

Tutorial:

References template

22th of December, 2000

rev. 1

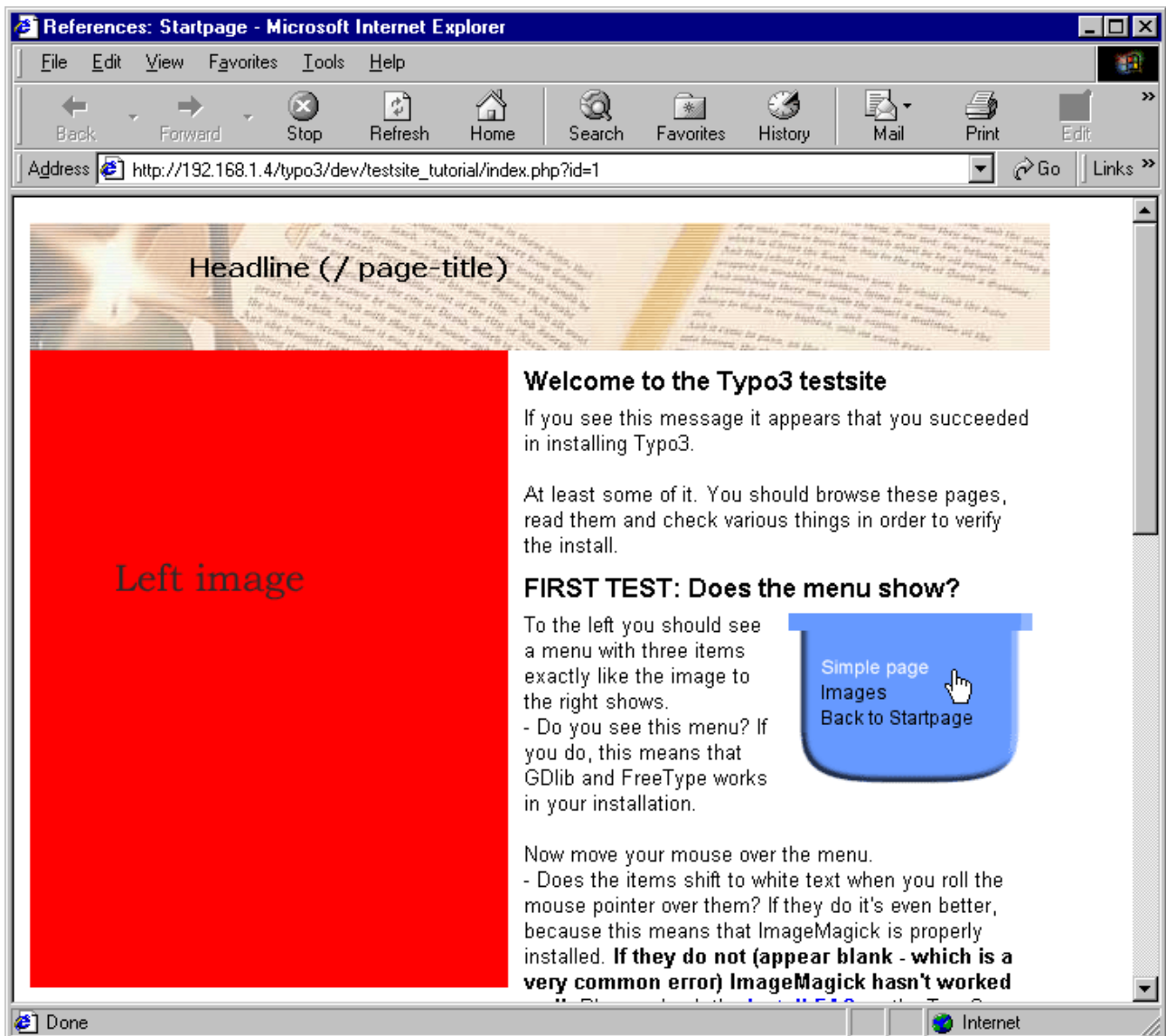
Copyright 2000, Kasper Skårhøj, kasper@curbysoft.dk

This document is published under the Open Content License

available from <http://www.opencontent.org/opl.shtml>

Setting up a new pagedesign for presentation of company references.

