Document Read Time

Introduction

From time to time, I come across sites that provide an estimate of how long it will take to read an article. Reading time estimators use a basic formula like this one: count the number of words, divide it by an average word per minute (wpm) read, and round up the number in minutes. Images need to be accounted for as well (which use a base of 12 seconds and reduce the value per image down to three seconds). It should be noted that this article will not deal with videos as part of the estimated reading time.

The reading time estimate can be useful however it has a few pros and cons that should be considered before applying to your articles.

Pros

- Invites readers to continue with the article and this reduces bounce rates on your site
- Long articles may seem off putting to some but if the read time seems relatively low, the reader is more likely to finish the article (this is, in part, the paradox of choice)

Cons

- If the reader things the estimate is too high, they may skip the article all together
- The estimate is just an estimate. The estimate is based on an average reading speed (in this case, an US English reader). Your audience's reading speed will vary. Some say that the average US English reader reads at a 6th grade level (265 words per minute). Others say that they read at a college level (300 wpm). You'll have to decide which benchmark to use for your reading time estimator.

- The estimate excludes the title of the document and navigation elements outside the article element
- I disagree with the estimate that the first image viewed takes 12 seconds, the second 11 seconds, and so on (subtracting one second per image thereafter until you reach the floor of 3 seconds). I believe the image may be information heavy and the read time can vary wildly. Plus, this image estimate doesn't take into account static or animated images.

In this document, we'll explore a simple JavaScript snippet that will scan an article element, count the number of words and images, and provide a crude reading time estimate.

Requirements

In order to complete the code below, you should be comfortable with JavaScript, jQuery, regular expressions, and HTML5.

Setting Up The Variables

We'll set a variable called wpm to 265. You can change this variable to suit your audience needs. The next variable will be called wordCount and it will grab the text of the article element and perform some Regex magic to detect "words", and split this result into an array to be counted. The imgCount variable returns the number of img elements it finds within the article element. Finally, the tally variable will be used to keep track of the reading time (in seconds) of all the images found in the article element.

```
var wpm = 265;
var wordCount = $('article').text().replace( /[^\w ]/g, "" ).split( /\s+/
).length;
var imgCount = $('article img').length;
var tally = 0;
```

Now we need to tally the images. A simple for loop event will tally the total number of images by using a reduction iteration method to figure out if the reading time is above 3 seconds or not.

```
for (i = 0; i < imgCount; i++) {
  if (imgCount - i < 3) {
    tally += 3;
  } else {
    tally += 12 - i;
  }
}
```

Finally, we prepend our calculations to the article element. We use the prepend method from jQuery to add a span element which as the calculation of the reading time estimate (in minutes) rounded up.

```
$('article').prepend('<span alt="reading time">~' + Math.round((wordCount +
tally) / wpm) + ' minutes reading time</span>');
```

Wrapping It Up

There you have it. A simple reading time estimator in 12 lines of JavaScript. Just for kicks, you could use an info icon with hover text stating the metrics for this estimate to put the reader at ease with the provided estimate.