Screen Capture with PhantomJS

Since PhantomJS is using WebKit, a real layout and rendering engine, it can capture a web page as a screenshot. Because PhantomJS can render anything on the web page, it can be used to convert HTML content styled with CSS but also SVG, images and Canvas elements.

The following script demonstrates the simplest use of page capture. It loads the GitHub homepage and then saves it as an image, `github.png`.

```
var page = require('webpage').create();
page.open('http://github.com/', function() {
  page.render('github.png');
  phantom.exit();
});
```

To run this example create a new file called `github.js`. Copy and paste the above code into the `github.js` file. In the command line, run this newly created script with PhantomJS:

```
phantomjs github.js
```

Beside PNG format, PhantomJS supports JPEG, GIF, and PDF.

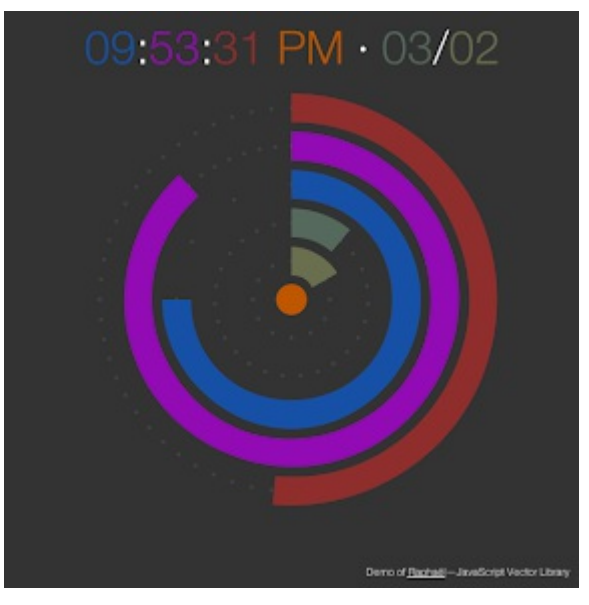
In the `examples` subdirectory, there is a script rasterize.js which demonstrates a more complete rendering feature of PhantomJS. An example to produce the rendering of the famous Tiger (from SVG):

```
phantomjs rasterize.js http://ariya.github.io/svg/tiger.svg tiger.png
```

which gives the following `tiger.png`:

Another example is to show polar clock (from RaphaelJS):

```
phantomjs rasterize.js https://dmitrybaranovskiy.github.io/raphael/polar-clock.html clock.png
```



Producing PDF output is also easy, such as from a Wikipedia article:

```
phantomjs rasterize.js 'http://en.wikipedia.org/w/index.php?title=Jakarta&printable=yes' jakarta.pdf
```

You can change the size of the screenshot and the webpage using the page's attributes:

```
  var page = require('webpage').create();
  //viewportSize being the actual size of the headless browser
  page.viewportSize = { width: 1024, height: 768 };
  //the clipRect is the portion of the page you are taking a screenshot of
  page.clipRect = { top: 0, left: 0, width: 1024, height: 768 };
  //the rest of the code is the same as the previous example
  page.open('http://example.com/', function() {
    page.render('github.png');
    phantom.exit();
  });
```

Canvas can be easily constructed and converted to an image. The included example colorwheel.js produces the following color wheel:

It is possible to build a web screenshot service using PhantomJS. Some related projects make it easy to create such a service.

← Back to the homepage