If your Acrobat viewing ap has its JavaScript enabled, you should just have seen our title flash a number of times before settling down to a continuous display. You should also be able to repeat the effect by using your double back arrow to return to document start, or else by clicking here.

When not used to excess, brief flashing can attract viewer’s attention. Particularly in a PowerPoint Emulation. Graphics or individual words can also be flashed.

To start, make three copies of the flashing page, the first two without the text or graphics to be emphasized, the last one with. Set this JavaScript entry page action on your first page...

```javascript
global.repeats = 6;
global.delay = 178;
this.pageNum++;
```

And this entry page action on your second page...

```javascript
aaa = app.setTimeOut("this.pageNum++", global.delay);
aaa;
```

And this entry page action on your third page...

```javascript
global.repeats--;
aaa = app.setTimeOut("if (global.repeats > 0)
                   
                   {this.pageNum--}
                   
                   ", global.delay);
aaa;
```
If your Acrobat viewing ap has its JavaScript enabled, you should just have seen our title flash a number of times before settling down to a continuous display. You should also be able to repeat the effect by using your double back arrow to return to document start, or else by clicking here.

When not used to excess, brief flashing can attract viewer’s attention. Particularly in a PowerPoint Emulation. Graphics or individual words can also be flashed.

To start, make three copies of the flashing page, the first two without the text or graphics to be emphasized, the last one with. Set this JavaScript entry page action on your first page...

```javascript
  global.repeats = 6 ;
  global.delay = 178 ;
  this.pageNum++ ;
```

And this entry page action on your second page...

```javascript
  aaa = app.setTimeOut ("this.pageNum++", global.delay) ;
  aaa ;
```

And this entry page action on your third page...

```javascript
  global.repeats-- ;
  aaa = app.setTimeOut ("if (global.repeats > 0)
  
    (this.pageNum--)
  
  
"
  
  global.delay) ;
  aaa ;
```
Our first dummy page is used for initialization. It sets the `global.repeats` as the number of flashes before continuous display. And the `global.delay` as the time delay in milliseconds for each flash. A `this.pageNum++` is then used to unconditionally go on to the next page. This page could also be blank or have previous content on it.

Our second dummy page (without flashing text) stalls for `global.delay` and then goes on unconditionally to the third page, again using `this.pageNum++`.

Our third page (with the flashing text) decrements our repeat count with a `global.repeats--`. It then tests the `global.repeat` count to see if any repeat trips are needed. If needed, the display is delayed by the `global.delay` time, and the JavaScript takes us back to the previous page without the flashing text.

Presentation will be better in the one page layout mode rather than continuous. Some details such as keeping the page numbers consistent and being printer friendly are left as an exercise for the student.

**For More Help**

Enhancements and improvements on this fast, convenient, and super flexible .PDF flasher can be made available to you on a Custom Consulting basis. Additional GuruGrams are found here, PostScript topics here, and Acrobat info here.

Further GuruGrams await your ongoing support as a Synergetics Partner.