This tutorial presumes that you have read through the tutorial on how to setup a pagedesign with GoLive (or some other HTML-design program!). In this case we work within the same concept but to create another design with a "page-related" image on the left of the page. It should be possible to change this image from Typo3. This is an example of what we're trying to achieve:

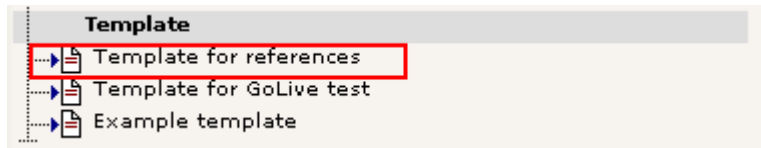


Basically you should prepare a template just like you did in the GoLive example. In fact you may make an exact copy of this record.
Anyway, clear the Constants-field and put this into the Setup-field of the duplicate of the "GoLive" template record:

```
page = PAGE
page.typeNum = 0

page.10 = TEMPLATE
page.10 {
    template = FILE
    template.file = fileadmin/template_references/template.html
    subparts.CONTENT < styles.content.get
}
```

Also change the title and make sure that the template is the first template in the record list!

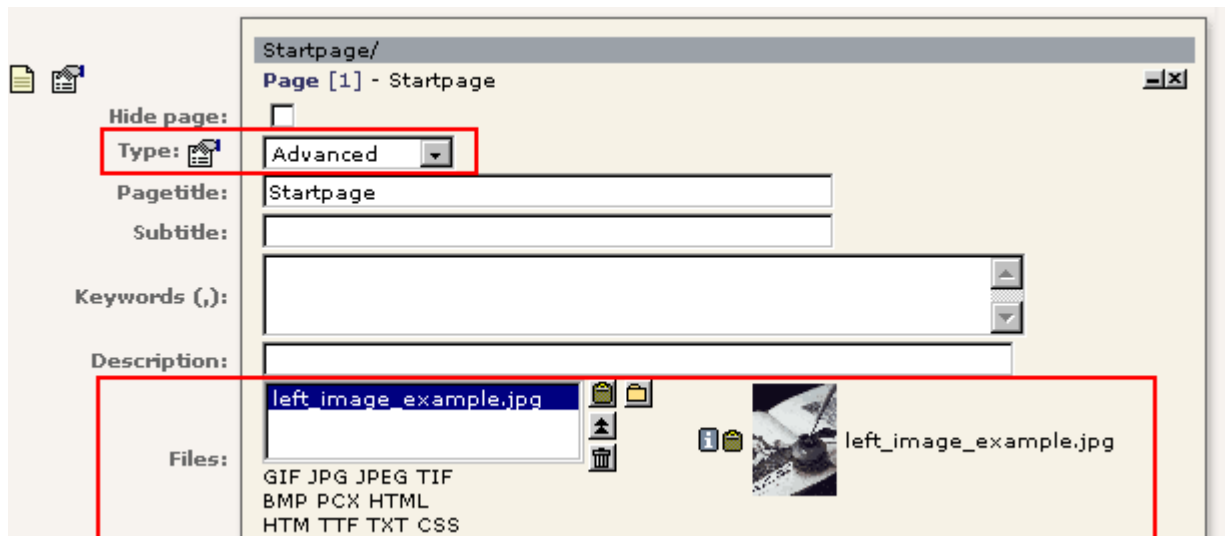


As you see this TypoScript code includes a template from "fileadmin/template_references/" which does not exist at this point. The files in this folder are included with this tutorial. You should now copy these files to the location "fileadmin/template_references/"

If you look at the frontend you should see the above layout of the site. (Remember to "Clear all cache"!)

The left image

The left image is the big task here. I have already prepared the template for the substitution of the default red image. This is not the hard thing though. It's rather; where in Typo3 should we store the image? The content-records are not good for this. You might use them, but it's not the best solution in this case. The solution is to attach the image to the page-record itself. If we change the type of the page to "Advanced", we can add media-files to the pagerecord. This feature is meant for exactly this kind of tasks!

