Compress and optimize pdf files in php

I'm not robot

reCAPTCHA

Continue

The Eric Novinson compression file involves methods to reduce the space needed to store a file in a computer system. Compressed files require significantly less storage space than uncompressed files, although compression can result in the loss of some data. File compression is designed to reduce data storage requirements that do not provide additional information, such as white space on a page. Squeezing files increases the speed of data transmission. High-speed Internet, such as cable and DSL connections, is not available in all places. Many computers use much slower communication systems, such as modems, that cannot transmit data very quickly. In addition, even a system that can handle large amounts of data still slows down when many users connect to it at once. The longer the file is required to send, the more likely it is that the transmission is interrupted, or that the transmission is corrupted and the user gets an unusable file. The disk space required on Internet servers is reduced by compressing files. Internet servers require physical storage tools such as hard drives, and compression of files allows you to store more information on these devices. The compression also reduces the time it takes an Internet server to find files stored on hard drives. Even if a client computer is connected to an Internet server using a high-speed fiber optic cable, the client still has to wait for the server to find the files in its own storage system. Squeezing files can also hide information. Not all computers can read the information stored in compressed files and list it in the file index. This can be useful if an internet server stores information that is not intended for the public, especially if the method of squeezing files is not reversible using commonly used computer software. In addition, encrypting a file so that unauthorized users can't read it increases the required storage space for the file, so compression of encrypted files is also useful. Many organizations, such as banks and online stores, use strong encryption to handle financial transactions, so using the compression method reduces their storage costs. Squeezing files saves energy. Every step of transferring files on the Internet uses the power, from the power required to run an Internet server hard drive, to the one used by the modem, router, and every other intermediary network device, before the information reaches the end-customer's computer. Squeezing files reduces energy bills associated with many devices. Squeezing files is mandatory on some internet servers. Server operators may not allow non-depressive transmission because they want to keep space in their systems. This requirement means that Internet users will need a file compression program, both to download data to these Internet servers and to read the downloadable data. According to The College, compression files are useful when sending attachments to emails that often have file size restrictions. Software engineers have always developed new ways to install a lot of data in a small space. This was true when our hard drives were tiny, and the advent of the Internet just made it more important. File compression plays a big role in connecting us, allowing us to send less data down the line so we can have faster downloads and place more connections on downloaded networks. So how does it work? To answer this question will include an explanation of some very complex mathematics, of course, more than we can cover in this article, but you don't need to understand exactly how it works mathematically to understand the basics. The most popular text compression libraries rely on two compression algorithms, using both simultaneously to achieve very high compression ratios. These two algorithms are L'77 and Huffman Coding. Huffman's coding is pretty complicated, and we won't go into the details of that here. First of all, it uses some quirky math to assign short binary codes to individual letters, reducing file sizes in the process. If you'd like to know more about this, check out this article on how the code works or this Computerphile explanation. On the other hand, the L-77 is relatively simple and is what we will talk about here. It seeks to remove duplicate words and replace them with a smaller key that represents the word. Take, for example, this short piece of text: the L'77 algorithm will look at this text, realize that it repeats howtogeek three times, and change it to it: Then, when it wants to read the text back, it will replace each instance (h) with howtogeek, causing us to return to the original phrase. We call compression like this without loss - the data you put into the same as the data that you come out. Nothing is lost. In fact, L'77 doesn't use a key list, but instead replaces the second and third phenomenon with a link in memory: So now that it gets to (h), it will look back at howtogeek and read that instead. If you are interested in a more detailed explanation, this video from Computerphile is very useful. This is an idealized example. In fact, most of the text is compressed with keys as small as just a few characters. For example, a word will be squashed even when it appears in words like there, and then. With repeated text, you can get some crazy compression ratios. Take this text file with the word howtogeek repeated 