

Come posare un watermark su un'immagine con la libreria GD

Nel [precedente tutorial](#), abbiamo visto come creare i thumbnail delle immagini caricate. Ora vedremo come posare dinamicamente un watermark sull'immagine a dimensione reale.

Abbiamo già creato la pagina *index.php* nella quale, per ogni miniatura, è associato un link che punta sempre alla medesima pagina (*large_w.php*). In questo link passiamo l'id dell'immagine nella querystring. Cosa farà la pagina *large_w.php*:

- Recupera il nome dell'immagine nel database a partire dall'id contenuto nella querystring
- Crea una nuova immagine sulla base dell'immagine selezionata
- Crea una nuova immagine sulla base del watermark che abbiamo scelto
- Fonde le due immagini
- Manda il risultato in output (senza creare fisicamente l'immagine)

Inizia quindi a creare il file *large_w.php*.

Come impostare gli header

Come prima cosa, dobbiamo impostare l'header HTTP indicando che stiamo per mandare in output un'immagine, nel nostro caso le immagini a dimensione reale le manderemo in formato jpg. Dunque, utilizzando la funzione *header()*, scriviamo la prima riga del documento.

Come selezionare la giusta immagine

Per andare a recuperare l'immagine selezionata, dovremo lanciare una query nel database cercando l'id che abbiamo passato nella querystring.

Includiamo quindi il file di connessione, scriviamo la semplice query ed eseguiamola

Ora l'oggetto *\$im* contiene tutte le informazioni sull'immagine.

Dobbiamo ora creare una nuova immagine sulla base di quella selezionata, nel modo che abbiamo già visto, distinguendo tra immagini jpg ed immagini png

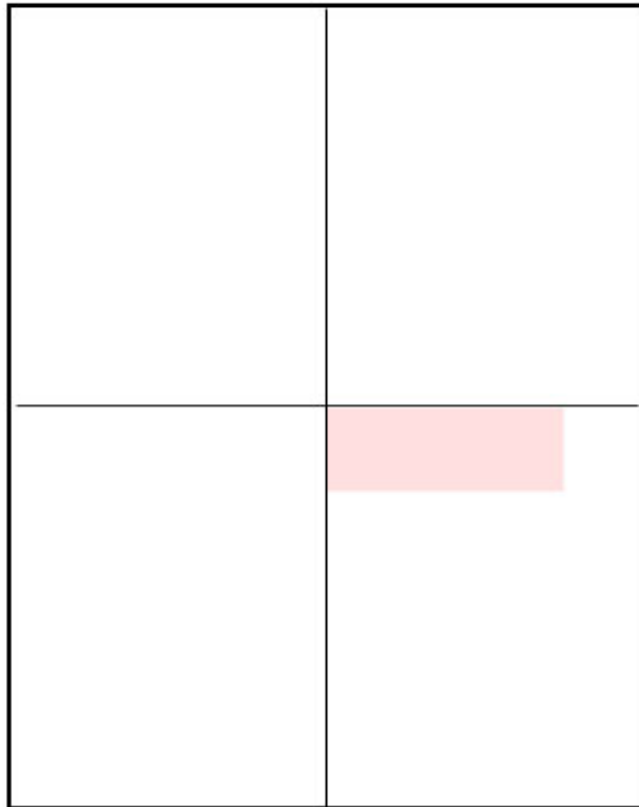
Ora la risorsa *\$image* contiene il riferimento all'immagine selezionata.

Creiamo ora una nuova immagine contenente il watermark. Io ho utilizzato il logo di YIW e l'ho posizionato nella cartella img.

Come puoi vedere nell'[esempio](#), ho posizionato il watermark esattamente al centro dell'immagine.

Come stabilire le coordinate per centrare il watermark.

Per centrare il watermark dovrò innanzitutto conoscere le dimensioni dell'immagine e poi ovviamente dividere per due. Ma poiché le coordinate per il posizionamento si riferiscono al vertice superiore sinistro, il risultato che otterrei sarebbe questo



Per centrarlo correttamente, dovrò sottrarre la metà della larghezza e dell'altezza del watermark, in questo modo

Come sovrapporre il watermark all'immagine

Per questa ultima operazione, utilizzeremo una funzione molto simile a quella che abbiamo considerato per creare i thumbnail (*imagecopyresampled()*) ma molto più specifica, ovvero [imagecopymerge\(\)](#),

Passiamo come primo parametro il riferimento all'immagine principale, come secondo argomento il riferimento all'immagine watermark, in seguito le coordinate del watermark, le coordinate dell'immagine sorgente, la larghezza e l'altezza del watermark ed infine il canale alpha che settiamo a 60 in modo da mostrarlo in trasparenza.

A questo punto *\$image* contiene l'immagine con il watermark. non dobbiamo fare altro che mandarla in output.

Come mandare l'immagine in output

Per mandare l'immagine in output, utilizzeremo la stessa funzione che abbiamo utilizzato nel tutorial precedente per salvare i thumbnail, con l'unica differenza che passeremo un unico parametro, ovvero il riferimento all'immagine.

La funzione *imagejpeg()* in questo caso manderà in output l'immagine, mentre se indico un percorso come secondo parametro la salverà (come abbiamo già visto).

Per finire, è buona abitudine ripulire la memoria inviando il riferimento delle immagini direttamente al garbage collector con l'apposita funzione [imagedestroy\(\)](#).

A dire il vero il garbage collector dovrebbe capire il ciclo di vita di una variabile e liberare la memoria automaticamente. Ma a quanto pare, per le variabili di tipo risorsa, potrebbe non essere così immediato (o almeno così sembra); e trattandosi di risorse piuttosto ingombranti, è meglio andare sul sicuro.

Conclusioni

Il metodo che abbiamo appena visto rispetto a quello di creare le immagini e salvarle ha un grande vantaggio. Essendo le immagini create al momento, non avrò nessun problema qualora decidessi di modificare il watermark. Allo stesso modo, creando i thumbnail al volo, potrò in ogni momento modificare la dimensione senza dover pensare a quelli già salvati con la dimensione precedente. D'altra parte questo metodo utilizza molte più risorse a livello di processore e di memoria. Come sempre l'adozione di un metodo o dell'altro va valutato attentamente in fase di progettazione.

Siamo così giunti alla fine di questa serie di tutorial nei quali abbiamo visto il processo di upload in generale, ed in seguito nello specifico l'upload di immagini con la creazione di thumbnail ed infine la posa di watermark. In questo modo abbiamo potuto dare un'infarinatura della libreria GD.