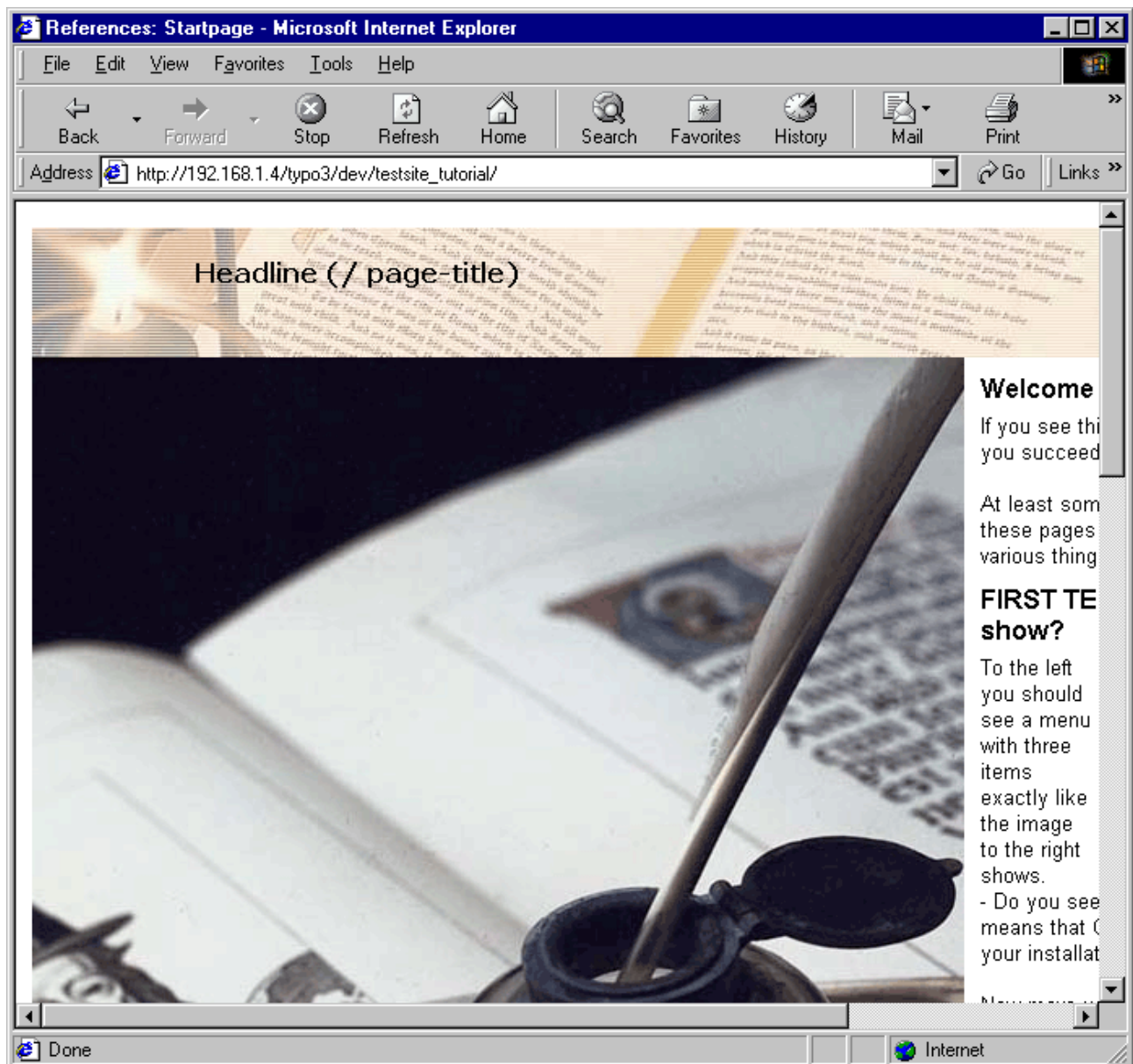


As you see I've already stored the image "left_image_example.jpg" with the pagerecord. The image comes with this tutorial.

Now we'll include the image on the page. Add the red lines:

```
page.10 = TEMPLATE
page.10 {
    template = FILE
    template.file = fileadmin/template_references/template.html
    subparts.CONTENT < styles.content.get
    subparts.LEFT_IMAGE = IMAGE
    subparts.LEFT_IMAGE.file {
        import.field = media
        import = uploads/media/
        import.listNum = 0
    }
}
```

This is the result:



The image seems to be too wide. Therefore we add a restriction on the width of the image:

```
subparts.LEFT_IMAGE.file {
    maxW = 300
    import.field = media
    import = uploads/media/
    import.listNum = 0
}
```

There we go!

The explanation is that **subparts.LEFT_IMAGE** is a **IMAGE** contentObject in TypoScript. The property ".file" is a **imgResource**, which happens to have a property ".maxW" for limiting the imagewidth. Please, refer to the TypoScript reference for these terms. You might experience some of the mysterious fog over TypoScript to disappear by such examples!

Oh, just one thing: I would like the whole page to have zero margins. Add this in the top of the template:

```
page.bodyTagMargins = 0
```

This should do the trick. The whole TypoScript code looks like this now:

```
page = PAGE
page.typeNum = 0
```

```
page.bodyTagMargins = 0

page.10 = TEMPLATE
page.10 {
  template = FILE
  template.file = fileadmin/template_references/template.html
  subparts.CONTENT < styles.content.get
  subparts.LEFT_IMAGE = IMAGE
  subparts.LEFT_IMAGE.file {
    maxW = 300
    import.field = media
    import = uploads/media/
    import.listNum = 0
  }
}
```

Advanced image processing

I would like to take the example a bit further. A very popular thing these days is to add horizontal lines to an image so it gets a "tv-screen"-effect. This can be automated with Typo3.

