Templates specification

Table of contents

Template structure

The file and directory structure of a template is as follows:

- *template.xml* – file containing template metadata (learn more about it in the next section).
- *thumbs.htm* – template file for thumbs page.
- *slide.htm* – template file for slide page.
- *preview.jpg* – gallery screenshot (320 × 240 px, 24 bit). It is advisable that the screenshot should contain usual user photos rather than pictures of posh cars, etc.
- *slide_loading.gif* – required image (usually 400 × 400 px), which is shown instead of a slide image while the latter is being loaded
- *thumbs_loading.gif* – required image (usually 120 × 120 px), which is shown instead of a thumbnail image while the latter is being loaded
- *images* folder – folder containing images used in the template.
  - *folder.png* – required image linking to album subdolders (an album can contain subfolders, i.e. animals » mamals » cats).

- *scripts* folder – folder containing template scripts.
- *styles* folder – folder containing template cascading style sheets.
  - All template styles should be held in one css file (*default.css*).
- *assets* folder – folder containing additional template files (sound, music, etc.)
- *preview_images* folder – folder containing two gallery screenshots:
  - image 800 × 600 px with all photos as a thumbnails (full page must be filled with photos)
  - image 800 × 600 px with one selected photo and other photos as a thumbnails

  you should name the screenshots like “template_big.jpg” and “template_big_2.jpg”

template.xml

Template metadata should be stored in *template.xml* file.

```
<TEMPLATE_INFO version="1.0" encoding="UTF-8">
<TEMPLATE>
      <DATE>YYYY.MM.DD HH.MM.SS</DATE>
      <INFORMATION>My template info</INFORMATION>
      <MIN_VERSION></MIN_VERSION>
      <NAME>My template</NAME>
      <OWNER></OWNER>
      <TAGS> simple, white, bright, etc. </TAGS>
      <TYPE>PWA</TYPE>
</TEMPLATE>
</TEMPLATE_INFO>
```

Template tags

As shown above, *template.xml* should contain tags describing a template. Tags should only describe the visual part of a template: color (e. g. *bright*, *yellow*), type (*artistic* or *simple*), theme (e. g. *holiday*, *birthday*), elements of design (e. g. *cats*, *pets*, *flowers*, *butterflies*), mood (e. g. *happy*, *gloomy*), etc.

The main aim of tagging is to help *Photo! Web Album* users during template search as much as possible, making it possible to provide them with the most suitable selection of designs.

You should not include technical details (such as *JavaScript* using, conformity with *XHTML Strict,* aspects of template make-up, etc.) as they are implicitly described by template structure itself or template constructions (see below).

In `<MIN_VERSION>` field type 1.1 – so it will be look like this `<MIN_VERSION>1.1</MIN_VERSION>`.
In `<INFORMATION>` field write a description of your template.
Leave the `<OWNER>` field empty.

Here you can see tags for some of our standard templates.

Beach
blue, turquoise, colored, colorful, bright, sand, sea, ocean, beach, grass, summer, water, aqua, vacation, holiday, travel, happy, fun, underwater, seaside, resort, serenity, shore, seashore, travel

Cats
blue, turquoise, black, colored, colorful, bright, cats, pets, animals, eyes, butterflies, flowers, fun, funny, happy, child, children

Dotted – Black
Black, simple, plain, neutral, elegant, dotted, dots, pure, one color, stylish, dark, frames, gloomy, blank

Dotted – White
White, grey, gray, simple, plain, neutral, elegant, dotted, dots, pure, light, one color, stylish, frames, blank

Ecological
green, light green, nature, natural, colored, colorful, artistic, leaf, plant, calm, delicate, plain, simple, calming, summer, spring, fresh

Frosty
White, light blue, light, snow, ice, pure, elegant, simple, plain, neutral, calm, winter, glow

Giraffe
Brown, beige, colored, colorful, ornament, patterned, sepia, warm, artistic, giraffe, animals, safari, sand, Africa, stylish, savanna, hot, heat, adventure, travel

Glamour
Black, grey, gray, silver, shine, silk, textile, fabric, stylish, diamonds, brilliants, glamour, artistic, jewel, pearl, fashion, fashy, glamorous, style, beauty, beautiful, modern

