX.[3]

[3] This is of course subject to a few simple caveats. For example, the filenames used must be possible on the machine in use. It is therefore a good idea to use three letter extensions for

Here is an example of a web page of this type viewed in Netscape Communicator on a Linux system. This is from an account of life in a residential school told by Dr. Mary John, Sr., OAC, in the S̲aik'uz̲ (Stoney Creek) dialect of Carrier, an Athabaskan language of the central interior of British Columbia. There is a single window containing two frames. Notice how the lower frame shows the definition of the word *nawhulnukts'ekoo*, the first word in the second line of the third paragraph. The user had clicked on this word just before the snapshot was taken, as we can see from the highlighting indicating that the link has been visited.



Bosdun hubughunek bots'udutel'eh

'Et s̲kehts'ekoo 'en 'uhutni Lhat yus̲k'ut teleh 'Et tube wheoonelhujoot 'Et 'ulhdz̲is tl'ukw'ut 'et niznentez Nak'az̲dli tube nilhdza 'Et 'aw 'a skeyoh ts'i natoosja ait'oh t'eooneszin Tube sulh dzoh 'uhooja

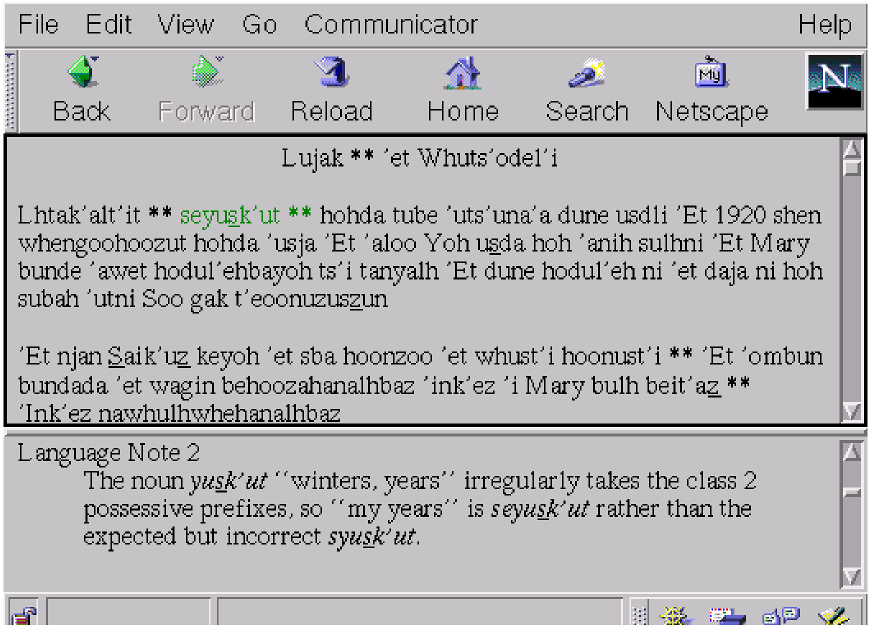
'Et skehhodul'eh–a 'et Gadulek nawhulnuk–ne 'en honelhgui 'Et dine nawhulnukts'ekoo 'enne skehhodulh'eh inle Skeh 'enne cha 'ut'en gha hughana 'Et skeh hodulh'eh susda 'et 'ahoolhyez̲ sdaningui inle 'Et

nawhulnukts'ekoo
N nuns, sisters. Literally, "priest women".

This allows the reader to obtain the definition of unfamiliar words immediately, without having to page through a dictionary.

The same technique can be used to provide grammatical and historical notes. Here is another snapshot of the same text. In this case the user has just selected the double asterisk after the word *seyus̲k'ut* in the upper frame. Again, we can see that the link has been visited because it is highlighted. This has brought up the grammatical note in the lower frame.

filenames, e.g. *htm* rather than the full *html*, so as to comply with the DOS limitation of extensions to this length.



Lujak ** ’et Whuts’odel’i

Lhtak’alt’it ** seyus̲k’ut ** hohda tube ’uts’una’a dune usdli ’Et 1920 shen whengoohoozut hohda ’usja ’Et ’aloo Yoh us̲da hoh ’anih sulhni ’Et Mary bunde ’awet hodul’ehbayoh ts’i tanyalh ’Et dune hodul’eh ni ’et daja ni hoh subah ’utni Soo gak t’eoonuzusz̲un

’Et njan S̲aik’uz̲ keyoh ’et sba hoonzoo ’et whust’i hoonust’i ** ’Et ’ombun bundada ’et wagin behoozahanalhbaz ’ink’ez ’i Mary bulh beit’az̲ ** ’Ink’ez nawhulhwhehanalhbaz

Language Note 2

The noun *yus̲k’ut* ‘‘winters, years’’ irregularly takes the class 2 possessive prefixes, so ‘‘my years’’ is *seyus̲k’ut* rather than the expected but incorrect *syus̲k’ut*.

