

Help

```
help commandName
commandName /?
help --> show all system commands
```

All System Commands

```
help
```

Clear Screen

```
Clear Screen          cls
```

Shortcuts

```
Tab                Autocomplete
arrow keys         back/forward history
up/down            commands
Control + C        stops current command
ALT + F7           clears the history
F7                 shows the history
```

Directory Make & Remove

```
mkdir              creates new subdirectory
subdir-            directory
mkdir              creates subdir & subdir/su-
subdir/su-         bsubdir
bsubdir            Hint: creates it, if they aren't
                    existing
rmdir              removes an empty dir
rmdir /s           removes directory with
directory          content
rmdir /s /q        removes directory with
directory          content & skips y/n question
```

Directory copy

```
copy src           copy files
dst
copy /Y src        copy files overwrites without
dest              quesiton
xcopy /s           copies dirs and subdirs
src dest           except empty ones
```

Directory copy (cont)

```
xcopy /YS src      as above without ask to
dest              overwrite
xcopy /YS src      copy to network share
\\server\share
robocopy /S src    copies dirs and subdirs
dest              except empty ones
robocopy /YS       as above without ask to
src dest           overwrite
```

Parameters

```
%0                Name of Script
%1 ... %9          Parameter 1 ... Parameter
                    9
%*                prints all parameters as a
                    string
(one line)       access it through a for loop
for %%x in         works with > 10
(%) do echo        parameters
%%x
```

Pass Parameters to sub script

```
parameters_scri-  @echo off
pt.bat            call parameters_subs-
                  cript.bat %*
                  EXIT /B %errorlevel%
parameters_s-     @ECHO off
ubscript.bat      echo subscript called
                  with %*
                  EXIT /B 0
```

```
EXIT /B specifies to exit the current batch
script instead of cmd.exe. If executed from
outside a batch script, it will quit cmd.exe
```

Arrays Way2

```
set               set value to a specific array
array[0]=0        element
set
array[1]=1
...
echo              access to array elements
%array[0]%
echo
%array[1]%
```

Arrays Way2 (cont)

```
missing loop over elements: indexed
```

Functions: label and goto

```
echo "Hi:"
goto :good
:bad
echo "I don't like scripting"
goto :eof
:good
echo "I like scripting"
goto :eof
```

DoubleClickExecution

```
@ECHO off
setlocal
REM check for interactive sesseion
SET interactive=0
ECHO %CMD% | FINDSTR /L
%COMSPEC% >NUL 2>&1
IF %ERRORLEVEL% EQ SET intera-
ctive=1
ECHO do work
REM pause if interactive (double-clicked
execution)
IF "%interactive%" == "0" PAUSE
EXIT /B 0
```

```
FINDSTR /L is use search string literally
```

Navigation

```
realative         parent          cd .., cd..
                  directory
                  subdirectory    cd subdir-
                                ectory
absolute          cd
                  C:\Windows
                  cd \Windows
change            Z:
drive
```

Wildcards

?	one arbitrary character (file?.txt)
*	arbitrary characters (0,1 or more)

File Show

type file	show file content
more	show file content pagewise
sort file	show file content sorted

Directory show Content

current folder	dir
absolute path	dir C:\Windows
dir Y:	different drive
dir /B	only show file names and directory names

Batch Files User Input

@ echo off	Syntax: set /P
set /P name="Input	var=prompt:
yor name: "	
echo Hi %name%	variable call

Batch Files File Input

@echo off	
echo Bill > file	
echo Gate >> file	
set /P filecontent =<	set /P var=< file
file	
echo First Line:	
%filecontent%	
FOR /f %%f in (file)	REM print each
Do (	line in file
echo %%f	
)	

Variables

setlocal	Only active in the current batch file (local)
set	
MY_ENV=abc	
endlocal	
set	creates or changes an environment variable that is active in the cmd. (global)
MY_ENV=abc	
set	shows the active environment variables
echo	print variable out
%USERNAME%	
defined	returns
MY_VAR	0 => Execution was successful
	>0 => Execution failed

IF Control Structures

IF test (command) ELSE	Syntax
(command)	

Control structures For

for {%% }%<Variable> in (<Set>) do <Command> [<CommandLineOptions>]
cmd.exe use ->%
batch script use ->%%

Functions

call :print_name Florian
echo print_name existst with %errorlevel%
goto :eof
:prinname
echo Hit %1
exit /B 0 <- REM sets exit status of :print_name

File Rename & Move

rename file1	rename file1 to file2
file2	
move file1 dir	move file 1 to dir (path dir/file1)

File Rename & Move (cont)

move	moves and renames file1 to file2
file1	(path: dir/file2) file2 is overwritten
dir/file2	from file1
move	moves and overwrites file1 to file2
/Y file1	without asking
file2	

File Names

Case-Insensitive

Max. 260 characters incl. invisible terminating null character

spaces and character set 128 - 255 allowed

Never <, >, :, ", /, \, |, ?, *

Hint: A-Z, a-z, 0-9, _, spaces, (underscore as space)

*.bat & *.cmd

dir /B --> only show file names and directory Names

Directory Rename & Move

rename	renames dir1 in dir2
dir1	
dir2	
move	if dir2 exist: dir1 into dir2 (path: dir2/dir1), else rename dir1 to dir2
dir1	
dir2	
move	file 2 is deleted. file1 get the name of file2 ans is set at the position of file2 with asking (Yes/No/All)
file1	
dir/file2	
move	file 2 is deleted. file1 get the name of file2 ans is set at the position of file2 without asking.
/Y file1	
file2	

File Redirects

type file.txt sort	give output of type to sort
sort < file.txt	give content (stdin) of file.txt
echo "hi" > file.txt	write "hi" (stout) to file.txt
type NUL > file.txt	create empty file ▲
rm file 2> errors.txt	redirects errors(stderr) ro errors.txt
rm file 2> NUL	discard errors
rm file 2>&1	redirect stderr to stdout
echo "you" >> file.txt	append "you" to file.txt

Comment

REM	Präfix after them the text
-----	----------------------------

Calculation

set /a i=0
echo i=%i%
set /a i=1+1 Operatoren +, -, *, /
set /a i+=1 Operatoren +, -, *, /

Arrays Way1

set array=1 2 3 4	
%array%	(echo) all Array elements
for %%a in (%array%) do (echo%%a)	loop over elements
set array=(and nothing else)	unset variable or array

Control Structures: Comparison Operators

Operator	Info	Equal
EQU	==	Equal
NEQ	!=	Not equal
LSS	<	Less
LEQ	<=	Less than or equal to
GTR	>	Greater than
GEQ	>=	Greater than or equal
NOT	!	Not

Editors

Notepad
Notepad++
VisualStudioCode

Editors

Notepad
Notepad++
VisualStudioCode
Atom

Terminals

cmd.exe
Windows Terminal
ConEmu
cmdr

Batch Filex Exit / Return

Specify exit code:	echo "abc" exit 0
Query error level	echo %ERRORLEVEL%
Exit/Return codes	0 => Execution was successful >0 => Execution fiales with code X