



EYEmenu 3

1. About EYEmenu

EYEmenu3 is a great new alternative to existing browser navigation menus on the market today; with an easy implementation, small size and efficient code, EYEmenu3 is the way to go.

Features:

- Unlimited number of submenus
- HTML elements attached by menus (rollovers, images, links, etc.), and if these elements have events attached which are used by the script, then these are completed, not replaced
- Contextual menu with submenus
- The main menu can be static (stays in a fixed position on the screen when the page is scrolled)
- The menu might be permanently centered
- Submenus and the contextual menu will stay on the screen
- Total control of the submenu display
- Each option has complex graphical formatting options, including an image which can be changed on mouse over
- For browsing in IE 5.5+ transition effects from DirectX can be used
- Submenus can have rounded corners in browsers with CSS level 3 support or based on the Mozilla engine

Supported Browsers:

- IE 5
- IE 5.5
- IE 6
- Netscape Navigator 6
- Netscape Navigator 6.2
- Netscape Navigator 6.21
- Netscape Navigator 7
- Opera 6
- Opera 7

2. EYEmenu Terms Of Use

This script is free when it is used on nonprofit sites. Commercial sites must put the following HTML code on the top of the page:

```
<a href="http://www.eyecon.ro" target="_blank"></a>
```

Regardless of the type of the site where the script is used, the "eyemenu3.js" file cannot be edited or modified.

3. Integration in HTML pages

EYEmenu3' great features are all wrapped up in one file, **eyemenu3.js**. This file must be included in all HTML pages where it will be used. To include EYEmenu3 in your pages, insert the line of code below in between the HTML "HEAD" tag:

```
<script type="text/javascript" src="eyemenu3.js"></script>
```

Definition place for menus

To define a menu in your page, the following script syntax must be placed in between the HTML "BODY" tag:

```
<script type="text/javascript">  
    //the menus are defined here  
</script>
```

4. Graphical appearance

Setting's function

The graphical appearance, text size, border, rounded corners and other effects must be defined in a function. This function is:

```
eye.stil = new Array();
```

You do fill the function with the desired data. Wherever value is not wanted, `null` must be used.

Possible settings:

There are 25 possible settings:

```
eye.stil = new Array (  
01:  '#eeeeee',  
02:  '#eeeeff',  
03:  '#365689',  
04:  '#17263e',  
05:  '#000000',  
06:  '#000000',  
07:  '1px',  
08:  'solid',  
09:  '#000000',  
10:  'normal',  
11:  '12px',  
12:  'Arial',  
13:  'ar.gif',  
14:  'ad.gif',  
15:  '0',  
16:  '16',  
17:  '16',  
18:  '#eeeeee',  
19:  '1px',  
20:  '1px',  
21:  'filtru DirectX',  
22:  '6px',  
23:  '#000000',
```

```
24: 6,  
25: 16  
)
```

Each of the settings has the following specifications:

- **01:** text color in options
['#rrggbb' | 'rgb(rrr,ggg,bb)']
- **02:** text color on mouse over on option
['#rrggbb' | 'rgb(rrr,ggg,bb)']
- **03:** background color
['#rrggbb' | 'rgb(rrr,ggg,bb)']
- **04:** background color on mouse over on option
['#rrggbb' | 'rgb(rrr,ggg,bb)']
- **05:** icon's background color in main menu's options
['#rrggbb' | 'rgb(rrr,ggg,bb)']
- **06:** icon's background color in submenu's and contextual menu's options
['#rrggbb' | 'rgb(rrr,ggg,bb)']
- **07:** menu border in pixels
['00px']
- **08:** menu border style
['none' | 'solid' | 'dashed']
- **09:** menu border color
['#rrggbb' | 'rgb(rrr,ggg,bb)']
- **10:** font style
['normal' | 'bold' | 'bold italic']
- **11:** font size in pixels
['00px']
- **12:** font type
['Arial' | 'verdana' | 'Helvetica, Geneva, Verdana, Arial, sans-serif']
- **13:** path to image which will indicate the existence of a side submenu
['http://www.foo.ro/xxx.gif' | 'xxxx.pgn']
- **14:** path to image which will indicate the existence of a submenu above or under the main menu
['http://www.foo.ro/xxx.gif' | 'xxxx.pgn']
- **15:** numeric distance around the icons in options
['0' | '2']
- **16:** icon's width in options
['8' | '16']
- **17:** icon's height in options ['8' | '16']
- **18:** option's border color on mouse over
['#rrggbb' | 'rgb(rrr,ggg,bb)']
- **19:** border's dimensions in pixels (around options)
['00px']
- **20:** distance in pixels from menus margin to options
['00px']
- **21:** is a series of characters which represent the transition effect. To learn all the available effects, visit <http://msdn.microsoft.com/workshop/samples/author/filter/>
['progid:DXImageTransform.Microsoft.Pixelate(MaxSquare=20, Duration=1, Enabled=false)']
- **22:** corner's radius in pixels for submenus
['00px']
- **23:** separators color
['#rrggbb' | 'rgb(rrr,ggg,bb)']

- **24:** vertical motion of the submenus versus the position in which they appear
[6]'10']
- **25:** horizontal motion of the submenus versus the position in which they appear
[6]'10']

5. Main menu's definition

The main menu is always visible. The following settings can be modified:

- Vertical or horizontal
- Horizontal menu text can be centered
- Can be a bar as wide as the screen with the options aligned right, left or centered
- Static menu

The main menu's definition starts with the following command:

```
eye.bar(width,x,y,isvertical,isstatic,centered,isbar,align)
```

where:

