

/CSS/ create time-sensitive sites



Stuart Loxton, web developer at design agency Strawberrysoup (strawberrysoup.co.uk), explains how to create subtle CSS changes to your site based on time of day

Knowledge needed Working knowledge of CSS + PHP

Requires Working test environment with PHP5 enabled

Project time 30 minutes

In today's web design environment it takes more and more to attract visitors visually, and even more to make them keep coming back. Of course, this mostly depends on content, but small, subtle details can make any visitor become excited. In this tutorial you'll create a site that changes its background based on the time a person visits it. Afterwards, you should have the knowledge needed to adapt the script to any other data you want to use.

All server-side examples in this tutorial will be based on PHP, so make sure you have a working test environment with PHP5 enabled. If you're using any other server-side scripting tool it's likely there'll be a guide to accomplish a similar effect online. However, reading through this guide you'll probably recognise similar syntax and functions, enabling you to convert it to the language you're using.

Included on this issue's CD is a folder, **tut_css**, that contains all of the base files needed to complete this project. Move the folder into your testing server's root directory. Open it in your preferred IDE and then view **index.php** in a browser. If you've got PHP enabled and everything was copied successfully then you should see a basic white page with a header, navigation on the left and lipsum on the right. It's now time to start the dynamic styles.

PHP sunsets and sunrises

Using PHP to retrieve and make decisions based on time has its advantages and disadvantages. The largest disadvantage is the fact that PHP can't access the user's computer's time and can only get the time of the server. The advantage is that PHP has a large set of functions that make dealing with time that little less painful. Open up **index.php** and, before the start **<html>** tag, type the following PHP. Afterwards, I'll explain the key parts of the script.

```
<?php
$time = date('G');
$sunrise = date_sunrise(time(), SUNFUNCS_RET_DOUBLE);
$sunset = date_sunset(time(), SUNFUNCS_RET_DOUBLE) + 1;
if($time >= $sunrise && $time < $sunrise + 2) $style = 'sunrise';
elseif($time >= $sunrise + 2 && $time < $sunset) $style = 'day';
elseif($time >= $sunset && $time < $sunset + 2) $style = 'sunset';
else $style = 'night';
?>
```

Lines 2-4 set three variables. **\$time** is set to the current hour. **\$sunrise** is set to the sunrise time for today. **\$sunset** is set to one hour later than the sunset time. From my experience, official sunset times start an hour or two before what most people would call sunset starts.

Lines 5-8 run a set of standard **if** statements to determine which style to show: sunrise, day, sunset or night. This is then set to the variable **\$style**. This can now be used whenever a reference to the time of day is needed, as I'll show below.

There are three main ways to having dynamic CSS. Which one you use depends on several factors: the number of styles that change per data; the number of style sheets that get loaded into the page, and the amount of customisation you have on your server.

You'll create a site that changes its background based on the time a person visits it

1) Setting a body id/class: The most popular method of using dynamic CSS is to set the body id or class of the page depending on the styles needed. The advantage of this is that no server changes need to be made: you can use the same id/class across style sheets with no extra code and it's easy to quickly prototype and debug. The disadvantage of this is that you can only use pre-defined constants. Because of this you can't have random colour palettes using this method.

2) Including style sheets: The second most popular method is to include extra style sheets depending on which styles you need. This can be beneficial when a lot of styles change depending on what's needed; this stops the style sheet having to include styles for all other possible states of the site together. The disadvantage is that you have to create a style sheet per state. This can be a big hassle if you're going to have a lot of different styles. It also inherits the flaw of not being able to access the data in the style sheet.

3) PHP in the style sheet: This approach can be done in several ways. When used successfully and appropriately, it can create the largest amount of freedom. With data being directly accessible in the style sheet, there's no limit to the amount of states possible. The disadvantage is that if you want your style sheets to still have the .css file extension you need to customise the server; this also means you lose the portability of the style sheet. The other



All in the timing Small, subtle details can bring a website to life. In this tutorial you'll create a site that changes its background based on the time a person visits it

option is to use .php as the CSS extension. Another disadvantage is that you're limited to having all the rules in one file without having to duplicate and reprocess the PHP.

In our example there are only a few things changing per state and there are only four states. Based on the advantages and disadvantages outlined above, setting a body id is probably the best way of achieving the effect.

Change the <body> tag to the following:

```
<body id="<?=$style?>>
```

Then open **base.css** and add the following to the end of the file.

```
/****** TIME SENSITIVE CSS *****/
#sunset {
background: url('../images/sunset.jpg') repeat-x #BE7001;
}
#sunrise {
background: url('../images/sunrise.jpg') repeat-x #EFC501;
}
#day {
background: url('../images/day.jpg') repeat-x #0A6FBF;
}
#night {
background: url('../images/night.jpg') repeat-x #00123A;
}
```

Open your page in a browser now and admire the scenery, depending on what time of day it is.

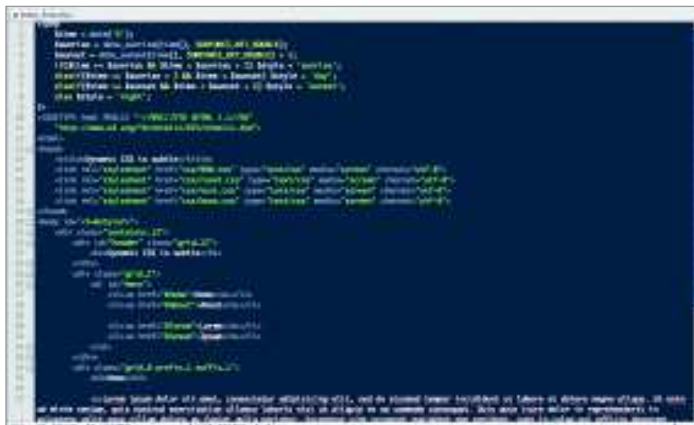
Including style sheets is another simple yet effective method. The practice involves having a separate style sheet for each state. To adapt the current site to use this method, create four style sheets called **sunset.css**, **sunrise.css**, **day.css** and **night.css**. Place the appropriate code for each state in its file, then remove the id on the body tag and instead place the following code in the <head> tag.

```
<link rel="stylesheet" href="css/<?=$style?>.css" type="text/css"
media="screen" charset="utf-8" />
```

Out of the three methods outlined this one is by far the most flexible and powerful. However, it also requires the most work, so if a stage doesn't work for you, go back and repeat again. If it still isn't working as it should then a quick search online will normally lead you to a solution to the same problem someone else had.