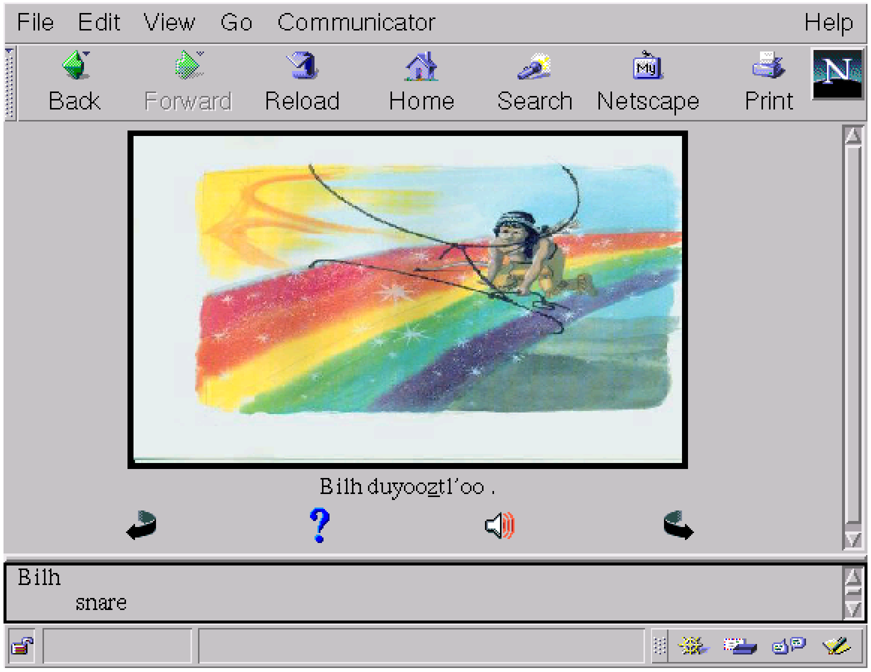
This kind of information is very difficult to obtain other than from an annotated text. The problem with grammars is that you have to know a lot about a topic in order to know where to look it up. This approach allows a teacher to include notes on whatever points she thinks are likely to be of interest to her intended audience. It allows the reader to decide when to look at the grammatical notes and, when she chooses to look at them, to see them immediately. An additional possibility is to add links to free translations of each sentence or paragraph.

Yet another virtue of this approach is that it avoids repetition. Often the same point calling for comment occurs several times in the text. In a printed annotated text, the note must be printed repeatedly if it is to appear on the same page as the text, so that the reader can easily look at it. A web page design like this allows the same note to be linked many times to different bits of the text.

2.4. Story Books

A similar approach can be taken with story books for children. One difference here is that we will want to have lots of pictures. Another is that we will not want to have the reader scroll around in a page of text but rather will want to simulate turning the pages of a printed book by displaying one page at a time.

Here is a sample page from a web page version of the story *The Boy Who Snared the Sun* by Catherine Bird, published by the Yinka Dene Language Institute.

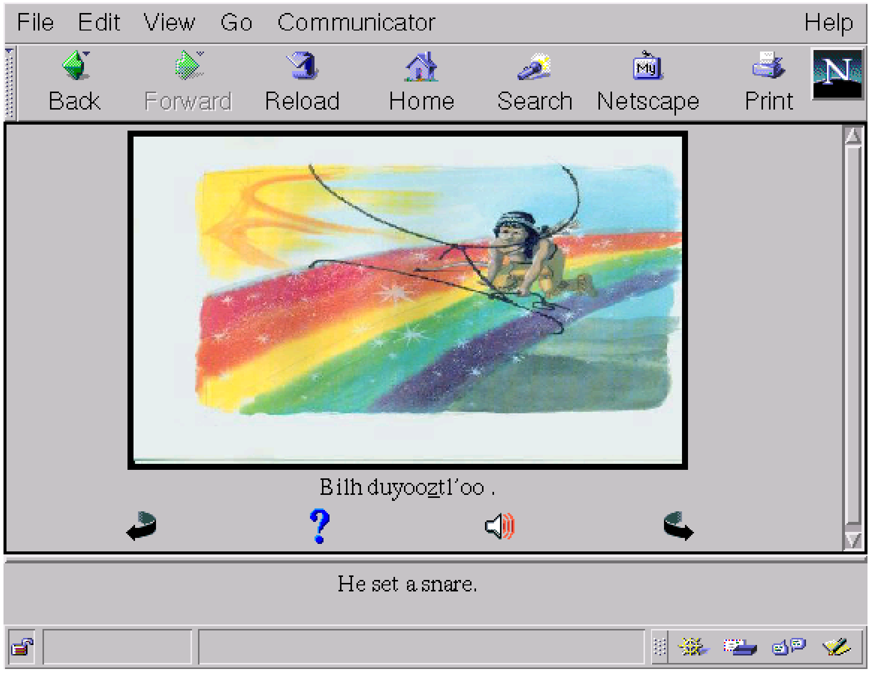


Bilh duyooẕtl’oo .

Bilh
snare

The page consists of a large colour illustration with a single line of Carrier text beneath it. As before, clicking on a word causes a definition to be displayed in the lower frame. We see that in the example the user has selected the word *Bilh*.

Below the text are four icons, each a link. The first one turns the page back. The last turns the page forward. The second link causes the English translation of the text to appear in the lower frame, as here:



Bilh duyooẕtl’oo .