- **width** – represents option's width
- **x** – the distance in pixels from left
- **y** – the distance in pixels from top
- **isvertical** – can have the value `true` or `false` and indicates if the menu will be vertical or horizontal
- **isstatic** – can have the value `true` or `false` and indicates if the menu will be static (stays in a fixed position on the screen when the page is scrolled) or not
- **centered** – can have the value `true` or `false` and indicates if the menu is always centered or not
- **isbar** – can have the value `true` or `false` and indicates if the menu is a bar or not (this option is still under testing and it's not working properly in all browsers)
- **align** – if the menu is a bar, indicates option's alignment and can have the following values:
 - 0 – options aligned left
 - 1 – options centered
 - 2 – options aligned right

After the main menu is created, options can be added. The following syntax is used to create an option:

```
eye.itm(icon,iconOver,text,link,target,submenu,side)
```

where:

- **icon** – represents the path to the image next to the option; if no image is desired, the value `null` is used;
- **iconOver** – changes the image next to the option on mouse over; if no rollover is desired, the value `null` is used;
- **text** – text in option;
- **link** – path to the link in option;
- **target** – target in option; if no target is desired, the value `null` is used;
- **submenu** – if option will contain submenus, the name of the submenu is specified (see Submenu's definition);if there are no submenus, the value `null` is used;
- **side** – the side where the submenu is displayed; can have the following values:
 - 0 – if the menu is vertical, the submenu is displayed to the right; if the menu is horizontal, the submenu is displayed under the menu;
 - 1 – if the menu is vertical, the submenu is displayed to the left; if the menu is horizontal, the

submenu is displayed over the menu;

If the main menu is vertical, the following separators can be inserted with the command:

```
eye.sep()
```

After all options are defined, the main menu must be closed with the following command:

```
eye.cls()
```

6. Submenu's definition

Submenus are used in the same manner as the main menu is defined, with some differences. A submenu is opened with the following command:

```
eye.sbm(width,name)
```

where:

- **width** – submenu's width;
- **name** submenu's name; helps identify it to be connected to other submenus;

```
eye.itm(icon,iconOver,text,link,target,submenu,side)
```

where:

- **icon** – represents the path to the image next to the option; if no image is desired, the value `null` is used;
- **iconOver** – changes the image next to the option on mouse over; if no rollover is desired, the value `null` is used;
- **text** – text in option;
- **link** – path to the link in option;
- **target** – target in option; if no target is desired, the value `null` is used;
- **submenu** – if option will contain submenus, the name of the submenu is specified (see Submenu's definition); if there are no submenus, the value `null` is used;
- **side** – the side where the submenu is displayed; can have the following values:
 - 0 – submenu is displayed on right-down;
 - 1 – submenu is displayed on left-down;
 - 2 – submenu is displayed on right-up;
 - 3 – submenu is displayed on left-up;

Between the submenu's options, separators can be inserted with the following command:

```
eye.sep()
```

After the submenu is defined, it must be closed with the command:

```
eye.cls()
```

7. Contextual menu

For Internet Explorer 5+ and Netscape Navigator 6+ the contextual menu can be replaced with an own menu. This menu is the same as any EYEmenu3 and benefits the same options.

The contextual menu is defined like a submenu, with one difference: the open way:

```
eye.cnt(width)
```

where **width** parameter represents the submenu's width.

Adding options, separators and closing is done the same way as on submenus:

```
eye.itm(icon, iconOver, text, link, target, submenu, side)
eye.sep()
eye.cls()
```

8. Attaching HTML elements

EYEmenu3 allows HTML elements to be attached (images, links, table cells, layers) to submenus. To make the process of attaching these elements easier, a more elegant method can be used. There are 9 methods to choose from.

EYEmenu 3 uses `onmouseover` and `onmouseout` triggers to display submenus attached by HTML elements. These elements can have functions attached to these two triggers; EYEmenu3 can determine this and not overwrite them (like with other scripts) and adds its own functions to the existing ones.

To understand better the way a submenu is attached to an HTML element, we'll use an example. Let's say that we want to attach a submenu with the name 'submenu' to a tag A (link). We have the link:

```
<a href="link" target="_blank"> example link </a>
```

All that is needed is to add the following in the opening tag:

```
eyemenu="submenu|1"
```

The tag will look like this:

```
<a href="link" target="_blank" eyedmenu="submenu|1"> example link </a>
```

What needs to be done therefore is to add the following attribute in the opening tag of the HTML element:

```
eyemenu="submenu|side"
```

where:

- **submenu** – submenu's name;
- **side** – where the submenu is displayed and has the following values:
 - 0 – from the upper-right corner of the element towards down;
 - 1 – from the lower-right corner of the element towards down;
 - 2 – from the lower-left corner of the element towards down;
 - 3 – from the lower-left corner of the element towards down-left;
 - 4 – from the upper-left corner of the element towards down-left;
 - 5 – from the upper-left corner of the element towards up-left;
 - 6 – from the upper-left corner of the element towards up;
 - 7 – from the upper-right corner of the element towards up;
 - 8 – submenu is displayed at the mouse coordinates when the mouse enters the element;

Recommendation: to avoid some incompatibilities using `onMouseOut` or `onMouseOver`

triggers, it is recommended to rewrite them as `onmouseover` and `onmouseout` respectively.

9. Avoiding problems with onload trigger

EYEmenu3 uses an `onload` trigger to initialize the menu. If this trigger is overwritten by another script, then the following command can be used to initialize the menu:

```
eye.incepe()
```

This way, problems with unloading elements can be avoided. At the end of the BOBY tag, the following can be added:

```
<script type="text/javascript">eye.incepe()</script>
```