

How to convert presentation files to a web-ready slideshow using JavaScript

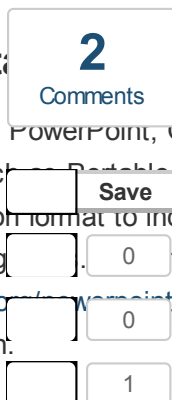
By Ryan Boudreaux in Web Designer, May 23, 2012, 11:00 PM PST

Ryan Boudreaux demonstrates a quick and easy method to convert presentation files, such as PowerPoint slides, to a web-ready slideshow using JavaScript.

This post will feature a simple set of JavaScript slideshow functions which does require you to manually save or export previously created presentation slides into individually named image files such as .png file types. While this might seem conventional, it is still quite easy to implement. One of the best reasons to convert presentation files to web-ready viewing is the elimination of software dependent viewing of specific file types (i.e., .ppt, .pps, .pptx, .key, .opd, .pez). Presentation slides converted to images allow viewing by all users no matter the presentation software. In addition, this simple set of JavaScript slideshow functions can also be used for creating a slide show of any gallery of images, and as long as they maintain a consistent height and width the presentation effects would remain the same.

On a quick side note, there are several frameworks that generate slideshows with impressive effects, and each of these utilizes their own combinations of web technology including Ajax, DHTML, JavaScript, HTML5, CSS3, jQuery, Python, and others. I will review several of these frameworks in later posts, which will include deck.js, impress.js, fathom.js, and landslide.

Export the presentation

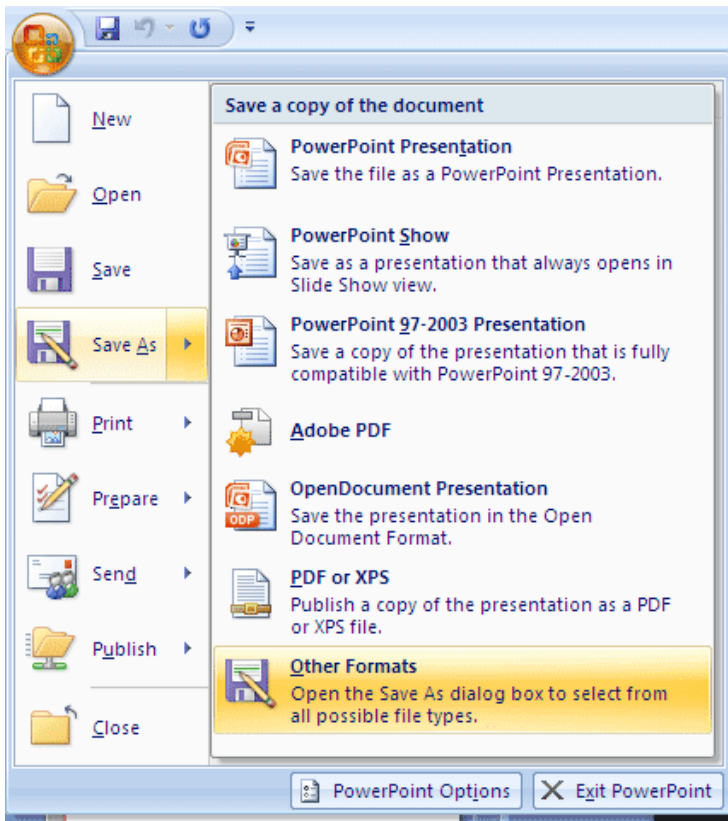


Whether you use Microsoft PowerPoint, Google Docs, Apple Keynote, or Prezi, the first step is to export the slides to individual image files such as Portable Network Graphics (PNG) format. This is not a literal export, as it is really a conversion from presentation format to individual image files. In this example, we will be saving the PowerPoint presentation slides as image files. The presentation I am using in this example was obtained from <http://mayas.mrdonn.com/slideshows.html>. For the example purposes of this tutorial, I will only use the first 15 slides in this demonstration.

For this demonstration, in PowerPoint 2007, the steps are quite easy.

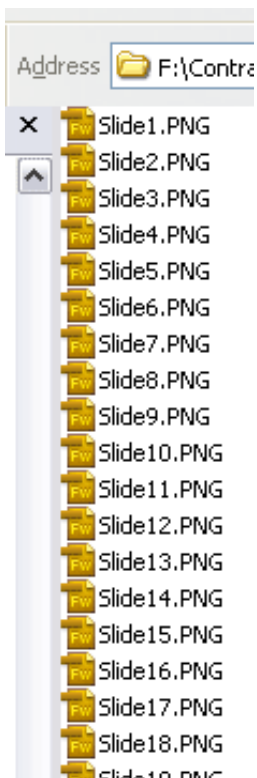
- Open the presentation in PowerPoint.
- Go to File | Save As and then select Other Formats.
- Select the image file type that you use in your web development; typical options include JPG, PNG, GIF, TIF, and BMP.
- Select the folder location to which you wish to save the images; each slide will be saved as an individual image.
- Select to save each slide as an individual image.

Figure A



In this case the presentation slides were saved to the selected folder as .PNG files as shown in **Figure B**:

Figure B



The code

This demonstration includes files within a directory named *web* and in the root of that directory is an HTML file

named *Slideshow.html*; a sub-directory named *slides* contains the images which were created from the presentation export in the previous step, and a sub-directory named *nav* contains the three navigation icons which are used for the Home, Previous, and Next onClick functions. The web directory with all the above mentioned files is [available for download](#) here.