100 times. The original text file is three kilobytes in size. When compressed, however, it takes only 158 bytes. That's almost 95% compression. Obviously, it's extreme example, since we repeated the same word over and over again. In general practice, you're probably likely about 30-40% compression using a compression format like the SIP on the file, which is basically text. By the way, this L-77 algorithm applies to all binary data, not just text, although text is generally easier to compress because of how many repetitive words most languages use. For example, a language such as Chinese may be a little more difficult to compress than English. How does compression work? Video and sound compression works very differently. Unlike text, where you can have no loss of compression, and no data is lost, with images we have what is called Lossy compression, where you lose some data. And the more you compress, the more data you lose. This is what leads to those horrible-looking JPEGs that people have uploaded, shared, and screenshots several times. Every time an image shrinks, it loses some data. Here's an example. This is a screenshot I took that wasn't compressed at all. I then took that screenshot and ran through Photoshop several times, each time exporting it as a low quality JPEG. Here's the result. Looks pretty bad, doesn't it? Well, this is only the worst-case scenario, exporting 0% JPEG quality every time. By comparison, here's a 50% JPEG quality that's almost indistinguishable from the PNG image source if you blow it up and take a closer look. PNG for this image was 200KB in size, but it is 50% jpeg quality only 28KB. So how does it save so much space? Well, the JPEG algorithm is a feat of technique. Most images store a list of numbers, with each number representing one pixel. JPEG doesn't do any of this. Instead, it stores images using what's called the Discreet Cosine Transform, which is a collection of sinus waves combined with varying intensity. It uses 64 different equations, but most of them are not addictive. This is what the quality slider for JPEG's Photoshop and other image apps does-choose how many equations to use. Apps then use Huffman coding to reduce the file size even more. This gives JPEGs an insanely high compression ratio, which can reduce a file that will be a few megabytes to a few kilobytes, depending on the quality. Of course, if you use it too much, you'll end up with it: This image is awful. But a small amount of JPEG compression can have a significant impact on file size, and this makes JPEG very useful for compressing images on websites. Most of the photos you see online are compressed to save download time, especially for mobile users with poor data connection. In fact, all the images on How-To Geek have been compressed to make page loading faster, and you've probably never noticed. Video Compression Video Works A Little Differently Than You'd think they'd just squeeze every frame of the video using JPEG, and they'd definitely do it, but there's a better method for video. We use what's called compression that calculates the changes between each frame and stores only them. So, for example, if you have a relatively still shot that takes a few seconds in the video, a lot of space is saved because the compression algorithm doesn't need to store all things in a scene that doesn't change. Inter-frame compression is the main reason we have digital TV and web video at all. Without it, the video would be hundreds of gigabytes larger than the average hard drive size in 2005, when YouTube launched. Also, since inter-frame compression works best with mostly stationary video, that's why confetti destroys the quality of the video. Note: GIFs don't do this, so animated GIFs are often very short and small, but still have a fairly large file size. Another thing to keep in mind about the video is its bitrate-volume data allowed in every second. If your bitrate is 200 kb/s, for example, your video will look pretty bad. The quality goes up as the bitrate goes up, but after a few megabytes per second, you get a decrease in profits. This is an enlarged shot taken from a video of a jellyfish. On the left - 3 Mb/s, on the right - 100 Mb/s. 30x increase in file size, but not a significant increase in quality. Typically, YouTube videos sit around 2-10Mb/s depending on your connection, as more and more likely will not be seen. This demo works better with the actual video, so if you want to check it out for yourself, you can download the same bitrate test videos used here. Audio compression audio compression works very similar to text and compression images. Where JPEG removes parts from an image you won't see, compression of sound does the same for sounds. You may not have to hear the creaking guitar pick up on the string if the actual guitar is much, much louder. The MP3 also uses bitrate ranging from the low end of 48 and 96 kbps (low end) to 128 and 240kbps (pretty good) to 320kbps (high-end audio), and you'll probably only hear the difference with exceptionally good headphones (and ears). There are also no loss-compression codecs for the audio core of which is FLAC- which uses L-77 coding to deliver completely loss-free audio. Some people swear to FLAC the perfect sound quality, but with the prevalence of MP3, it seems most people either can't say or don't mind the difference. Difference. compress and optimize pdf files in php