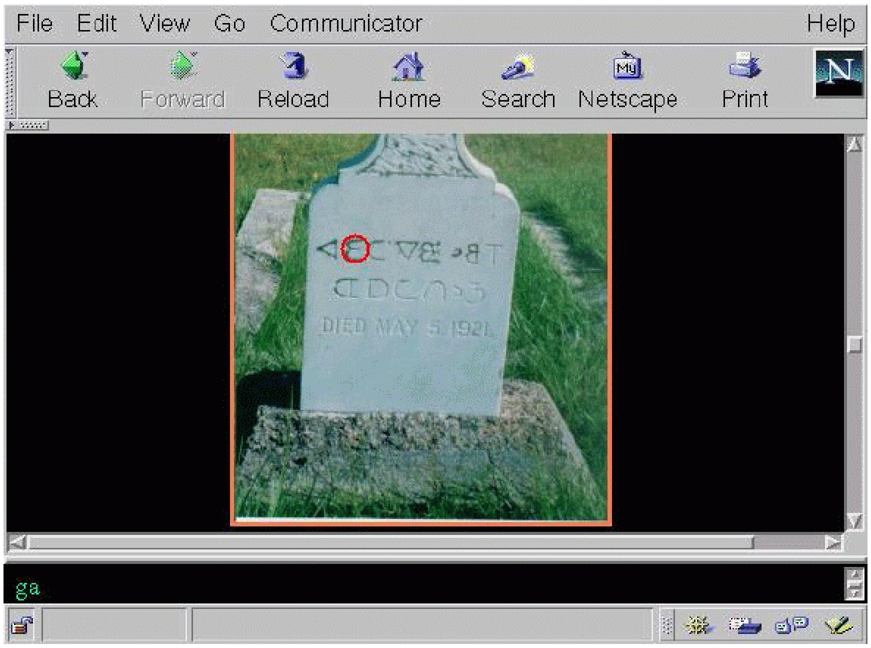
He set a snare.

The third link plays an audio file so that the user can hear the written material.

Creating stories as web pages has several virtues. One is the ability to include audio. A second is the ability to provide both definitions and translations without displaying them up front. The reader is encouraged to first try to understand the Carrier and only to get help if necessary. Third, it is easy to make different versions of the web page for different dialects. There is a limit to the number of dialects that can be comfortably accomodated on a printed page, and putting several versions on the same page can be confusing for the students.

2.5. Annotated Inscriptions

Here is a screenshot showing a gravestone with an inscription in the Carrier language in the so-called syllabic writing system. The user has just clicked on the second character in the first line, circled in red.[4] In response, ga, the equivalent of this character in the Carrier Linguistic Commitee writing system, the dominant writing system in use at present, has been displayed in the lower frame.



ga

The same approach can be used with maps. Selecting a point on a map could trigger zooming in on that point or the display of the name of the place or other information. Similarly, parts of a scene could be labelled. For example, in a photograph of fish hanging in a smokehouse, selecting a skewer would display the word for ”skewer”, selecting a drying rod displays the word for ”drying rod”, and selecting the fish display the word for ”fish”.

[4] The red circle is not part of the image in the actual web page. It has been added to the illustration in this paper for clarity.

3. Creating Web Pages

Web pages are simply text files written in a special format using Hypertext Markup Language or HTML. They may contain references to other files containing images, sound, video clips, and so forth. You can write web pages using any word processor or text editor. Some have special tools to make it easier to write HTML, but you can always insert the HTML codes that you require by hand.

Contrary to popular mythology, HTML is not a computer programming language, and writing HTML does not require the skills of a computer programmer. Some kinds of web pages do require knowledge of computer programming (such as those that interact with databases and allow you to make credit card purchases) but HTML itself requires no knowledge of computer programming. Both of the examples discussed above use only HTML. No programming skill was required to create them.[5]

If you wish to become well acquainted with HTML, there are many manuals and textbooks available. However, it is so simple to do useful work that you should be able to create web pages of the types described here using only the information in this paper.

3.1. Some HTML Basics

HTML is a set of commands that allow you to mark up a text, indicating how portions of the text are to be displayed and what action should be taken in response to selection of links. Most HTML commands apply to a certain range of text. They therefore having a beginning and an end. The convention is that the tag of the end is the same as that of the beginning with a slash (/) prepended. Thus, <I> begins a region to be *italicized* and </I> ends it. Similarly, <B> begins a region to be **emboldened** and </B> ends it, and <u> begins a region to be <u>underlined</u> and </u> ends it. To center some text, precede it with <center> and follow it with </center>.

A file should be identified as an HTML file. At the very beginning of each file, put the tag <HTML>, and at the very end put the closing tag </HTML>. A new paragraph is begun by the tag <P>. In this case, the closing tag </P> may be omitted if you wish to.

3.2. Including Images

A tag may contain specifications of attributes. For example, to cause an image to be displayed, you use the <IMG> tag. One of the attributes of this tag is the name of the file containing the image.

[5] Automating the creation of web pages, e.g. by converting text in a simpler format to HTML, may involve actual computer programming.

For example, here is the relevant portion of the text about skins above that causes the picture of the stretching frame to appear:

```
<P>
<IMG SRC="StretchingFrame.jpg" border=4 align=left width=250 height=250>
</P>
<br clear=all>
```

The photograph of the stretching frame is included by the <IMG> tag. The SRC attribute tells the browser that the image is in the file *StretchingFrame.jpg*.[6] The *border* attribute specifies a border four pixels thick. *align* says to align the image at the left edge of the page. Finally, *width* and *height* specify that the image should be resized to fit into a 250 pixel by 250 pixel space.[7]

3.3. Links

The crucial thing that makes HTML a hypertext markup system is the *link*, which, when selected, moves us to another piece of text or causes something else to be displayed. A region of text is marked as a link by the paired delimiters <a> and </a>. What should be done when the link is selected is determined by the attributes specified in the opening tag <a>.