In order to do this you should know a bit about grayscale masks known from the image processing world. Basically, if you apply a mask to an image the grayscale values of the mask will represent the transparency of each pixel in the image.

Try to open the "horizontalline_mask.gif" image that comes with this tutorial. This is the mask we're going to use.

Now this may get a bit advanced. Try to change the proper part of the template record to this:

```
subparts.LEFT_IMAGE = IMAGE
subparts.LEFT_IMAGE.file = GIFBUILDER
subparts.LEFT_IMAGE.file {
    XY = [5.w],[5.h]
    5 = IMAGE
    5.file {
        maxW = 300
        import.field = media
        import = uploads/media/
        import.listNum = 0
    }
}
```

The image is now generated as a GIFBUILDER object where the image is inserted as an object - or "layer" - on the GIFBUILDER canvas. The canvas size is fetched from object number 5 - the image! - and therefore you should see absolutely no difference when you reload the template. Only that the image is now a gif-file.

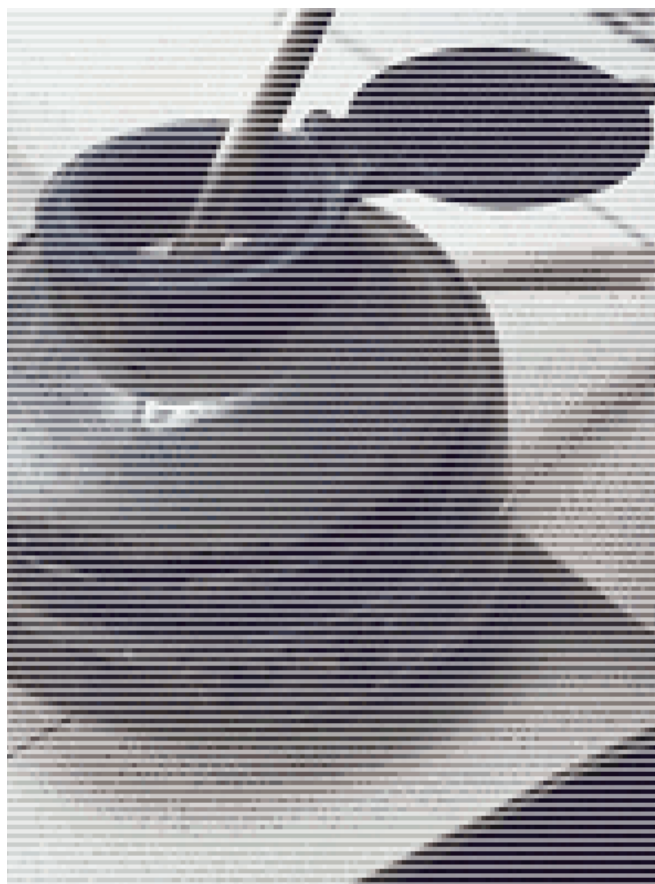
Now I want to apply my mask, so I add this line:

```
subparts.LEFT_IMAGE.file {
    XY = [5.w],[5.h]
    5 = IMAGE
    5.file {
        maxW = 300
        import.field = media
        import = uploads/media/
        import.listNum = 0
    }
    5.mask = fileadmin/template_references/horizontalline_mask.gif
}
```

And this is what I get:



Zoomed up a bit:



