BASH: scripts

Script writing, in simple terms, consists of gathering commands in a file that we would normally type on the command line. We can then mark such a file as executable with the command `chmod u+x file` and execute with the command `./file`. The command line (BASH) is used to run **programs** - therefore: **each line of the script looks like this:** `program arguments`.

Analyze the following code snippet:

```
i=1
while test $i -lt 10
do
  echo $i
  cp file file_$i
  i=$(expr $i + 1)
done
```

It's easy to see that:

- in assigning the variable `i=1` we do not use spaces, because the `bash` shell knows then, that it is an assignment. `i = 1` would mean a call to the `i` program with the `=` and `1` arguments.
- in the expression `expr $i + 1`, we must keep spaces so that the program `expr` gets three arguments `$i`, `+`and `1`, not one `$i+1`.
- in the `while` loop, we cannot write `i<10`, but we must use some program. The program `test` was created for all kinds of tests. In this case, we run it with the arguments `$i`, `-lt` and `10`, where the option `-lt` means 'less than'.

Useful programs

If we already know that each script in the `bash` shell is a series of calls to programs, we need a lot of small programs from which we can create scripts.

Examples:

- `echo text` — Prints text to the screen.
- `cat file` — Prints the contents of the file to the screen.
- `grep text` — Reads characters from the input and prints only lines containing "text".
- `grep text files` — Search for "text" in files.
- `cd directory` — Enters the directory.
- `ls directory` — Prints the contents of the directory.
- `cp files directory/` — Copy files to the directory.
- `cp file1 file2` — Copy the file named "file1" to the file named "file2".
- `mv files directory/` — Moves files to the directory.
- `mv file1 file2` — Changes the file name from "file1" to "file2".
- `sed 's/text1/text2/g'` — Reads the text from the input and prints it, replacing "text1" with "text2".
- `cut -d " " -f 1` — Reads text from the input, separates it into fields using a space as the separator (`-d " "`) and prints the first field (`-f 1`).
- `seq number1 number2` — Prints on the screen a sequence of numbers from 'number1' to 'number2'.

Redirection of exit input

By default, all programs read from the keyboard and write on the screen. However, both input and output can be redirected.

- `program >file` — What the program would write to the screen will be written to the file (the file will be overwritten if it exists).
- `program >>file` — What the program would write to the screen will be added to the file (the file will be created if it did not exist).
- `program <file` — The program will get the contents of the file as if we entered it from the keyboard.
- `program1 | program2` — What 'program1' would write to the screen will be entered 'from the keyboard' into 'program2'.
- `` `program` `` or `$(program)` — What the program would write to the screen will be **pasted here in the code** (see examples). *We strongly recommend the second version* The inverted apostrophe `` ` `` is on the key with the tilde `~`.

Examples:

- `echo Text >file` — will write 'Text' to the file (the file will be overwritten if it exists)
- `echo Text >>file` — will add 'Text' to the file (the file will be created if it did not exist)
- `grep Text <file` — will search for lines containing 'Text' and print them on the screen

- `echo Text | sed 's/x/y/g'` — Replace every occurrence of 'x' with 'y' in 'Text'. So the command will write "Teyt" on the screen.
- `echo $name | sed 's/.txt/.dat/g'` — Replaces `.txt` with `.dat` in the variable `name`. The result will write to the screen.
- `name2=$(echo $name | sed 's/.txt/.dat/g')` — As before, but the result will be written to the variable `name2`.
- `ls directory >file` — will write the contents of the directory to the file (the file will be overwritten if it exists).
- `cp `ls` directory` or `cp $(ls) directory` — will copy files to the `directory` according to the list returned by `ls`.
- `cp `cat file` directory` or `cp $(cat file) directory` — will copy files to the `directory` according to the list contained in the file.

Loops and conditional expressions

conditional statement if

```
if program arguments
then
    commands1
else
    commands2
fi
```

If the `program arguments` command is successful (the program returns 0), *command1* will be executed. Otherwise *commands2* will be executed.

while loop

```
while program arguments
do
    commands
done
```

A loop that will execute *commands* while the `program arguments` is successfully executed (until it fails).

for loop

```
for i in list
do
    commands
done
```

A loop that in turn inserts each item from the list into the variable `i` and then executes *commands*.