For example, here is the text of the web page about skins above that contains the link to the page describing the tool *benatadugwut*:

```
<P> The other way is to drape the hide over a post with
the flesh side exposed and to peck at it with a pecking tool called a
<a href="Tools.htm#pecker">benatadugwut</a>.
This technique is slower than scraping, but it is less likely to damage
the hide. These days it is used especially for deer hides. They are
smaller than moose hides, so the extra time does not matter so much,
and they are finer and thinner, so it is important not to nick them.
```

You can see that the link tags surround the word *benatadugwut*. This link contains a single attribute, namely: `href="Tools.htm#pecker"`. The keyword *href* tells us that this link is to some text, which is to be found in the named file. If, as in this case, we want to refer to a particular part of the file, the label of the specified part of the file is given after a cross-hatch (#). The reference `href="Tools.htm#pecker"` tells the

[6] The suffix *jpg* indicates that this file is in the JPEG format, one of several graphics formats that most web browsers can deal with.

[7] The tag <br clear=all> is a "break". It causes a clean break between the image and the following text. There are other ways of handling this if you want the text to flow around the image.

browser to look in the file named *Tools.htm* and within it find the material labelled *pecker*. In this case, this is a piece of text which includes an image.

3.4. Multi-Word Sequences as Links

Almost all of the links that we have shown consist of a single word. This will generally be the case if the information you want to give is lexical, that is, the meanings of words. However, you are free to include more than one word in a link. Just put the material you want within the link between the tags <a> and </a>. There is an example of this above. In the text about skins, a little before the mention of the *benatadugwut* there is a mention of another tool, a *curved knife*. This too is a link, the selection of which leads to another section of the same page dealing with tools that contains information about *benatadugwut*.

In linguistic materials, such as an annotated text, you might want to do this in the case of an idiom or other expression extending over more than one word. You might also do this if you are giving grammatical or cultural information about whole sentences or paragraphs.

3.5. Creating Interactive Annotated Texts

Here is how to create interactive annotated texts. For each project, you need to create three or more files: one for the text, one or more for the annotation, and a top-level file to define the layout of the window and put the text and annotation together.

The top level file should look like this. What it does is to define the layout of the screen and tell the browser from what files to obtain the material for the two frames. This is the file whose name is given to the web browser. The browser will then load the other files.

Specifically, it sets up two ”frames” within a window on the computer screen. The smaller, lower frame, named ”defwind”, will occupy 100 vertical pixels and will be filled with text from the file called ”lexfile.htm”. The user will not be allowed to change the size of the definition frame once it is created. The larger one, named ”textwind”, will occupy the remainder of the window and will be filled with text from the file called ”Text.htm”.

```
<HTML>

<HEAD>

<TITLE>Life at Lejac</TITLE>

</HEAD>

<FRAMESET rows="*,100">

    <FRAME src="Text.htm" name="textwind">

    <FRAME src="lexfile.htm" name="defwind" NORESIZE>
```

```
</FRAMESET>
</HTML>
```

You will of course substitute your own title for ”Life at Lejac”, and choose your own file names and and frame names.

Here is an excerpt from the text file that underlies Mary John’s story above.

```
<HTML>
<HEAD>
<BASE target="defwind">
<TITLE>Life at Lejac</TITLE>
</HEAD>
<BODY VLINK=green LINK=black>
<center><a href="lexfile.htm#Lujak">Lujak</a>
<a href="notes.htm#hist1"><B>**</B></a>
<a href="lexfile.htm#Qet">'et</a>
<a href="lexfile.htm#WhutsQodelQi">Whuts'odel'i</a></center>
<P>
<a href="lexfile.htm#LhtakQaltQit">Lhtak'alt'it</a>
<a href="notes.htm#gr1"><B>**</B></a>
<a href="lexfile.htm#seyu_skQut">seyu<u>s</u>k'ut</a>
<a href="notes.htm#gr2"><B>**</B></a>
<a href="lexfile.htm#hohda">hohda</a>
<a href="lexfile.htm#tube">tube</a>
<a href="lexfile.htm#QutsQunaQa">'uts'una'a</a>
<a href="lexfile.htm#dune">dune</a>
<a href="lexfile.htm#usdli">usdli</a>
...
<a href="lexfile.htm#QEt">'Et</a>
<a href="lexfile.htm#dine">dine</a>
<a href="lexfile.htm#nawhulnuktsQekoo">nawhulnukts'ekoo</a>
<a href="lexfile.htm#Qenne">'enne</a>
<a href="lexfile.htm#skehhodulhQeh">skehhodulh'eh</a>
<a href="lexfile.htm#inle">inle</a>
</BODY>
</HTML>
```

The entire file is marked as HTML. It is divided into two sections, a *head* and a *body*, by the HEAD and BODY tags. The only things in the head section are the title and a BASE tag whose target attribute specifies the frame in which links from this file are to be displayed. This is necessary to prevent the links from simply causing the new file to be displayed in place of the text in the text frame. This should be the same as the name you give in the frameset definition. The BODY tag has two attributes, LINK and VLINK. These specify the colours to be used for links that have not been visited and links that have been visited.

The remainder of the body section contains the text itself. In general, each word is represented, surrounded by a <a>/</a> pair to make it a link.[8] The first three words are surrounded by a <center>/</center> pair so as to center the title.