FIRST

To the
a menu
exactly
the right
- Do you
you do
GDlib a
in your

Now m
- Does

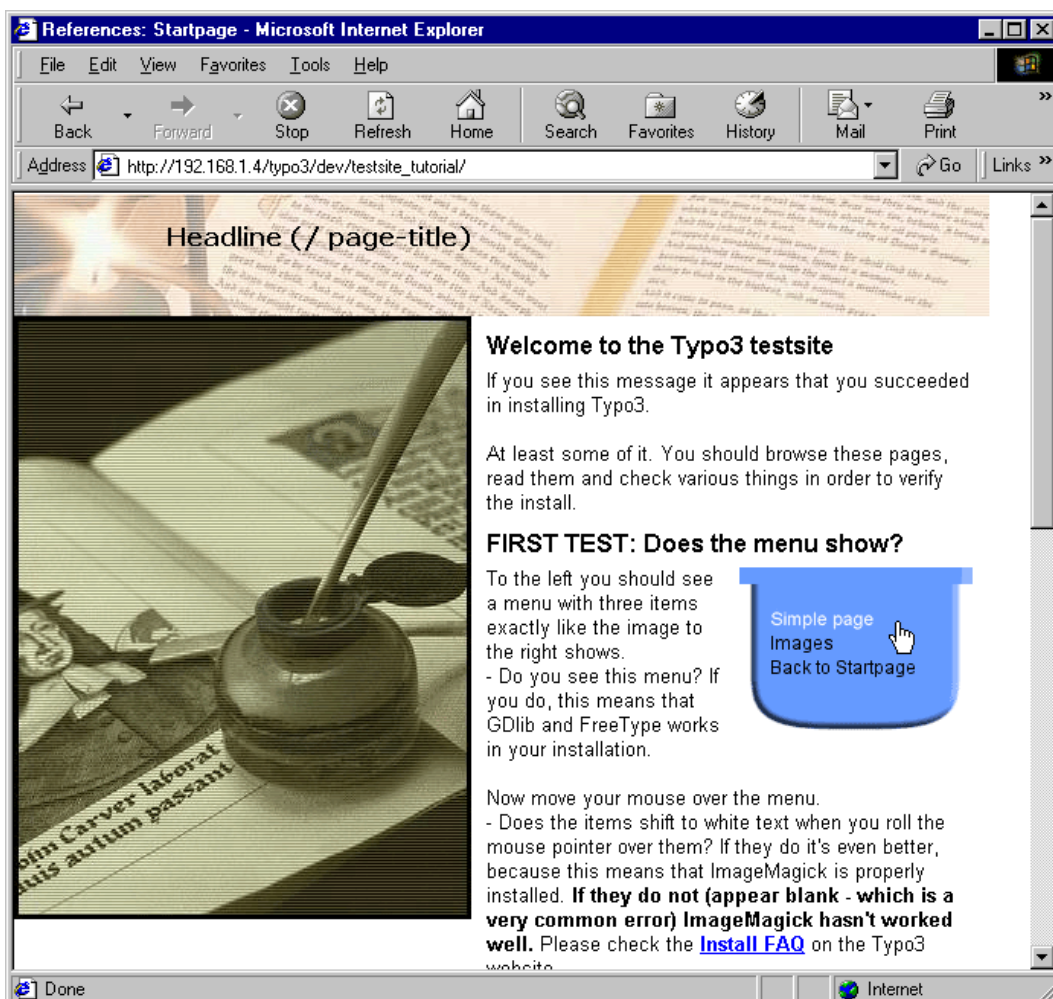
Now I would like to change a few things:

- add "olive" as my background color of the image
- add a black border to the image
- output jpg instead.

```
subparts.LEFT_IMAGE.file {
    format = jpg
    backgroundColor = olive
    XY = [5.w],[5.h]
    5 = IMAGE
    5.file {
        maxW = 300
        import.field = media
        import = uploads/media/
        import.listNum = 0
    }
    5.mask = fileadmin/template_references/horizontalline_mask.gif

    10 = BOX
    10.dimensions = 0,0,300,3
    10.color = black
    11 < .10
    11.align = ,b
    12 < .10
    12.dimensions = 0,0,3,600
    13 < .12
    13.align = r
}
```

This is the final result:



Content

I'll not go into the content rendering in this example, so here you must refer to the GoLive based tutorial.

But we still need to insert the page-header. Add the red lines to your template:

```
page = PAGE
page.typeNum = 0
page.bodyTagMargins = 0

page.10 = TEMPLATE
page.10 {
  template = FILE
  template.file = fileadmin/template_references/template.html
  subparts.CONTENT < styles.content.get
  subparts.HEADLINE = TEXT
  subparts.HEADLINE.field = title
  subparts.LEFT_IMAGE = IMAGE
  subparts.LEFT_IMAGE.file = GIFBUILDER
  subparts.LEFT_IMAGE.file {
    format = jpg
    backColor = olive
    XY = [5.w],[5.h]
    5 = IMAGE
    5.file {
      maxW = 300
      import.field = media
      import = uploads/media/
      import.listNum = 0
    }
    5.mask = fileadmin/template_references/horizontalline_mask.gif

    10 = BOX
    10.dimensions = 0,0,300,3
    10.color = black
    11 < .10
    11.align = ,b
    12 < .10
    12.dimensions = 0,0,3,600
    13 < .12
    13.align = r
  }
}
```