Gothic
Black, navy, dark, gothic, stained glass, stylish, artistic, ornament, patterned, elegant, antique

Happy Birthday
beige, red, bright, holiday, birthday, artistic, children, child, happy, present, ribbon, bow, frame, wood, celebration, party, anniversary, white, cake, album

Jeans
Blue, navy, jeans, artistic, colored, textile, fabric, leather, stylish, modern, casual

Merry Christmas
Blue, navy, winter, artistic, Merry Christmas, New Year, happy, holiday, celebration, decoration, snow, snowflake, holly, mistletoe, miracle, dark, exciting, Santa Claus

Metallic
Grey, gray, blue, silver, steel, shine, metallic, artistic, dark, shiny, stripes

Modern – Chocolate
Brown, beige, yellow, chocolate, cocoa, simple, neutral, stylish, warm, elegant, antique, retro, vintage, ornament, patterned, nostalgic, cappuccino, coffee, tasty

Modern – Pure
White, beige, yellow, simple, plain, neutral, stylish, pure, warm, elegant, antique, retro, vintage, ornament, patterned, nostalgic

Modern – Warm
White, yellow, beige, warm, simple, neutral, stylish, elegant, antique, retro, vintage, ornament, patterned, nostalgic

Old Album
Artistic, beige, brown, gray, grey, old, album, antique, retro, vintage, nostalgic, wood

Pets
blue, navy, green, colored, colorful, bright, dog, pets, animals, eyes, butterflies, flowers, grass, night, stars, child, children, fun, funny, happy

Simple
White, blue, navy, grey, gray, simple, neutral, black, frames, plain

Sport
Black, dark, sport, sporting, sports, football, ball, tennis, athlete, athletics, hobby, race

Stand
Brown, beige, wood, wooden, board, casual

Seasons – Autumn
yellow, ochre, orange, brown, colored, colorful, bright, stripes, warm, simple, plain, calm, elegant, autumn

Seasons – Spring
Green, light green, lilac, colored, colorful, bright, stripes, simple, plain, happy, elegant, spring, fresh, calm, calming

Seasons – Summer
Green, light green, colored, colorful, bright, stripes, simple, plain, happy, elegant, summer, spring, calm, calming

Seasons – Winter

Blue, white, colored, colorful, bright, stripes, simple, plain, happy, elegant, winter, calm, cool, cold, calming

Wedding
White, pink, light, pure, artistic, elegant, wedding, love, happy, holiday, celebration, hearts, rings, ornament, shine, glamour, pearl, red, newly-wed, wife, husband

Zebra
black, white, stripes, artistic, zebra, animals, safari, Africa, grey, gray, stylish, travel

Template constructions

To define template constructions, constants and functions are used.

Constant names

<%CONSTANT_NAME%>. Constant names should be written in upper case characters. If a constant name is built from several words, these words should be separated by underscores.

Function names

<%functionName()%>. The first word in a function name should start with a lower case letter; each next word should start with an uppercase letter.

File and folder names

folder_name. File and folder names all use lowercase with words separated by an underscore.

Names of HTML elements, CSS classes, etc.

. The rules for names of HTML elements and CSS classes are the same as for function names (the first word starts with a lower case letter; each next word starts with an uppercase letter). It is recommended that you use both tags (both *id* and *name*) because:

- The *id* attribute can act as more than just an anchor name (e.g., style sheet selector, processing identifier, etc.)
- However, some older and alternative browsers do not support anchors created with the *id* attribute.
- In its turn, the *name* attribute allows anchor names with entities.

Common constants and functions

Below is a list of constants and functions used in templates.

Constant	*Description*
<%GALLERY_TITLE%>	Gallery title
<%GALLERY_DESCRIPTION%>	Gallery description
<%INDEX_PAGE_URL%>	Link to the very first page (if the template contains first.htm, it will link to it; if it has no such page, it will link to the first page of thumbs.htm)
<%HOMEPAGE_URL%>	Link to gallery owner homepage

<%IS_HOMEPAGE_URL_NOT_EMPTY%>	Indicates whether the link to homepage is provided
<%GALLERY_HEADER%>	Place for customizable gallery header
<%GALLERY_FOOTER%>	Place for customizable gallery footer
<%IS_EXPORTED%>	Indicates whether the album has been exported (for displaying or hiding some elements after export only)