For example:

```
for i in *.jpg
do
    mv $i IMG/a_$i
done
```

It will transfer any file with the extension `.jpg`, adding the prefix `a_` to the `IMG` directory (e.g. `picture.jpg` will change into `IMG/a_obrazek.jpg`).

Image processing

Program `convert`

- `convert file.gif file.jpg` - convert a file in GIF format to JPG
- `convert file1.jpg -resize 50% file2.jpg` - will reduce the image twice
- `convert file1.jpg -resize 100 file2.jpg` - will reduce the image so that the shorter dimension is 100 pixels
- `convert file1.jpg -resize 100x100 file2.jpg` - will reduce the image to fit in a square of 100 by 100 pixels
- `convert file1.jpg -resize 100x100n! file2.jpg` - will reduce the image to 100 by 100 pixels
- `convert -size 320x85 canvas:none -font Bookman-DemiItalic -pointsize 72 -` - will create a picture fuzzy-magick.jpg, with the text "Magick" marked

All options for `convert`

- `-adaptive-blur geometry` - adaptively blur pixels; decrease effect near edges
- `-adaptive-resize geometry` - adaptively resize image with data dependent triangulation.

- `-adaptive-sharpen geometry` - adaptively sharpen pixels; increase effect near edges
- `-adjoin` - join images into a single multi-image file
- `-affine matrix` - affine transform matrix
- `-alpha` - on, activate, off, deactivate, set, opaque, copy, transparent, extract, background, or shape the alpha channel
- `-annotate geometry text` - annotate the image with text
- `-antialias` - remove pixel-aliasing
- `-append` - append an image sequence
- `-authenticate value` - decipher image with this password
- `-auto-gamma` - automagically adjust gamma level of image
- `-auto-level` - automagically adjust color levels of image
- `-auto-orient` - automagically orient image
- `-background color` - background color
- `-bench iterations` - measure performance
- `-bias value` - add bias when convolving an image
- `-black-threshold value` - force all pixels below the threshold into black
- `-blue-primary point` - chromaticity blue primary point
- `-blue-shift factor` - simulate a scene at nighttime in the moonlight
- `-blur geometry` - reduce image noise and reduce detail levels
- `-border geometry` - surround image with a border of color
- `-bordercolor color` - border color
- `-brightness-contrast geometry` - improve brightness / contrast of the image
- `-caption string` - assign a caption to an image
- `-cdl filename` - color correct with a color decision list
- `-channel type` - apply option to select image channels
- `-charcoal radius` - simulate a charcoal drawing
- `-chop geometry` - remove pixels from the image interior
- `-clamp` - restrict colors from 0 to the quantum depth
- `-clip` - clip along the first path from the 8BIM profile
- `-clip-mask filename` - associate clip mask with the image
- `-clip-path id` - clip along a named path from the 8BIM profile
- `-clone index` - clone an image
- `-clut` - apply a color lookup table to the image
- `-contrast-stretch geometry` - improve the contrast in an image by 'stretching' the range of intensity value
- `-coalesce` - merge a sequence of images
- `-colorize value` - colorize the image with the fill color
- `-color-matrix matrix` - apply color correction to the image.
- `-colors value` - preferred number of colors in the image
- `-colorspace type` - set image colorspace
- `-combine` - combine a sequence of images
- `-comment string` - annotate image with comment
- `-compose operator` - set image composite operator
- `-composite` - composite image
- `-compress type` - image compression type
- `-contrast` - enhance or reduce the image contrast
- `-convolve coefficients` - apply a convolution kernel to the image
- `-crop geometry` - crop the image
- `-cycle amount` - cycle the image colormap
- `-decipher filename` - convert cipher pixels to plain
- `-debug events` - display copious debugging information
- `-define format:option` - define one or more image format options
- `-deconstruct` - break down an image sequence into constituent parts
- `-delay value` - display the next image after pausing
- `-delete index` - delete the image from the image sequence
- `-density geometry` - horizontal and vertical density of the image
- `-depth value` - image depth
- `-despeckle` - reduce the speckles within an image
- `-direction type` - render text right-to-left or left-to-right
- `-display server` - get image or font from this X server
- `-dispose method` - layer disposal method
- `-distort type coefficients` - distort image
- `-dither method` - apply error diffusion to image
- `-draw string` - annotate the image with a graphic primitive
- `-duplicate count,indexes` - duplicate an image one or more times
- `-edge radius` - apply a filter to detect edges in the image
- `-emboss radius` - emboss an image
- `-encipher filename` - convert plain pixels to cipher pixels
- `-encoding type` - text encoding type
- `-endian type` - endianness (MSB or LSB) of the image
- `-enhance` - apply a digital filter to enhance a noisy image
- `-equalize` - perform histogram equalization to an image
- `-evaluate operator value` - evaluate an arithmetic, relational, or logical expression
- `-evaluate-sequence operator` - evaluate an arithmetic, relational, or logical expression for an image sequence
- `-extent geometry` - set the image size
- `-extract geometry` - extract area from image