Each link initiator contains an href attribute naming the file and label within the file to which to go. You can label your annotations as you wish. In the case of a lexicon, it makes sense to use the words themselves as labels, so that is what I have done here. However, HTML does not allow punctuation in labels. If the language you are working with uses punctuation within words, you will have to convert it to something else in your labels. It doesn't matter what you convert it to so long as it is something that HTML allows and that you can do it quickly and consistently. This situation arises in Carrier since the apostrophe is used as part of the spelling of words. Since Carrier does not use the letter Q, I have simply used a *Q* in the labels whenever there is an apostrophe in the Carrier word.

Notice that you are perfectly free to insert material specifically to support additional notes. For example, in the above example every word is a link to its definition but there are also linguistic and historical notes. These are indicated by **. This informs the reader that a note is present and provides something to select in order to follow the link.

The material to which links lead does not have to come from a single file.[9] In the example above, the definitions come from one file (*lexfile.htm*), the linguistic and historical notes from another (*notes.htm*).

Here is an excerpt from *lexfile.htm*:

```
<HTML>
<HEAD>
<TITLE>Glossary</TITLE>
</HEAD>
<BODY>
<P>
<DL>
```

[8] By default most browsers underline links. If you are making every word a link, you probably will not want the entire text to be underlined. (This is also true if the language is one that uses underscores in its writing system.) Each of the browsers has a setting that allows the user to turn off the underlining of links.

[9] You might think that it did because of the *src* field in the top level file that names the file from which the frame is filled. However, this specification only tells the browser what to put in the frame initially. You are not required to stick to this file.

```
    <DT><a name="nawhulnuktsQekoo"> nawhulnukts'ekoo</a></DT>
    <DD>N nuns, sisters. Literally, "priest women".</DD>
    <spacer type=vertical size=50>
    <DT><a name="nawhulyeh-a"> nawhulyeh-a</a></DT>
    <DD>N recreation room, playroom</DD>
    <spacer type=vertical size=50>
    </DL>
    </BODY>
    </HTML>
```

In most respects it should be quite familiar. The guts of it are in the body, which consists of a definition list, delimited by a <DL>/</DL> pair. Each item consists of a `DT` and a `DD`, followed by a `spacer` tag to prevent more than one definition from being in view at once. The label that allows the browser to choose the appropriate portion of the file is the value of the `name` attribute of the `a` tag.

The definition list layout provides a convenient way of arranging this kind of material, but is not necessary. You can lay it out in any way you like. The browser will always display the text starting at the location of the `name` attribute.

The notes are arranged in a similiar fashion. Here is an excerpt from the file containing the linguistic and historical notes.

```
    <HTML>
    <HEAD>
    <TITLE>Linguistic and Historical Notes</TITLE>
    </HEAD>
    <BODY>
    <P>
    <DL>
    <DT><a name="hist1"> Historical Note 1</a></DT>
    <DD>Lejac Residential School, named after Father Lejacq of the Oblates of
Mary Immaculate.</DD>
    <spacer type=vertical size=50>
    <DT><a name="gr1"> Language Note 1</a></DT>
    <DD>The number "six" <I>lhtak'alt'it</I> is in the multiplicative form
because this is required with units of extent of time, such as years, months,
and weeks.</DD>
    <spacer type=vertical size=50>
    <DT><a name="gr2"> Language Note 2</a></DT>
    <DD>The noun <I> yu<u>s</u>k'ut</I> "winters, years"
    irregularly takes the class 2 possessive prefixes, so "my years" is
    <I>seyu<u>s</u>k'ut</I> rather than the expected but
```

```
incorrect <I>syu<u>s</u>k’ut</I>.</DD>
<spacer type=vertical size=50>
<DT><a name="gr3"> Language Note 3</a></DT>
<DD><I>’et whust’i hoonust’i</I> is
literally "I like I live there." Carrier has no verb forms
that are not marked for the person and number of the subject.
There is no equivalent to the English participle <I> living</I> or
to the infinitive <I> to live</I>.</DD>
<spacer type=vertical size=50>
...
</DL>
</BODY>
</HTML>
```

3.6. Specifying Link Targets

When a link is selected, the text or image must be displayed somewhere. This is determined by the `target` attribute. In the case of the annotated text discussed above we specified the target as an attribute of the `BASE` tag. That was convenient because we wanted all of the links in that page to have the same target, namely the lower frame in which definitions are displayed.

The `target` attribute may also be specified for individual links.[10] This allows us to have different links on the same page with different targets. This ability is used in the storybook described above. Each page has four kinds of links.

- a link to the audio file
- links from words to their definitions
- a link to the English translation of the text
- links to the preceding and following pages

The audio file is not displayed. The definitions and English translation are displayed in the lower frame, but when we move forward or backward to a new text page, we want it to be displayed in the main, text frame. The links to the preceding and following pages must therefore have different targets from the links from words to their definitions. We could specify a target for each link, but since there are quite a few links from words and only the two page transition links, the easiest thing is to specify the definition frame as the target in the `BASE` tag for the page, omit the `target` attribute from the word definition links, and specify the text frame as the target in the page transition links.

[10] If a target is given for a particular link, this overrides the target for the `BASE` tag.

