

# Projektdokumentation

## Watermarky

---

<b>Upload</b>	<b>2</b>
<b>Upload Beschränkung von Dateiformat und Grösse</b>	<b>3</b>
<b>Bildbearbeitung</b>	<b>3</b>
<b>Responsive/Usability</b>	<b>5</b>

## Upload

Form enctype [attribut](#)

WaterMarky.php

```
<!-- UPLOAD -->
    <div class="container p-4 mb-2 bg-light text-dark">
        <h4>Upload pictures</h4>
        <form action="upload.php" method="post"
enctype="multipart/form-data">
            <div class="input-group mb-3">
                <input type="file" name="file"
id="inputGroupFile03">
                <div class="col"></div>
                <button class="btn btn-primary" type="submit"
name="submit" id="upload">Upload</button>
            </div>
        </form>
    </div>
```

upload.php

```
//get uploaded file
$file = $_FILES['file'];

//get file extension
$tmp = explode('.', $file['name']);
$fileExt = strtolower(end($tmp));

//check if file extension is allowed
$allowed = array('jpg', 'png', 'bmp', 'svg');
if(!in_array($fileExt, $allowed))
    return popMsg("Files must be of one of the following types
.jpg/.png/.bmp/.svg You cannot upload files of type ".$fileExt);

//check if an error occurred
if($file['error'] !== 0)
    return popMsg("There was an error uploading your file error
code: ".$file['error']);

//check if file is bigger than 1GB
if($file['size'] > 1000000000)
```

```
        return popMsg("The file is too big! Cannot upload files of  
size larger than 1GB");  
  
        //create unique filename  
        $uniqFileName = explode('.', $file['name'])[0].".".uniqid('',  
true).".".$fileExt;  
  
        //upload  
        if(move_uploaded_file($file['tmp_name'],  
'upload/'.$uniqFileName))  
        {  
            $_SESSION['newFile'] = true;  
            return popMsg("Successfully uploaded");  
        }  
        return popMsg("There was an error uploading your file!");
```

## Upload Beschränkung von Dateiformat und Grösse

In upload.php wird die file extension in eine Variable gespeichert und mit einem Array der erlaubten file extensions verglichen. Sollte die file extension nicht im Array enthalten sein wird eine Fehlermeldung ausgegeben.

```
//get file extension  
$tmp = explode('.', $file['name']);  
$fileExt = strtolower(end($tmp));  
  
//check if file extension is allowed  
$allowed = array('jpg', 'png', 'bmp', 'svg');  
if(!in_array($fileExt, $allowed))  
    return popMsg("Files must be of one of the following types  
.jpg/.png/.bmp/.svg You cannot upload files of type ".$fileExt);
```

## Bildbearbeitung

Neues ImagickDraw objekt

enchant\_pictures.php

```
//create new draw object  
$draw = new ImagickDraw();
```

```
//echo "You have written :".$_POST['text']; // Displaying
Selected Value

// Check if a text was added
// Draw text
if (isset($_POST['text'])) {
    // Font properties
    $draw->setFont('Arial');
    $draw->setFillColor('white');
    $draw->setFontSize( 40 );
    $draw->setStrokeColor('black');
    $draw->setStrokeWidth(1);

    $yText = $img_Height - 20;
    $image->annotateImage($draw, 0, $yText, 0, $_POST['text']);
}

//echo "You have chosen:".$_POST['inputShape']; // Displaying
Selected Value

// Check if a shape was added
// Draw shape
if (isset($_POST['inputShape'])) {
    switch ($_POST['inputShape']) {

        case 'rectangle':
            // rectangle properties
            $draw->setFillColor('yellow');
            $draw->setStrokeColor( new ImagickPixel( 'red' ) );
            $x1Rect = ($watermark_Width * 0.25) + $x;
            $y1Rect = ($watermark_Height * 0.015) + $y;
            // Draw the rectangle
            $draw->rectangle( $x1Rect, $y1Rect, $x1Rect +
100, $y1Rect + 100);
            $image->drawImage($draw);
            break;

        case 'circle':
            // circle properties
```

```
$draw->setFillColor('yellow');  
$draw->setStrokeColor( new ImagickPixel( 'red' ) );  
$xCircle = ($watermark_Width / 2) + $x;  
$yCircle = ($watermark_Height / 2) + $y;  
$rCircle = 100;  
// Draw the circle  
$draw->circle ($xCircle, $yCircle, $xCircle +  
$rCircle, $yCircle);  
$image->drawImage($draw);  
break;  
}  
}  
  
// Draw the watermark on your image  
$image->compositeImage($watermark, Imagick::COMPOSITE_OVER, $x,  
$y);  
  
// save the image  
$image->writeImage(getcwd(). "/enchanted_pics/pic_watermark." .  
$image->getImageFormat());  
$_SESSION['preview_pic'] = "enchanted_pics/pic_watermark." .  
$image->getImageFormat();  
  
return popMsg("Successfully enchanted picture");
```

## Responsive/Usability

Es wurde Bootstrap eingesetzt.

Um die Inputs und Titel der Webseite Responsive darzustellen wurden sie in div's verschachtelt welchen Bootstrap Klassen gegeben wurden:

```
class="form-group"  
class="form-row"  
class="col"
```

Dem Preview Bild wurde ebenfalls solch eine Klasse zugewiesen:

```
class="img-thumbnail"
```

Die Inputs der Webseite wurden klar angeschrieben und es werden Meldungen ausgegeben wenn:

hochzuladende Datei zu gross oder von nicht zugelassenem Dateiformat

```
return popMsg("Files must be of one of the following types  
.jpg/.png/.bmp/.svg You cannot upload files of type ".$fileExt);
```

```
if($file['size'] > 1000000000)
    return popMsg("The file is too big! Cannot upload files of
size larger than 1GB");
```

Datei erfolgreich hochgeladen wurde

```
return popMsg("Successfully uploaded");
```

Der Dowload Button gedrückt wird bevor das Bild erstellt wurde

```
return popMsg("You have to create your picture first before you can
download it! \nPress Do The Image Magick! ;)");
```

Datei erfolgreich heruntergeladen wurde

```
return popMsg("Successfully downloaded");
```

noch kein Bild hochgeladen wurde aber auf den "Do the Image Magick" Button gedrückt wurde

```
return popMsg("You have to upload some Pictures first by clicking
Choose file and Upload :)");
```

Bild erfolgreich bearbeitet wurde

```
return popMsg("Successfully enchanted picture");
```