- `-family name` - render text with this font family
- `-fft` - implments the discrete Fourier transform (DFT)
- `-fill color` - color to use when filling a graphic primitive
- `-filter type` - use this filter when resizing an image
- `-flatten` - flatten a sequence of images
- `-flip` - flip image in the vertical direction
- `-floodfill geometry color` - floodfill the image with color
- `-flop` - flop image in the horizontal direction
- `-font name` - render text with this font
- `-format string` - output formatted image characteristics
- `-frame geometry` - surround image with an ornamental border
- `-function name` - apply a function to the image
- `-fuzz distance` - colors within this distance are considered equal
- `-fx expression` - apply mathematical expression to an image channel (s)
- `-gamma value` - level of gamma correction
- `-gaussian-blur geometry` - reduce image noise and reduce detail levels
- `-geometry geometry` - preferred size or location of the image
- `-gravity type` - horizontal and vertical text placement
- `-green-primary point` - chromaticity green primary point
- `-help` - print program options
- `-identify` - identify the format and characteristics of the image
- `-ift` - implements the inverse discrete Fourier transform (DFT)
- `-implode amount` - implode image pixels about the center
- `-insert index` - insert last image into the image sequence
- `-intent type` - type of rendering intent when managing the image color
- `-interlace type` - type of image interlacing scheme
- `-interline-spacing value` - the space between two text lines
- `-interpolate method` - pixel color interpolation method
- `-interword-spacing value` - the space between two words
- `-kerning value` - the space between two characters
- `-label string` - assign a label to an image
- `-lat geometry` - local adaptive thresholding
- `-layers method` - optimize or compare image layers
- `-level value` - adjust the level of image contrast
- `-limit type value` - pixel cache resource limit
- `-linear-stretch geometry` - linear with saturation histogram stretch
- `-liquid-rescale geometry` - rescale image with seam-carving
- `-log format` - format of debugging information
- `-loop iterations` - add Netscape loop extension to your GIF animation
- `-mask filename` - associate a mask with the image
- `-mattecolor color` - frame color
- `-median radius` - apply a median filter to the image
- `-mode radius` - make each pixel the 'predominant color' of the neighborhood
- `-modulate value` - vary the brightness, saturation, and hue
- `-monitor` - monitor progress
- `-monochrome` - transform image to black and white
- `-morph value` - morph an image sequence
- `-morphology method kernel` - apply a morphology method to the image
- `-motion-blur geometry` - simulate motion blur
- `-negate` - replace each pixel with its complementary color
- `-noise radius` - add or reduce noise in an image
- `-normalize` - transform image to span the full range of colors
- `-opaque color` - change this color to the fill color
- `-ordered-dither NxN` - ordered dither the image
- `-orient type` - image orientation
- `-page geometry` - size and location of an image canvas (setting)
- `-paint radius` - simulate an oil painting
- `-ping` - efficiently determine image attributes
- `-pointsize value` - font point size
- `-polaroid angle` - simulate a Polaroid picture
- `-posterize levels` - reduce the image to a limited number of color levels
- `-precision value` - set the maximum number of significant digits to be printed
- `-preview type` - image preview type
- `-print string` - interpret string and print to console
- `-process image-filter` - process the image with a custom image filter
- `-profile filename` - add, delete, or apply an image profile
- `-quality value` - JPEG/MIFF/PNG compression level
- `-quantize colorspace` - reduce image colors in this colorspace
- `-quiet` - suppress all warning messages
- `-radial-blur angle` - radial blur the image
- `-raise value` - lighten/darken image edges to create a 3-D effect
- `-random-threshold low,high` - random threshold the image
- `-red-primary point` - chromaticity red primary point
- `-regard-warnings` - pay attention to warning messages.
- `-region geometry` - apply options to a portion of the image
- `-remap filename` - transform image colors to match this set of colors
- `-render` - render vector graphics
- `-repage geometry` - size and location of an image canvas
- `-resample geometry` - change the resolution of an image

