TipWizard 2.0 beta documentation

How to use TipWizard, and a brief introduction/tutorial to JavaScript coding
by Colin Tucker - ctucke02@postoffice.csu.edu.au

Welcome

Hi, and thanks for downloading TipWizard 2.0. I hope that you can find some use for it on your web site!

Using TipWizard

TipWizard is pretty easy to set up and use, but just make sure you have the following:

- Netscape 3.0 (necessary)
- some basic understand of JavaScript (recommended)
- HTML formatting experience (recommended)
- a graphics program (such as Paint Shop Pro) to create custom titles and buttons (recommended)

What is it?

The simplest definition I can give you is that TipWizard is a JavaScript document that creates a HTML display of random tips (quotes, etc.)

How do I use it?

To use TipWizard, you must supply three things:

- your tip data (in HTML)
- the number of tips
- some URL (or 'command line') parameters

The first two items require a modification of the TipWizard script (in the future, I may create an editor that will remove all JavaScripting away from TipWizard). The third item (or items) are specified in the URL for your TipWizard.

To see some examples (you don't need to understand everything just yet), load the included **examples.html** file into Netscape to see the TipWizard in action.

How is TipWizard best implemented?

TipWizard is happiest when it's running in it's own frame, or a partition of a web document - much like what you see on the TipWizard web site.

TipWizard can not currently be embedded in a single document, because TipWizard IS a separate document itself.

A Java applet may be developed eventually that would allow embedding, but currently, you must run TipWizard in it's own frame for correct operation (unless you want to link to it from a document, but it's meant to be on-screen while the document is displayed too).

Refer to your HTML reference for information on setting up frames - but don't worry, they're not too hard to set up either.

Section One - What are these parameter thingys?

TipWizard currently requires 11 parameters to function correctly. These parameters are passed into the script through it's location (or URL - uniform resource locator).

A URL is anything you see in your 'Location:' field in Netscape. Some examples are:

http://www.yahoo.com	The Yahoo web catalog
ftp://archie.au	An FTP site
http://farrer.riv.csu.edu.au/~ctucker/tw	The TipWizard homepage

Each frame in your web site has a source, or SRC. For example, here is the actual HTML code for the index of the TipWizard site:

```
<HTML>
<HEAD>
<TITLE>Welcome to the TipWizard Homepage</TITLE>
</HEAD>
<FRAMESET ROWS="*,130" BORDER=0 FRAMEBORDER=0>
        <FRAME NAME="InfoFrame" SRC="info.html">
        <FRAME NAME="TipFrame"
       SRC="tip-wizard.html?black|white|yellow|FFDD00|DDBB00|true|100%|arial|3|2|n"
              SCROLLING="No">
</FRAMESET>
</HTML>
```

Notice how there are two frames defined? The first frame points to 'info.html', and the second frame points to 'tip-wizard.html' - with a whole lot of other garbage tacked on the end. This garbage is the parameter input that TipWizard needs to function correctly.

What is the parameter syntax?

To access TipWizard, you must specify the name of the TipWizard HTML file:

tip-wizard.html

Followed by a question mark:

tip-wizard.html?

..and then your parameters, each separated by a vertical bar, which looks like this: |

The parameters are:

BGCOLOR | FGCOLOR | LINKCOLOR | ALINKCOLOR | VLINKCOLOR | BORDERS_ON |

TABLE_WIDTH | FONT | FONT_SIZE | TIP_STYLE | TIP_NUMBER

Each parameter must be separated by a | - with no spaces, such as: p1|p2|p3|p4|p5|...|p11 etc.

IMPORTANT -- THE PARAMETERS MUST BE GIVEN IN THE ABOVE ORDER, OTHERWISE

TIP WIZARD WILL NOT WORK, OR WILL RETURN STRANGE RESULTS

How do I use each parameter?
Below, I will define what each parameter does, and how to specify it. Remember, each parameter must be fed into the TipWizard in the order below.

1. BGCOLOR

Specifies the background colour of the TipWizard document, and is directly analogous to the following HTML statement:

```
<BODY BGCOLOR='given color'>
```

The color you specify as a parameter may be a colour keyword, as specified by Netscape, such as black, red, yellow, blue etc. Or, it may be a hexadecimal code representing the RGB value of that color, e.g. FFFFFF (white), 000000 (black), FFAA00 (orange), 005500 (dull green) etc.