By way of example, here is the HTML underlying the text page illustrated above from *The Boy Who Snared the Sun*.[11]

```
<HTML><HEAD>
<BASE target="lexframe">
</HEAD>
<BODY VLINK=black LINK=black>
<DIV ALIGN=CENTER>
<IMG SRC="ssi13.jpg" border=4 align=center width=367 height=220>
<br clear=all><spacer type=vertical size=4>
<a href="SnareSunLex.htm#Bilh">Bilh</a>
<a href="SnareSunLex.htm#duyoo_ztlQoo">duyoo<u>z</u>tl'oo</a>
.
</DIV>
<CENTER>
<A HREF="Page11.htm" target="textframe">k'oh lhehainonle'</A>
<A HREF="nedo12.htm"> nedo k'un'a</A>
<A HREF="p12.wav">boo<U>z</U>olh<U>ts</U>'ai</A>
<A HREF="Page15.htm" target="textframe">nu<u>s</u> lhehainonle'</A>
</CENTER>
</BODY></HTML>
```

3.7. Images as Links

Remember that a link is anything surrounded by link tags. This need not be text. It can be an image. For example, here is a paragraph consisting of an image with a caption. The image is a link. When selected, the browser will display the file *OwlInfo.htm*. This example is taken from a publications catalogue. The image is the cover of a book; the text to which the link leads is a description of the book. The same technique could be used in language teaching materials to play an audio recording or in cultural materials to bring up some text describing what is in the picture.

```
<P>
<a href="OwlInfo.htm"> <IMG SRC="owl300.jpg" border=4 width=165 ></a>
```

[11] This example contains the `DIV` tag, which we have not seen before. This tag may be used to assign attributes to divisions of the text. The attribute applies to whatever follows <DIV> and precedes the next </DIV>. In this case, the attribute specified is alignment. Since some tags do not allow the `align` tag or have limitations on how it may be used, it is often useful to put alignment information into a `DIV` tag.

```
<br clear=all>
<spacer type=vertical size=8>
<b>The Owl Story</b>
</P>
```

You can make use of images as links to bring up text describing the image or to play audio files.

3.8. Image Maps

Instead of making a whole image a link, it is possible to associate different links with different parts of an image. This is done by means of an *image map*, which defines the different regions and associates each region with a URL.[12]

The image itself is handled in essentially the same way as any other image, only the `IMG` tag requires an extra attribute, the `USEMAP` attribute. The value of this attribute is the name of the map that associates regions in the image with links. For example, in the the HTML for the image of the gravestone discussed above is:

```
<IMG SRC=”agadagrv.jpg” border=3 USEMAP=”#agadamap”>
```

The image is to come from the file *agadagrv.jpg*, to have a border three pixels wide, and to use the map named *agadamap* located within the same file.

The key to creating image maps is creating the maps. A map is an object of type `MAP` containing one or more `AREA`s. Each area specifies the type of area, which may be a circle, a polygon, or a rectangle, the coordinates and, if a circle, radius, of the area, and the link to follow if the area is selected. Here is `agadamap`:

```
<MAP name="agadamap">
<AREA shape="circle" coords="62,250,6" href="dulchars.htm#a">
<AREA shape="circle" coords="82,250,6" href="dulchars.htm#ga">
<AREA shape="circle" coords="100,250,5" href="dulchars.htm#da">
<AREA shape="circle" coords="109,250,4" href="dulchars.htm#Q">
<AREA shape="circle" coords="120,250,6" href="dulchars.htm#oo">
<AREA shape="circle" coords="135,250,6" href="dulchars.htm#tsQaerr">
<AREA shape="circle" coords="153,251,4" href="dulchars.htm#n">
<AREA shape="circle" coords="162,252,5" href="dulchars.htm#kQu">
<AREA shape="circle" coords="179,252,6" href="dulchars.htm#t">
```

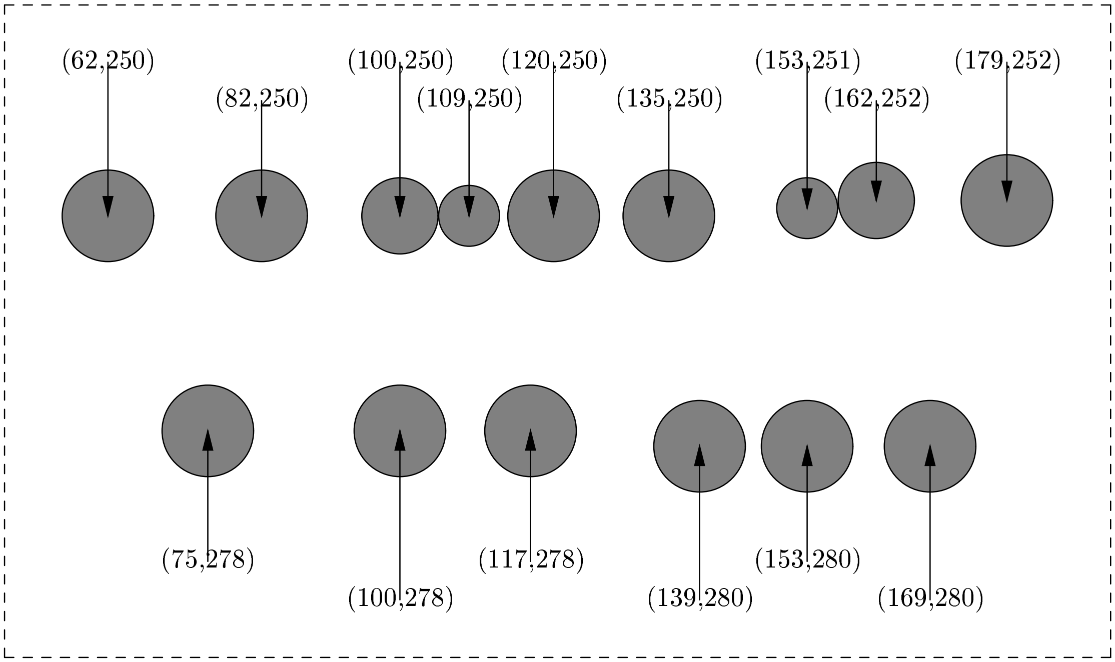
[12] The image maps described here are *client-side image maps*. This means that everything necessary is provided by the browser. This makes it possible to use them on a machine that does not have networking set up and makes it unnecessary to set anything special up on the server. They are also independent of the type of server used. There is another approach to image maps called *server-side image maps*. These are less convenient and require different formats for different servers.

```
<AREA shape="circle" coords="75,278,6" href="dulchars.htm#ba">
<AREA shape="circle" coords="100,278,6" href="dulchars.htm#tuerr">
<AREA shape="circle" coords="117,278,6" href="dulchars.htm#na">
<AREA shape="circle" coords="139,280,6" href="dulchars.htm#do">
<AREA shape="circle" coords="153,280,4" href="dulchars.htm#n">
<AREA shape="circle" coords="169,280,6" href="dulchars.htm#dli">
</MAP>
```