Setting .css to be run as php

This is a matter of personal opinion; however, I feel that having a .css extension helps define what an external asset is doing. This stage isn't required, so if you don't mind having CSS with the .php ending, or your server doesn't allow you to make the following changes, then skip this step.



Pros and cons PHP can't access the user's computer's time, but on the plus side it has a large set of functions that make dealing with time a little less painful

Taking it further

Your next steps in dynamic CSS

Dynamic CSS isn't restricted to simple time and date manipulations: any data that your server or client's browser can access can be used. Whether it's pre-determined states such as seasons or based on data hosted elsewhere, here are a few extra ideas.

Location

Retrieving the visitor's location is a tricky thing, although there are several popular libraries and snippets readily available to make things easier. Mozilla is also pioneering the methods in which a server and site can detect a user's location. Having the background change to complement your location could be an easy way to make your site more localised.

Weather

Having your site's background change depending on the weather could add function as well as design to a site. Visitors to a local harbour could quickly determine the current weather or water height before even getting to the page.

RSS Feeds

Easily accessible and retrievable RSS feeds now span the spectrum of content on the web. Colour palettes, political data, surveys and pretty much anything you could want is probably already in RSS format.

Local site data

If your site has dynamic content, why not complement it with some dynamic styles? If you have a blog then you could have the background represent the category that was last blogged about.

Create a .htaccess file in the root of your site and place the following line in it:

```
AddHandler application/x-httpd-php .css
```

This sets PHP to handle all files with a .css extension, enabling you to place php in a .css file as if it were exactly the same as a .php file. If it fails to do anything at first, restart Apache and see if that solves the problem.

Setting the content-type

The most important thing to do when using dynamic CSS is to stop PHP setting the content-type header as **text/html**. You need your CSS files to be sent to the browser with the content-type **text/css**. To do this, place the



Go further The code in this tutorial changes the background according to time; the same technique can be used to adapt to location or weather





Sunny disposition Strawberrysoup's site for Oasis Overland (www.oasisoverland.co.uk) will feature a dynamic background that changes depending on the time of day



Sky's the limit This time-sensitive background fits in nicely with the adventure travel specialist's ethos and enhances the website's appeal to its target market

>> following line at the top of **base.css** (or **base.php** if you didn't complete the previous step).

```
<?php header('Content-Type: text/css'); ?>
```

To get the background to change now, place this just below the line you just wrote at the top (you can remove all the PHP from the **index.php**).

```
<?php
$time = date('G');
$sunrise = date_sunrise(time(), SUNFUNCS_RET_DOUBLE);
$sunset = date_sunset(time(), SUNFUNCS_RET_DOUBLE) + 1;
if($time >= $sunrise && $time < $sunrise + 2) $style = 'url(..images/sunrise_bg.jpeg) repeat-x orange';
elseif($time >= $sunrise + 2 && $time < $sunset) $style = 'url(..images/day_bg.jpeg) repeat-x blue';
elseif($time >= $sunset && $time < $sunset + 2) $style = 'url(..images/sunset_bg.jpeg) repeat-x black';
else $style = 'url(..images/night_bg.jpeg) repeat-x black';
```

```
?>
body {
background: <?=$style?>;
}
```

Opening the page in a browser should now show you a background simulating the great outdoors.

PHP isn't the only way to add dynamic elements in your sites: JavaScript has been the raw base to site dynamics for years. Of the three methods above, the first two will work fine in JavaScript with adaptations. JavaScript has a few disadvantages. There's a delay between page load and the new styles. JavaScript is also notorious for browser differences, and it doesn't have the same resources as PHP, so can't provide sunrise and sunset times. For this reason, in the examples below the state will be statically set to **day**.

```
var state = 'day';
document.getElementsByTagName('body')[0].id = state;
```

A general breakdown of the code above is that it assigns the variable **state** a value of **day**. The second line then starts by selecting the document, finds all the elements with the tag name **body** (should always be just one) and then gives the first (0 nth element) the id of **state**.

```
var state = 'day';
var cssFile = 'css/' + state + '.css';
var link = document.createElement('link');
link.setAttribute('href', cssFile);
document.getElementsByTagName('head')[0].appendChild(link);
```

Based on the three PHP methods demoed here, there are a number of different ways you can make your CSS dynamic. If you need more ideas, though, just see the Taking it Further box on the previous page. ●

Resources Time-sensitive sites



Taproot Creative
This agency's website utilises two style sheets, day and night. The added touch of seeing the site's colours change gives a nice touch of personality.
taprootcreative.com



Katgal.com
One of the earliest examples of time-sensitive switchers was on katgal.com. The new design doesn't have it but the example is still live at the URL below.
katgal.com/styleswitcher.htm



About the author

Name: Stuart Loxton
Site: www.strawberrysoup.co.uk
Areas of expertise: Standards-compliant web design, application development and eCommerce
Clients: Mike and Ike, BUPA, West Sussex County Council
Who would play you in a movie? Mike Myers