IMPORTANT -- DO NOT SPECIFY A HASH (#) WITH A HEX CODE, E.G. #FFFFFF, or #ABC004. CURRENTLY, TIPWIZARD WILL NOT FUNCTION CORRECTLY WITH A HASH IN PLACE.

Examples

```
tip-wizard.html?black|.....
tip-wizard.html?yellow|.....
tip-wizard.html?34FF7A|.....
```

2. FGCOLOR

Same as above, but it sets the text colour (or foreground) of the document.

Examples

```
tip-wizard.html?black|white|.....
tip-wizard.html?white|green|.....
tip-wizard.html?black|FFFF66|.....
```

3. LINKCOLOR

Same again, but sets the colour of text links which you may define in your tip data.

Examples

```
tip-wizard.html?black|white|yellow|.....
tip-wizard.html?000000|FFFFFF|0088EF|.....
```

4. ALINKCOLOR

Sets the colour of active links in your tips, i.e. what colour is displayed as the mouse is pressed on a link.

Examples

tip-wizard.html?black|white|yellow|red|.....
tip-wizard.html?000000|white|00FF00|blue|.....

5. VLINKCOLOR

Sets the colour of visited links in your tips.

Examples

tip-wizard.html?black|white|yellow|red|purple|.....
tip-wizard.html?000000|white|00FF00|blue|FF0000|.....

You can experiment with different values to see which is best for your display, and don't forget to view the document 'colors.html' in this package - it has a list of every colour keyword Netscape recognises.

Also, if you leave one of the colour parameters blank, or all of them, the default Netscape colours will be used, e.g. tip-wizard.html?|||||......... or.......... tip-wizard.html?black|white||||.....

6. BORDERS_ON

TipWizard uses a table to display your tips. This parameter allows you to toggle the border of that table. Possible values are true or false.

Examples

tip-wizard.html?black|white|yellow|red|purple|true|........... or borders are on
tip-wizard.html?black|white|yellow|red|purple|false|........... borders are off

7. TABLE_WIDTH

Again, this applies to the table - it sets it's width on the page, or effectively the width of the TipWizard itself. You can specify the width in pixels, e.g. 100, 239, 400 etc. or you can specify the width as a percentage of the frame, e.g. 100%, 45%, 67%

Examples

tip-wizard.html?black|white|yellow|red|purple|true|100%|..... width 100% of frame
tip-wizard.html?black|white|yellow|red|purple|true|250|..... width 250 pixels
tip-wizard.html?black|white|yellow|red|purple|true|67%|..... width 67% of frame

8. FONT

Specifies which typeface to use for the text, which of course you are assuming that the end-user has as well. Simply specify the name.

Examples

tip-wizard.html?black|white|yellow|red|purple|true|100%|times|...... shows tips with Times
tip-wizard.html?black|white|yellow|red|purple|true|100%|arial|...... shows tips with Arial
tip-wizard.html?black|white|yellow|red|purple|true|100%|verdana|...... shows tips in Verdana

9. FONT_SIZE

Specifies the size of the tip text - note, this is not in point size (like in a word processor). To give you an idea, a value of 2 gives you text around this size, and you can go larger or even smaller. e.g. 5 is around this size.

Examples

tip-wizard.html?black|white|yellow|red|purple|true|100%|times|2|..... body text sized Times
tip-wizard.html?black|white|yellow|red|purple|true|100%|verdana|4|..... a larger Verdana
tip-wizard.html?black|white|yellow|red|purple|true|100%|arial|1|..... pretty small Arial

10. TIP_STYLE

Ahh, here's where the fun starts. You can currently choose one of three different styles for your TipWizard.

Style 1 gives you title and button images that look best on a white background.

Style 2, which is what you see on the web site, gives you title and button images that look best on a black background.

Style c (for custom), allows you to include your own title and buttons.

Actually, there's another style too:

Style off, turns images off, and uses text links instead.

Now, in accessing the images, TipWizard REQUIRES that two directories are present in the same directory as the TipWizard HTML file:

tw-ti &
tw-bt

The tw-ti directory contains GIF images for titles, and the tw-bt directory contains GIF images used for buttons.