- `-resize geometry` - resize the image
- `-respect-parentheses` - settings remain in effect until parenthesis boundary.
- `-roll geometry` - roll an image vertically or horizontally
- `-rotate degrees` - apply Paeth rotation to the image
- `-sample geometry` - scale image with pixel sampling
- `-sampling-factor geometry` - horizontal and vertical sampling factor
- `-scale geometry` - scale the image
- `-scene value` - image scene number
- `-seed value` - seed a new sequence of pseudo-random numbers
- `-segment values` - segment an image
- `-selective-blur geometry` - selectively blur pixels within a contrast threshold
- `-separate` - separate an image channel into a grayscale image
- `-sepia-tone threshold` - simulate a sepia-toned photo
- `-set attribute value` - set an image attribute
- `-shade degrees` - shade the image using a distant light source
- `-shadow geometry` - simulate an image shadow
- `-sharpen geometry` - sharpen the image
- `-shave geometry` - shave pixels from the image edges
- `-shear geometry` - slide one edge of the image along the X or Y axis
- `-sigmoidal-contrast geometry` - increase the contrast without saturating highlights or shadows
- `-smush offset` - smush an image sequence together
- `-size geometry` - width and height of image
- `-sketch geometry` - simulate a pencil sketch
- `-solarize threshold` - negate all pixels above the threshold level
- `-splice geometry` - splice the background color into the image
- `-spread radius` - displace image pixels by a random amount
- `-statistic type geometry` - replace each pixel with corresponding statistic from the neighborhood
- `-strip` - strip image of all profiles and comments
- `-stroke color` - graphic primitive stroke color
- `-strokewidth value` - graphic primitive stroke width
- `-stretch type` - render text with this font stretch
- `-style type` - render text with this font style
- `-swap indexes` - swap two images in the image sequence
- `-swirl degrees` - swirl image pixels about the center
- `-synchronize` - synchronize image to storage device
- `-taint` - mark the image as modified
- `-texture filename` - name of texture to tile onto the image background
- `-threshold value` - threshold the image
- `-thumbnail geometry` - create a thumbnail of the image
- `-tile filename` - tile image when filling a graphic primitive
- `-tile-offset geometry` - set the image tile offset
- `-tint value` - tint the image with the fill color
- `-transform` - affine transform image
- `-transparent color` - make this color transparent within the image
- `-transparent-color color` - transparent color
- `-transpose` - flip image in the vertical direction and rotate 90 degrees
- `-transverse` - flop image in the horizontal direction and rotate 270 degrees
- `-treedepth value` - color tree depth
- `-trim` - trim image edges
- `-type type` - image type
- `-undercolor color` - annotation bounding box color
- `-unique-colors` - discard all but one of any pixel color.
- `-units type` - the units of image resolution
- `-unsharp geometry` - sharpen the image
- `-verbose` - print detailed information about the image
- `-version` - print version information
- `-view` - FlashPix viewing transforms
- `-vignette geometry` - soften the edges of the image in vignette style
- `-virtual-pixel method` - access method for pixels outside the boundaries of the image
- `-wave geometry` - alter an image along a sine wave
- `-weight type` - render text with this font weight
- `-white-point point` - chromaticity white point
- `-white-threshold value` - force all pixels above the threshold into white
- `-write filename` - write images to this file