Function	*Description*
<%insertPager({*radius*}, {*beginPager*}, {*noncurrentNumberCell*}, {*currentNumberCell*}, {*emptyCell*}, {*endPager*})%>	The function returns pager links to adjacent pages. *radius* – the number of pager elements on each side of the current element If *radius* = 3, the pager can look like this: | … | 16 | 17 | 18 | **19** | 20 | 21 | 22 | … | If *radius* = 0, it will look like this: | … | **19** | … | If *radius* = –1, all elements are displayed. *beginPager* – HTML code of pager beginning *noncurrentNumberCell* – HTML code of pager element with non-current page number *currentNumberCell* – HTML code of pager element with current page number *emptyCell* – HTML code of empty pager element (used for alignment as in *Etalon* template) *endPager* – HTML code of pager end
<%if({*condition*}, {*trueBlock*}, {*falseBlock*})%>	*condition* – expression that evaluates to true or false; if condition is empty string, it evaluates to false *trueBlock* – HTML code inserted in case of true condition *falseBlock* – HTML code inserted in case of false condition

Constants and functions, used in *thumbs.htm* only

Constant	*Description*
<%NEXT_THUMBS_PAGE_URL%>	Link to next thumbs page
<%PREV_THUMBS_PAGE_URL%>	Link to previous thumbs page
<%UPPER_THUMBS_PAGE_URL%>	Link to upper level thumbs page
<%FIRST_THUMBS_PAGE_URL%>	Link to first thumbs page
<%LAST_THUMBS_PAGE_URL%>	Link to last thumbs page
<%ITER_THUMBS_PAGE_URL%>	Link to alternate page
<%ITER_THUMBS_PAGE_NUMBER%>	Alternate page number
<%ITER_THUMB_IMAGE_SRC%>	Path to thumbnail image
<%ITER_THUMB_IMAGE_WIDTH%>	Thumbnail image width
<%ITER_THUMB_IMAGE_HEIGHT%>	Thumbnail image height
<%ITER_THUMB_IMAGE_TITLE%>	Thumbnail image title
<%IS_TOPMOST_THUMBS_PAGE%>	Indicates whether the current page is the topmost
<%IS_FIRST_THUMBS_PAGE%>	Indicates whether the current page is the first one
<%IS_LAST_THUMBS_PAGE%>	Indicates whether the current page is the last one

Function	*Description*
<%insertThumbsMatrix({*width*}, {*height*}, {*beginMatrix*}, {*beginRow*}, {*cell*}, {*endRow*}, {*endMatrix*})%>	The function inserts the thumbnail image table. *width* – number of columns *height* – number of rows *beginMatrix* – HTML code indicating table

	beginning *beginRow* – HTML code indicating row beginning *cell* – HTML code of table cell *endRow* – HTML code indicating row end *endMatrix* – HTML code indicating table end

Constants and functions, used in *slide.htm* only

Constant	*Description*
<%CURRENT_THUMBS_PAGE_URL%>	Link to current thumbs page
<%NEXT_SLIDE_PAGE_URL%>	Link to next page
<%PREV_SLIDE_PAGE_URL%>	Link to previous page
<%FIRST_SLIDE_PAGE_URL%>	Link to first page
<%LAST_SLIDE_PAGE_URL%>	Link to last page
<%ITER_SLIDE_PAGE_URL%>	Link to alternate page
<%ITER_SLIDE_PAGE_NUMBER%>	Number of alternate page
<%ITER_SLIDE_PAGE_THUMB_SRC%>	Path to thumbnail image
<%IS_FIRST_SLIDE_PAGE_PAGE%>	Indicates whether the current page is the first
<%IS_LAST_SLIDE_PAGE_PAGE%>	Indicates whether the current page is the last
<%SLIDE_IMAGE_SRC%>	Path to image
<%SLIDE_IMAGE_WIDTH%>	Image width
<%SLIDE_IMAGE_HEIGHT%>	Image height
<%FULL_SIZE_IMAGE_URL%>	Link to full size image
<%SHOW_FULL_SIZE_IMAGE_URL%>	Indicates whether the link to full size image should be displayed
<%SHOW_EXIF%>	Indicates whether EXIF information should be displayed
<%EXIF_CAMERA_MANUFACTURER%>	Information about camera manufacturer imported from EXIF
<%EXIF_CAMERA_MODEL%>	Information about camera model imported from EXIF
<%EXIF_ORIGINAL_DATE%>	Information about original date of shot imported from EXIF
<%EXIF_ORIGINAL_RESOLUTION%>	Information about original resolution imported from EXIF
<%EXIF_ISO_EQUIVALENT%>	Information about ISO imported from EXIF
<%EXIF_METERING_MODE%>	Information about metering mode imported from EXIF
<%EXIF_EXPOSURE_TIME%>	Information about exposure time imported from EXIF
<%EXIF_APERTURE_VALUE%>	Information about aperture imported from EXIF
<%EXIF_FOCAL_LENGTH%>	Information about focal length imported from EXIF
<%EXIF_FLASH_FIRED%>	Information about flash imported from EXIF

