



My Industry Is.. ▼

About  
MISNews  
& Insights  
Blog

Let's Talk

Quote My Website

## TECHNOLOGY &amp; INNOVATION

# Random Content Generator

Learn Random Content, Random Content Rotation, Web Design, CSS, XHTML, Perl. A method of displaying random content on a Web page using CSS, that is both Search Engine friendly and accessible.

Enhanced  
CreativityIncreased  
InnovationInnovative  
ThinkingBoosted  
Self-  
ExpressionMIS Web Design  
LECTURER

1 THRS 45MINS

0 VIDEOS



## Paid Breakfast Coffee Menu at Course Talk

Next Hangout: Full

Past Hangout: Aug 28th @  
9:15amNext Hangout: Sept 28th @  
9:15amFuture Hangout: Oct 19th @  
9:15amRESERVE UP TO  
**2 HOURS**BREAKFAST MENU ITEMS  
**19**LOCATION  
**KIERLAND COMMONS**

*Technical level: **Intermediate** // Date: **8<sup>th</sup> December 2003** // Author: **Nigel Peck***

After reading Random Image Rotation by Dan Benjamin I thought I would share a method I recently used to rotate 'Success Stories'. The concepts and script I will present you with in this article can be used in the same way as Dan's to keep a page from becoming dull to repeat visitors.

The concept is similar but not the same; the script I will present to you here can be used to create random rotation of any (X)HTML code you choose. It is server based and not dependant on JavaScript.

## Overview

You create a number of <div>s. You import a style sheet generated by a perl script. The style sheet sets all but one random <div> to display:none. Your design looks fresh, Search Engines pick all of the content up (so any links in the random content are seen every time), and screen reader users get it all too\* so it keeps your content accessible. Simple, but useful.

\* - As pointed out by Joe Clark some screen readers respect display:none so this does not apply to those delinquents. You can use something other than display:none if you wish.

## Installing the Script

The first thing you need to do is to grab the script.

The script is written in Perl and should be usable on nearly all Web hosts around, please refer to your Web host's support desk or help files if you do not know how to get this script working. The only line you may need to change is the first line of the script that points to the perl executable on your server.

## The Example

As an example we will be using three 'Success Stories' that are a page of content each:

- success1.html
- success2.html
- success3.html

On our homepage we only have room to point to one of these Success Stories at a time when we are displaying the page to a visual browser, but there's no reason that screen readers and search engines shouldn't see all three every time.

We create three sections of code that will be used on the homepage for leading users into the full success story content pages:

```
<div id="success1" class="success-story">
<p><a href="/success1.html">Foo Ltd. has been invaluable in
bringing our product to market successfully.....</a></p>
</div>
```

```
<div id="success2" class="success-story">
<p><a href="/success2.html">Foo Ltd. delivered a highly
professional service.....</a></p>
</div>
```

```
<div id="success3" class="success-story">
<p><a href="/success3.html">Foo Ltd. provided the project
management skills we were looking for.....</a></p>
</div>
```

We have a <div> for each section of content that we want to be in the rotation.

The content of the <div>s is not important to the script, the important part is the id of each <div>, which has two sections; a prefix and a number. The class of the <div> can be whatever you need for CSS display purposes. You can of course use the id but you would have each id in your selector, using one class makes this easier.

## The Prefix

The first part of the id is the prefix, this is everything that comes before the number, in this example:

success

We will be passing this prefix to the script.

## The Number

The second part of the id is the number; these must always start at 1 and go up in increments of 1 (1,2,3 etc.). You can have as many as you like.

We will be passing the highest number to the script.

## The Script

The script uses a simple concept, it generates a style sheet which will hide all but one (random) <div> using display:none. We add it to our homepage using the standard <link> element:

```
<link type="text/css" rel="stylesheet" href="/cgi-bin/  
random_content.pl?t=3&p;=success" />
```

The important part here is the href that point to the style sheet:

```
/cgi-bin/random_content.pl?t=3&p;=success
```

Let's break it up and look at each section in turn:

`/cgi-bin/random_content.pl?`

The path to the location of the script on your server followed by a question mark.

`t=3`

t for total. You should replace 3 with the total number of content <div>s you used above.

`&`

You need the ampersand to separate the two values we are passing.

`p=success`

p for prefix. You should replace success with the prefix you used for the ids of your divs.

And that's it. Users that don't make use of the CSS such as search engine crawlers and screen readers that ignore CSS will get all of the content every time. With a little imagination there are many interesting things you can do with this simple script. It worked very nicely for me; I hope it does for you too.

## Changing The CSS

As I mentioned earlier you do not have to use `display:none`, you can change it if you wish. One alternative is to absolutely position the content so that it is off the visible screen area, say 1000 pixels to the left, to do this you would change the following line in the script:

```
my $css = '{display:none;}';
```

to:

```
my $css = '{position:absolute; left:-1000px; top:0px;}';
```

This should make non-random content more accessible to screen readers. The choice is yours.

## A Parting Comment

Thanks to the brilliance of a certain browser you may have problems if you try to view the output of the script directly in your browser. This is because it does not listen to the MIME type that the script outputs when you view the output in the browser (it works fine when it imports it as a style sheet).

If you want to view the output using this browser you will need to add `&debug=1` to the URL you use to access the script. This causes the script to output using a MIME type of text/plain instead of text/css. Don't use `debug=1` when importing as a style sheet, only add it if you are having problem viewing the output of the script in your browser for testing purposes.

The other alternative is to rename the script to `.css` which should also fix this problem without the `debug=1` as the aforementioned wonderful browser will then believe it's really getting css.

If you would like to see the site that I used this technique on feel free to mail me and I'll send you the URL: [info@miswebdesign.com](mailto:info@miswebdesign.com). I'd also be interested to hear from you if you have suggestions for improvements.

# Up Your Potential with Professionals & Like-Minded Owners

- ✔ Explore New Horizons
- ✔ Tech Mastery
- ✔ Collaborative Learning







We're the web design agency for ambitious brands. We fix website conversion problems and align brand identity.

**EMAIL US**  
office@miswebdesign.com

**I have a ..**

**Automotive Website**

**Sub-Contractor Website**

**General Contractor Website**

**Law Firm Website**

**Financial Services Website**

**Not Listed. See Industry List**

**I'm seeking a...**

**Phoenix Web Designer**

**Las Vegas Web Designer**

**Boston Web Design**

**Orlando Web Designer**

**Not Listed. See City List**

**MIS Web Design Agency**

**About Company**

**News & Insights Blog**

**Quote My Website**

**Let's Talk**

---

Masters of Internet Strategy Web Design © Copyright 2025.  
All rights reserved.  
MIS Web Design and the MIS Web Design logo are registered trademarks of MIS Web Design Agency.

**Terms of Service**

**License**

**Privacy Policy**