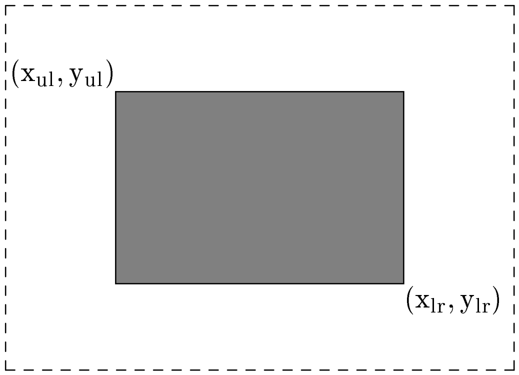
All of the regions defined here are circles. The coordinates are the x and y coordinates of the center of the circle and the radius of the circle.

Coordinates are given in pixels, with positive y downward. That is, the origin (the (0,0) point) is in the upper left hand corner. The first two regions are therefore both six pixels in radius, with centers at the same height twenty pixels apart. The last region is lower than these two because its y coordinate is larger.

Here is a diagram of the circles defined in the image map:

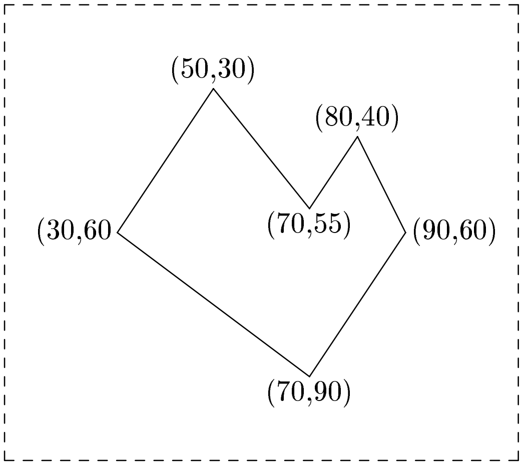


To describe a rectangle, you specify the shape as ”rect” and give the coordinates of the upper left and lower right corners, i.e.: $x_{ul}, y_{ul}, x_{lr}, y_{lr}$.



To describe a polygon, you specify the shape as ”poly” and give the coordinates of the vertices, first the x coordinate for each, then the y, i.e.: $x_1, y_1, x_2, y_2, \ldots, x_n, y_n$. The points must be listed in order, but any point may serve as the starting point.

Consider, for example, this polygon:



The coordinates of this polygon might be specified as: `coords="30,60,50,30,70,55,80,90,90,60,70,90"`.

Note that the regions set up in the image map are not part of the image. They are used only to determine what to do if the user selects a certain point in the image. If you want to represent the same regions graphically, you will have to edit the image yourself to do so.

The most tedious part of creating an image map is determining the coordinates of the regions you wish to use. This can be done manually by displaying the image in a graphics program that gives you a readout of the location of the cursor. There are also special purpose programmes that automate this task. One such programme is MapEdit.

3.9. Playing Sound Files

Sound files should be in one of the standard formats, such as *.WAV* or *.AU*. To play a sound file on following a link, simply name the sound file rather than a text file as the value of the HREF attribute. For example, suppose that you are creating an audio dictionary in which selecting an image causes the name of the object to be played back from a sound file. If you record the word for "owl" in your language and put it in a file named *owl.wav*, the following slight modification of the above HTML would cause this word to be played when the user selected the picture of the owl.

```
<P>
<a href="owl.wav"> <IMG SRC="owl300.jpg" border=4 width=165 ></a>
<br clear=all>
</P>
```

3.10. Exotic Characters

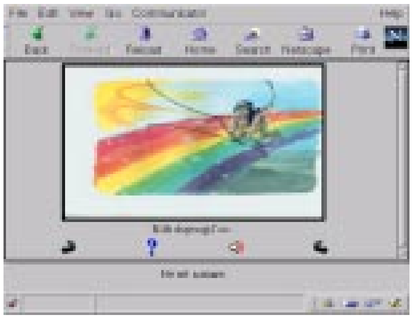
The existing HTML standard only provides for the extended ASCII characters, that is, roughly, the characters usually found on an English-language typewriter together with a variety of characters used in European languages (such as vowels with accents), and a few other things, such as underlining. The non-English characters are generally represented by special names. For example, the letter *e* with an acute accent, *é*, is represented by the code *&eacute;*. A full list of these special symbols and their codes may be found in any HTML manual.