Function	*Description*
<%insertThumbsPager({*radius*}, {*beginPager*}, {*noncurrentThumbCell*}, {*currentThumbCell*}, {*emptyCell*}, {*endPager*})%>	The function returns the thumbnails pager to adjacent pages. *radius* – the number of pager elements on each side of the current element *beginPager* – HTML code indicating pager beginning *noncurrentThumbCell* – HTML code of pager element with non-current image thumbnail *currentThumbCell* – HTML code of pager element with current image thumbnail

	emptyCell – HTML code of empty pager element (may be used for alignment, as in *Etalon* template) *endPager* – HTML code indicating pager end

XHTML

It is recommended that HTML code should follow XHTML-transitional standard. You can follow these guidelines: http://www.w3.org/TR/xhtml1/#diffs (please also take notice of the list of the most common errors here: http://en.wikipedia.org/wiki/XHTML#Common_errors).
You can ignore this declaration: `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`, if it does not make it possible to implement your design ideas.

Flexibility and cross-browser compatibility

It is also recommended to make cross-browser templates. This means they should look equally good (and preferably the same) in *Internet Explorer 6* and *7*, *Firefox 2* and *Opera 9*. It is also advisable to check that the template looks good in *Safari 3*.
The key recommendation for creating a cross-browser HTML template is to specify all properties explicitly. The less you set by default, the less problems there will be. If you encounter troubles, we also recommend trying to replace CSS parameters with corresponding HTML parameters, or vice versa.
The main recommendation for creating a cross-browser Javascript code is to use a cross-browser library, e.g. KLayers (Lesser GPL license).

Templates design should be as flexible as possible. Image height and width should not be fixed. It is also strongly recommended not to fix the size of thumbnails table, and therefore, page height and width. It is also advisable not to fix font size, i.e. set font size in *em* or *%*, but not in *pt* or *px*.

CSS vs. HTML

All parameters that can be defined by the user should be moved to the css file. This is utterly important for the album to be reorganized instantly.

Charset

The charset is 8-bit Unicode.

