

```

;; PORTFOLIO. Take an opened file and generate a new image 300dpi 20x30cm
;; and a caption. Useful for polite photo portfolio generation.
;;
;; Copyright (C) 2008 Giannella Davide
;;
;; This program is free software; you can redistribute it and/or modify
;; it under the terms of the GNU General Public License as published by
;; the Free Software Foundation; version 2 of the License,
;;
;; This program is distributed in the hope that it will be useful,
;; but WITHOUT ANY WARRANTY; without even the implied warranty of
;; MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
;; GNU General Public License for more details.
;;
;; You should have received a copy of the GNU General Public License along
;; with this program; if not, write to the Free Software Foundation, Inc.,
;; 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
;;
;; http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt
;;
;; INSTALL
;; =====
;;
;; Create an scm file (ie. portfolio.scm) inside your ~/.gimp-2.2/scripts
;; directory with this code inside it. Restart gimp.
;;
;; USAGE
;; =====
;;
;; From within the image right-click -> Script-Fu -> My Stuff -> Portfolio

(set! pixel-per-cm 118)
(set! PORTRAIT 1)
(set! LANDSCAPE 0)

;;
; parse the file path and return it without extension
;
; @param filename String the file path
; @return the file path without the extension
(define (parse-name filename)
  (set! fparts (strbreakup filename "."))
  (set! y 0)
  (set! out "")
  (set! lngt (- (length fparts) 1))
  (while (< y lngt)
    (set! out (string-append out (nth y fparts)))
    (set! y (+ y 1))
    (if (< y lngt)
        (set! out (string-append out "."))))
  )
  (car (cons out '()))
)

;;
; retrieve the dimensions of the provided drawable and return the scaled width+height
;
; @param img SF-DRAWABLE the drawable
; @param long-side-landscape INT32 the long side for the landscape format
; @param long-side-portrait INT32 the long side for the portrait format
; @return a list with at the first item the width and the last the height (width, height)
(define (get-drw-dimension img long-side-landscape long-side-portrait)
  (let*
    (
      (w (car (gimp-drawable-width img)))
      (h (car (gimp-drawable-height img)))
      (w2 0)
      (h2 0)
    )
    (cond
      ((> w h)
       (set! w2 long-side-landscape)
       (set! h2 (/ (* h long-side-landscape) w))
      )
      ((< w h)
       (set! w2 (/ (* w long-side-portrait) h))
       (set! h2 long-side-portrait)
      )
    )
    (list w2 h2)
  )
)

;;
; scale and move the image layer in order to impaginate it
;
; @param INT32 layer the active layer id
; @param INT32 pixel-per-cm the amount of pixels per cm.

```



```

        -1
        x-portrait
        y-portrait
        text-portrait
        -1
        TRUE
        font-size
        1
        font-face)))
    )
)

(gimp-context-set-foreground original-foreground)
)
)

;;
; take the provided image and create a portfolio layout by that.
; this means 20x30cm page at 300dpi with 1cm margins, space for binding,
; and a sample text for caption.
;
; @param SF-IMAGE img the active image
; @param SF-DRAWABLE drw the active drawable
(define (portfolio img drw)
  (let*
    (
      (fname (parse-name (car (gimp-image-get-filename img))))
      (width-paged 3543)
      (height-paged 2363)
      (dpi 300)
    )
    (set! img-dup (car (gimp-image-duplicate img))) ; duplicate the actual image (no editing on existing file)
    (set! drw-flatten (car (gimp-image-flatten img-dup))) ; flatten the new image (only one layer)
    (set! img-flatten (car (gimp-drawable-get-image drw-flatten)))
    (gimp-image-set-resolution img-flatten dpi dpi) ;ensure to have 300dpi res

    (set! img-paged (car (gimp-image-new width-paged height-paged RGB)))
    (set! layer-background (car (gimp-layer-new img-paged width-paged height-paged RGB-IMAGE "Background" 100 NORMAL-MODE)))
    (gimp-image-set-resolution img-paged dpi dpi)
    (gimp-image-add-layer img-paged layer-background -1)
    (gimp-edit-fill layer-background 2)
    (gimp-display-new img-paged)

    (set! layer-image (car (gimp-layer-new-from-drawable drw-flatten img-paged)))
    (gimp-image-add-layer img-paged layer-image 0)

    (gimp-image-delete img-flatten) ;close the duplicated flattened image

    (scale-and-move-layer layer-image pixel-per-cm)

    (add-caption img-paged (get-drawable-orientation layer-image))

    ;adding guides for margin
    (gimp-image-add-vguide img-paged (* pixel-per-cm 2))
    (gimp-image-add-vguide img-paged (* pixel-per-cm 29))
    (gimp-image-add-hguide img-paged (* pixel-per-cm 1))
    (gimp-image-add-hguide img-paged (* pixel-per-cm 19))

    (gimp-image-set-filename img-paged (string-append (parse-name (car (gimp-image-get-filename img))) "-paged.xcf"))

    (gimp-displays-flush)
  )
)

(script-fu-register "portfolio"
  "<Image>/Script-Fu/My-Stuff/Portfolio"
  "Take the active image and impaginate it for portfolio"
  "Davide Giannella"
  "(C) 2008 Davide Giannella"
  "May 2008"
  ""
  SF-IMAGE "Image" 0
  SF-DRAWABLE "Drawable" 0
)

```