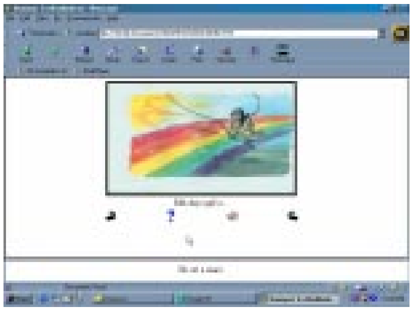
There is at present no easy way to get HTML to make use of special fonts. The next HTML standard is expected to support UNICODE, the international character set standard that will allow for nearly all of the symbols used in the world's languages, but this is not yet available. For the present, there are only two choices. The first is to convert your material into images. The second is to make use of *dynamic fonts*. To use dynamic fonts, the font must be converted into a special format. A few fonts are already available in this format. Software can be purchased with which you can convert your own fonts to the special format. The font is then stored on a special web site. You then add to your web pages information that tells the web browser to obtain the font from the web site on which it is stored. This approach works if you can legally post the converted font on the special web site and if it is acceptable to you to have to access the special web site. A variant is for you to install the font server (the program that provides the fonts from the special web site) on your own web site. You can find out about dynamic fonts from the web site: *http://www.truedoc.com/webpages/ getstart/*. For most people this is a rather complicated approach. If you wish to pursue it, you will need someone with a fair amount of computer expertise to set it up.

4. Platform Dependence

As noted above, a great advantage of using HTML is that it is platform-independent: your web pages will work on a PC running Windows, on a MacIntosh, on a UNIX or LINUX system, or on any other computer that provides a standard web browser. Moreover, any of the major web browsers will do; it does not matter whether the user prefers Netscape or Internet Explorer. However, in designing your web pages you should be aware that their detailed appearance may differ from platform to platform.

Here is an example of how the same files appear on different platforms. The lefthand image is a screenshot made under Linux running the X window system. On this machine, the window system was not set for optimal resolution; the righthand image is a screenshot of the same web page viewed under Windows 98 on the same hardware, with the graphics system set up for maximum resolution.


File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print



In both cases the browser is Netscape, but there are differences in detail in its appearance as implemented on the two different operating systems. Notice how the higher resolution of the display under Windows 98 makes the illustration (whose size is specified in pixels) occupy a smaller portion of the window.

5. Automating Web Page Creation

As you can see, the basic ideas are quite simple. If you want to include only the occasional annotation, such as grammatical notes, you can easily create the necessary files by hand. However, if you are going to make every word a link you will probably want to automate the process.

There are two aspects to automation. First, assuming that you have your text in a file and your annotations in another file, you can easily write a program, or have one written, that will convert them to the appropriate HTML format. Second, if your annotations are definitions, you may wish to generate them automatically from an on-line dictionary rather than writing them by hand.

How hard it is to automate web page creation depends on what you need to do. The initial versions of the text and definition files for Mary John's story were generated by two very simple programs and written in AWK, a language widely used on UNIX systems for text processing. As a rough measure of simplicity, note that one program consisted of 21 lines of code, the other of 19, and that in both cases eight lines are devoted to inserting the HTML codes for underscores, which are used in the Carrier writing system. Another widely used language in which such tasks can easily be automated is PERL. This means that if you wish to automate web page generation, you can learn to do it yourself without learning very much about computer programming. Or, if you prefer to get someone else to do it, just about any programmer should be able to do it for you, and the time and expense will not be very great.

6. Conclusion

Web page technology provides a number of useful opportunities for improved materials for studying language and culture. Interactive, multimedia materials can be prepared easily, with little computer expertise. Materials prepared as web pages can be used by virtually anyone with access to a computer. They are platform independent, so long as a standard web browser is available. Languages whose writing system requires the use of non-ASCII characters are not yet supported by the HTML standard but are expected to be in the near future.

Further Reading

There are many books available on HTML and creating web pages. A useful reference is Stephen Spainour and Valerie Quercia's *Webmaster in a Nutshell: A Desktop Quick Reference*, published by O'Reilly. An easy-to-follow but very comprehensive textbook is Laura Lemay's *Teach Yourself Web Publishing With HTML 4 in 14 Days*, Second Professional Reference Edition, published by Sams.net.

Internet Resources

Examples of some of the ideas discussed here may be found on the Yinka Dene Language Institute web site: *http://www.cnc.bc.ca/yinkadene/ydindex.htm*. Annotated texts are to be found in the section entitled *Dakelh (Carrier) Language*, in the subsection *Annotated Texts*. Image maps are used in the information on the Déné syllabics. To find this, select first *Dakelh (Carrier) Language*, then *Writing Systems*, and within this find the link *Déné syllabics*.

Links to dozens of sites dealing with native languages may be found on the First Nations Languages of British Columbia web site:
http://www.cnc.bc.ca/yinkadene/fnlgsbc.htm. Select the link *Links* near the bottom

of the page. One good site to look at is the Halkomelem Ethnobiology web site: *http://www.sfu.ca/halk-ethnobiology*.

A variety of useful icons, such as the arrows, loudspeaker, and question mark used in the children's story illustrated in section 2.4, may be found in Anthony's Icon Library at: *http://www.cit.gu.edu.au/˜anthony/icons/index.html*. A source of images of all sorts is Barry's Clip Art Server: *http://www.barrysclipart.com/*.