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

Semi-transparency

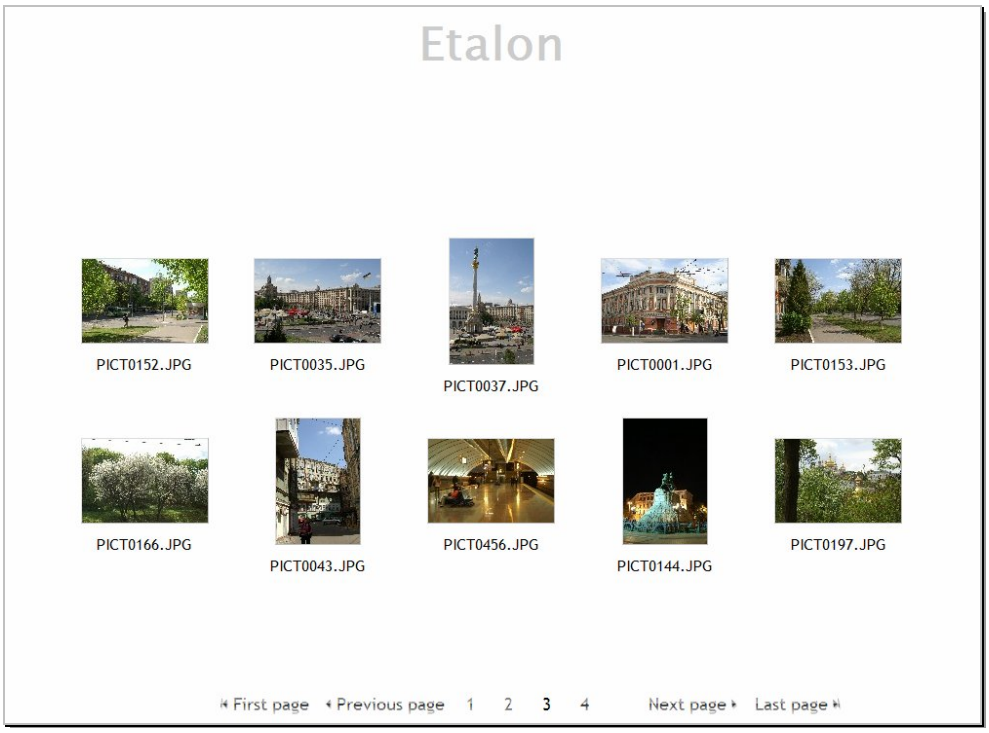
We recommend making all required PNG files 32-bit (with an alpha channel) for them to display correctly in different templates (on different backgrounds).
For 32-bit PNG files to display correctly in *Internet Explorer* lower than version 7 you can use the script in *png.htc* file of *Etalon* template.

Convenience of navigation

Standard templates should have the following navigation scheme:

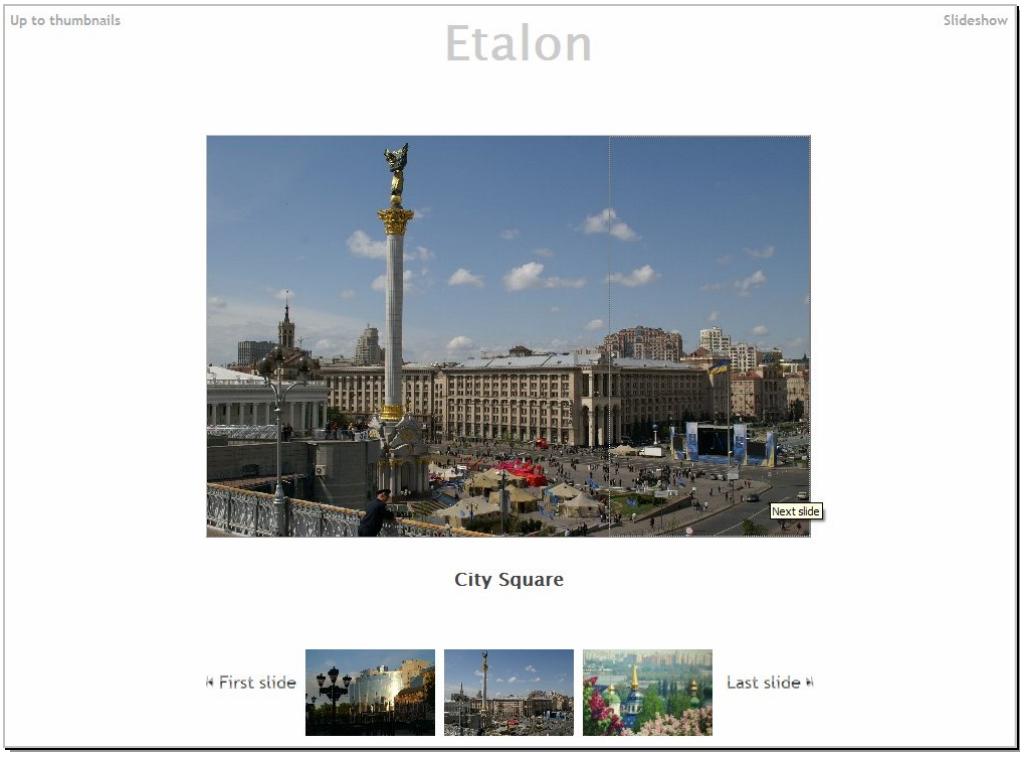
Pager navigation

An example of pager navigation is shown in the screenshot below. Each cell width should be constant (*2em* or *3em*). When the user clicks on, say, 18, he or she gets to page 18, while cell 19 gets directly under the cursor.