Our HTML example file starts with creating several styles which are within the <head> of the document for ease of use, and it includes no text decoration for all anchor tags, setting the border to zero for all images, defining a box shadow for the "pic" id, setting the body background to color #619C80, setting the content id to center all text alignment with a top margin of 25 pixels, and finally setting the caption label "lbl" id font to larger. The style code is displayed below:

```
<style type="text/css">

  a {   text-decoration: none; }

  img { border: 0px; }

  #pic { -webkit-box-shadow: 5px 5px 10px rgba(58, 51, 50, 0.62);

        -moz-box-shadow: 5px 5px 10px rgba(58, 51, 50, 0.62);

        box-shadow: 5px 5px 10px rgba(58, 51, 50, 0.62); }

  body { background: #619C80; }

  #content { text-align: center;

            margin-top: 25px; }

  #lbl {font-size: larger; }

</style>
```

Next, we add in our content, which is contained within a <section > with id="content", and starts with our initial image with source where id="pic", alt and titles are also set. Then we add in an <article> which sets the image captions with id="lbl". Next, we add in a paragraph which contains three anchor tags with onClick functions for returning "Home", advancing to the "Next" image, or going back to a "Previous" image. Each anchor contains an appropriate icon for each function. The <section> code is displayed below:

```
<section id="content">

  <!-- Place the first image/slide here -->

  <br>
```

Next, we will add in our JavaScript at the bottom of the page, but before the closing `</body>` tag. First we set the variable `number` to zero, and then we define the `image` array with each image listed by image source, then image title and alternate text, then the image caption. The start of the JavaScript begins with the `<script>` tag, as shown below.

Add in a few lines of code below the opening `<script>` tag, and then we will enter in the variable and image array as shown below.

<http://www.techrepublic.com/blog/web-designer/how-to-convert-presentation-files-to-a-web-ready-slideshow-using-javascript/>

```
['slides/slide08.png', 'The Soil - Slide 8', 'The Soil'],

['slides/slide09.png', 'Geographical Dispersion - Slide 9', 'Geographical Dispersion'],

['slides/slide10.png', 'Mayan History - Classical Period - Slide 10', 'Mayan History - Classical Period'],

['slides/slide11.png', 'Mayan History - Classical Period - Slide 11', 'Mayan History - Classical Period'],

['slides/slide12.png', 'Mayan Agriculture - Slide 12', 'Mayan Agriculture'],

['slides/slide13.png', 'Mayan Agriculture cont. - Slide 13', 'Mayan Agriculture cont.'],

['slides/slide14.png', 'Cenotes - Slide 14', 'Cenotes'],

['slides/slide15.png', 'Mayan Religion - Slide 15', 'Mayan Religion']

]
```

Next, we add in our first function call to the `slideshow(slide_num)`, which sets the `document.getElementById` for the image source, image title, image alt and then the innerHTML image array. This function is displayed below:

```
function slideshow(slide_num) {

    document.getElementById('pic').src=imgArray[slide_num][0];

    document.getElementById('pic').title=imgArray[slide_num][1];

    document.getElementById('pic').alt=imgArray[slide_num][1];

    document.getElementById('lbl').innerHTML=imgArray[slide_num][2];

}
```

Next, we add the `slideshowUp()` function, which enables the `onClick` of the "Next" anchor and icon; when clicked, the next image in the array is displayed. This function is displayed below.

```
function slideshowUp() {

    num++;

    num = num % imgArray.length;

    slideshow(num);

}
```

Our next function is the `slideshowBack()`, which enables the `onClick` of the "Previous" anchor and icon. This function is displayed below.

```
function slideshowBack() {  
  
    num--;  
  
    if (num < 0) {num=imgArray.length-1;}  
  
    num = num % imgArray.length;  
  
    slideshow(num);  
  
}
```

And the final function, `slideshowHome()` enables the `onClick` of the "Home" anchor and icon, which when clicked will bring the first image in the array to be displayed. This also resets the array to 0 with `num = 0`; . The final function is displayed below.

```
function slideshowHome() {  
  
    num = 0;  
  
    slideshow(num);  
  
}
```

The initial slideshow image is displayed in **Figure C** as shown in Google Chrome 18.0.1.

Figure C



This is a simple and quick method of incorporating presentations into your website, and it gives your users access to

presentations without dependence on a particular file extension. In addition to the user input navigation, another option would be to include a set of JavaScript functions for automatically rotating the images with a set timeout display.



About Ryan Boudreaux

Ryan has performed in a broad range of technology support roles for electric-generation utilities, including nuclear power plants, and for the telecommunications industry. He has worked in web development for the restaurant industry and the Federal g...

[Full Bio](#) [Contact](#) [See all of Ryan's content](#)

2 comments

[Log In](#)

2 people following

		+ Follow Conversation	Share
Post comment as...			

[Newest](#) | [Oldest](#) | [Top Comments](#)**bernadbao**

Jun 10, 2013

Thanks for your valuable information. I found a powerpoint to html5 conversion tool named 'html5point' from digitalofficepro.com . It works great and retains animations and transitions even after conversion. <http://www.digitalofficepro.com/Powerpoint/powerpoint-to-html5-converter.html>

[Like](#) [Reply](#)**network**

May 25, 2012

Thanks for the nifty tip, but a warning for others who download the sample files. The names of the image files in the html file are cased differently than the files in the slides folder. I had to rename the files to match the html file before it would work. Works slick except for that one issue. Thanks again. Tom