IMPORTANT -- THESE DIRECTORIES MUST BE IN THE SAME FOLDER AS TIPWIZARD, AND THEIR NAMES CAN NOT CHANGE, UNLESS YOU MODIFY THE SCRIPT TO ACCESS A NEW LOCATION

Using Titles

Titles are easy to set up - there are normally **3** images in the tw-ti directory:

ti-tw1.gif ti-tw2.gif ti-c.gif

The first is used for whitish backgrounds (style 1), the second is used for blackish backgrounds (style 2), and the third is custom (style c).

You can put any GIF image in place of the ti-c.gif file, and when style c is selected, your new image will be displayed as the title. Every image MUST be a gif.

IMPORTANT -- YOU MUST CALL YOUR CUSTOM TITLE: ti-c.gif OR YOUR IMAGE WILL NOT BE DISPLAYED. REMEMBER, IT MUST BE A GIF.

Using Buttons
Buttons are a little more tricky to set up, but conceptually, the process is no different to titles.

There are normally **12** images in the tw-bt directory:

bt-p1.gif	Style 1 'Previous' button
bt-pd1.gif	Style 1 'Previous' button, disabled
bt-n1.gif	Style 1 'Next' button
bt-nd1.gif	Style 1 'Next' button, disabled
bt-p2.gif	Style 2 'Previous' button
bt-pd2.gif	Style 2 'Previous' button, disabled
bt-n2.gif	Style 2 'Next' button
bt-nd2.gif	Style 2 'Next' button, disabled
bt-pc.gif	Style c 'Previous' button
bt-pdc.gif	Style c 'Previous' button, disabled
bt-nc.gif	Style c 'Next' button
bt-ndc.gif	Style c 'Next' button, disabled

Really, for each corresponding group, the files are all based on the same name, with a style identifier tacked on the end - 1 for style 1, 2 for style 2, c for style c.

You can replace the 'c' images with your own, and the rules are the same as for titles, make them GIFs, call them the same file names as the originals, and make sure that they stay in the tw-bt directory.

Examples of using the TIP_STYLE parameter

tip-wizard.html?white|black|yellow|red|purple|true|100%|times|2|1|...

...gives you a style 1 TipWizard - notice how the colour definitions have changed to match

tip-wizard.html?black|white|yellow|red|purple|true|100%|times|2|2|...

...gives you a style 2 TipWizard - there is now a black background, which works well for

2

tip-wizard.html?black|white|yellow|red|purple|true|100%|times|2|c|...

...gives you a custom TipWizard - try this to see the default custom images

tip-wizard.html?black|white|yellow|red|purple|true|100%|times|2|off|...

...turns off images, and gives you a fully text-based TipWizard.

Clearly there is a lot of experimenting to be done with colours and styles, but that is half the fun! If you come up with a really groovy style, e-mail me the URL and I'll check it out.

Also, in the future I may release a 'themes' archive for the TipWizard, which will give you a wide selection of custom appearances.

11. TIP_NUMBER

Aarrgh, finally the last parameter, and indeed, the most crucial. The tip number parameter tells the TipWizard which tip to display, and this forms the basis for using the prev and next buttons.

Actually, it's rare that you would start the TipWizard at a particular tip, even though this parameter allows you to do that. TipWizard has been designed to be random, so that every time your page is reloaded, a new, randomly selected tip is displayed. This is where the n value comes in.

Using the n value

If you were to specify a number as the tip_number, you would be shown a tip corresponding to that number as defined in your script. e.g. tip_number 1 displays tip 1, tip_number 4 displays tip 4, tip_number 20 displays tip 20.

However, if you replace the number with the letter n, you get tip n. What does this mean? Well, n tells the TipWizard to select a tip at random, and return that tip. So in one instance, tip_number n might show tip 13, and on the next reload, tip_number n may show tip 7. See how it works?

The idea is, your frame source (or SRC) will point to a tip wizard with tip_number n, so that when the page is loaded it will show a random tip, from which the user can scroll back and forwards (prev and next) through all of the tips using the buttons.

Examples

tip-wizard.html?black|white|yellow|red|purple|true|100%|times|3|2|6 generates tip 6

tip-wizard.html?white|black|blue|red|purple|false|100%|arial|4|1|n generates tip n

or a random tip

And that's it for the parameters!

In fact, if you copied the above examples, and pasted them into your browser (making sure the location points to the current directory of the tip-wizard.html file), you would actually get a complete, working TipWizard - try it out!