Navigation with the help of the photo itself

The image is divided into 3 sections. A click on the left part links to the previous image. A click on the right part links to the next image. A click on the central part links to the thumbs page.



Keyboard navigation shortcuts

Besides that, keyboard navigation should also be supported.

Left arrow	Previous page
Right arrow	Next page
Enter	On the slide page: open corresponding thumbs page
Space	On the slide page: Starts / stops slideshow

Subfolders support

The *slide.htm* page should contain *Up to thumbnails* button:

```
<a href="<%CURRENT_THUMBS_PAGE_URL%>">Up to thumbnails</a>
```

The *thumbs.htm* page should contain *Up one level* button:

```
<%if(
 {<%IS_TOPMOST_THUMBS_PAGE%>},
 { },
 {<a href="<%UPPER_THUMBS_PAGE_URL%>">Up one level</a>})%>
```

Transactions

We recommend using the following transaction in our standard templates:

```
<meta http-equiv="Page-Enter" content="blendTrans(Duration=0.5)" />
<meta http-equiv="Page-Exit" content="blendTrans(Duration=0.5)" />
```

Slideshow

The *slide.htm* should contain the *Start / Stop slideshow* button (as in *Etalon* template).

Printable version

The *slide.htm* should also contain the *Open full size image* button that links to the original image.

```
<a href="fullsize/IMG_0008.JPG" target="_blank"><img src="images/download.png" title="Open full
size image" width="17" height="17" border="0" alt="" /></a>
```

Real look

Each gallery should look and feel like a real object that this particular gallery depicts, not just like an HTML page with pictures. That is, if the gallery is designed as a cloudy sky, it would be inappropriate if photos threw shadows directly on the sky.
If a gallery looks like the centerfold of a glamorous magazine, navigation links should be either placed outside magazine pages, or look like, say, turned-in angles, but not like simple text links (i.e. *Next Slide*, *Previous Slide*).

Effects

We recommend using simple effects in our templates, i.e. highlighting image frame when hovering the cursor over the image, adding semi-transparent objects onto photos, etc. Advanced effects are welcome too, but they should be thoroughly tested in different browsers with different settings.

Image preloading

Images that appear only when moving the cursor over them should be preloaded. If you use *Adobe Dreamweaver* it can be done by inserting *Rollover Image* object or just with the following code:

```
<head>
<script language="javascript" type="text/javascript">
<![CDATA[
function MM_swapImgRestore() { //v3.0
  var i,x,a=document.MM_sr; for(i=0;a&&i<a.length&&(x=a[i])&&x.oSrc;i++) x.src=x.oSrc;
}

function MM_preloadImages() { //v3.0
  var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
    var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0; i<a.length; i++)
    if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}}
}

function MM_findObj(n, d) { //v4.01
  var p,i,x;  if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
    d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
  if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
  for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=MM_findObj(n,d.layers[i].document);
  if(!x && d.getElementById) x=d.getElementById(n); return x;
}

function MM_swapImage() { //v3.0
  var i,j=0,x,a=MM_swapImage.arguments; document.MM_sr=new Array; for(i=0;i<(a.length-2);i+=3)
   if ((x=MM_findObj(a[i]))!=null){document.MM_sr[j++]=x; if(!x.oSrc) x.oSrc=x.src;
x.src=a[i+2];}
}
]]>
</script>
</head>

<body onLoad="MM_preloadImages('next_light.gif')">
<a href="<%NEXT_SLIDE_URL%>" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('nextSlideButton','','next_light.gif',1)"><img src="next.gif" alt="Next
slide" name="nextSlideButton" id="nextSlideButton" width="119" height="79" border="0" /></a>
</body>
```

Requirements priority

If, while making up the template, you encounter problems that can be solved only by ignoring some of the recommendations and requirements listed in this document, we advise you to use this priority list:

1. Subfolders support.
2. Convenience of navigation.
3. Flexibility of image height and width.
4. Flexibility of thumbs table size.
5. Cross-browser compatibility.
6. Real look.
7. Image preloading.
8. Multi language support.
9. Simple effects.
10. Complex effects.
11. Font size flexibility.
12. CSS.
13. XHTML.