Section Two - Modifying the TipWizard JavaScript

So, with the information given above you can go off and create a TipWizard. But hang on, something's not right - why do Colin's stupid, dumb and boring tips keep coming up? Wheres my bleedin' tips?

Well, now you'll learn how to **edit** the tip-wizard.html file to create your own!!

WARNING - MAKE A BACKUP OF THE ORIGINAL TIP-WIZARD.HTML FILE SOMEWHERE BEFORE YOU GO EDITING IT - BETTER YET, KEEP A COPY OF THE ORIGINAL TIPWIZARD ZIP ARCHIVE SOMEWHERE IN CASE YOU WRECK THE IMAGES TOO - I DO HAVE FAITH IN YOU, BUT I SPENT SHEER MINUTES MAKING THOSE IMAGES, AND IF YOU WRECK THEM, I'LL CRY.

Editing the File

Load the tip-wizard.html file into Notepad or some other suitable text editor - but remember we have to save the file as html, so always remember to explicitly call the file 'tip-wizard.html', otherwise it will save as text.

Setting the number of tips

Now, don't worry about all the bits you may not understand - just look a few lines down into the script, and you'll see a line like this:

```
        numberOfTips = 7   // <---- SET YOUR NUMBER OF TIPS HERE
```

The first step in making your own tips is to change this number to the EXACT AMOUNT of tips you want to use. Perhaps the best way is to write up your tips on paper first, and then count them (you know, 1 2 3 4 5 etc.) and then change the existing number (in this case, a 7) to your number.

Don't worry about text with a // in front of it, they are just comments to help you, and do nothing but that.

So if you have 15 tips, the line would become:

```
        numberOfTips = 15   // <---- SET YOUR NUMBER OF TIPS HERE
```

It's that easy. Remember that if you add more tips later on to your file, this number MUST change.

Including your tip data (or text)

Ok, now that that is done, you are only one step away from 'tip-dom'.

If you scroll through the script right down to near to bottom, you'll find a **function** called

```
tipData(number)
```

Look at what is already present in this function. See how it works? The TipWizard (at the appropriate time) asks this function for a particular tip by specifying the tip's number. It's a bit like an array.

Look at the tip data - see how it is exactly what comes up as the default tips for TipWizard? Notice the HTML formatting.

Now if you understand what's happening, and how the data is formatted, it should be a piece of cake to add your own data.

Adding a tip

Tips follow this format:

```
if (number == x)
     return "--my tip with html formatting--"
```

Lets look at what it's doing:

The first line basically says in English: 'If the TipWizard asks for this tip, I'd better show it'
The == means equals, because if you look at the script, a single = means to assign a value to a particular variable.

The second line says: 'Ok, it wants this tip, so I'll give it this string for it's tip text.'

And that's basically it.

Your tips should start at 1, and finish at numberOfTips (the value which you specify at the top of the script). So if you want four tips, you'll have something like this:

Top of script: numberOfTips = 4

Bottom of script:

```
function tipData(number) {

          if (number == 1)
               return "This is my first tip"
          if (number == 2)
               return "..and this is my second one..."
          if (number == 3)
               return "...geez, for a third tip I'm pretty boring..."
          if (number == 4)
               return "...oh well, lucky last - I'm number four."
}
```

And that's all you have to do - pretty easy, isn't it? Isn't it? Hello? Are you still reading?

HTML formatting

Each of your tips can include HTML formatting, such as <B> for bold text, <FONT> for different fonts, and sizes. Experiment around, and see what you can do - also, look at the original tip data to see how I did it.

One important thing to remember though is that for any of your HTML that needs quotes e.g. " ", you must use single quotes e.g. ' ', not double, because if you put a double quote in, JavaScript will think you have finished the string you are returning.

So if you want to put an image in, for example, use:

<IMG SRC='myimage.gif'>

RATHER than:

<IMG SRC="myimage.gif">

..or you'll get errors - don't worry, I've done it plenty of times.

And that's it - the end!

Have fun creating your own TipWizards, and if you get them working well, possibly with your own custom images, please e-mail me the URL so I can check out your hard work!

Contacting me

If you'd like to ask any questions, give any comments, or to send great pails of abuse to me telling me how bad my TipWizard is (note - this is humour), then e-mail me at:

ctucke02@postoffice.csu.edu.au

and remember the web site at:

http://farrer.riv.csu.edu.au/~ctucker/tw

Have fun, and good luck!