|  |  |
| --- | --- |
| **Commando's** | |
| Cp | Copy |
| Dd | Convert file + copy |
| Ln | Make link |
| Mktemp | Make temp file/dir |
| Touch | Make new file if it doesn't exist |
| Cat | Concat files and print >1 |
| Head/tail n | Prints first/last n lines |
| uniq | unieke dingen in een bestand vinden |

|  |  |
| --- | --- |
| **Random syntax** | |
| rm -- -rf | -- zorgt ervoor dat hetgeen erachter als naam beschouwd wordt |
| echo {1,2,3}{0..9} | echo'd alle combinaties 1/2/3 met 0->9 |
| {00..29} | zet er alle nullen voor indien enkelcijfer-getal |
| **Rekenkundige bewerkingen** | |
| z=$((x\*y)) | |
| let z=x\*y | |

|  |  |
| --- | --- |
| **Arrays** | |
| data=( 0 1 2 3 4 ) |  |
| data[20]=10 |  |
| ${data[@]} | all values |
| ${#data[@]} | size |
| ${data[n]} | n'th value |
| data=( $(wc ..) ) | fills array with output from command, split up bij IFS |
| unset data[1] | deletes arrayelement 1 |

|  |  |
| --- | --- |
| **In/Output shizzle** | |
| 0 | std input |
| 1 | std output |
| 2 | std error |
| > | Redirect output to <file> (overwrites) |
| >> | Add output to <file> |
| > /dev/null | Trash |

|  |  |
| --- | --- |
| **Filter shizzle** | |
| Grep | grep "BEGIN\|END\|IF\|ENDIF" bestandsnaam Basic |
|  | grep -E "BEGIN|END|IF|ENDIF" bestandsnaam  Extended |
|  | grep -F "BEGIN<RET>END<RET>IF<RET>ENDIF" bestandsnaam |
| sort |  |
| join |  |
| cut -d ':' -f 1 | Splits op delimiter ':', print eerste deel.  optie -s: enkel de lijnen met een delimiter worden beschouwd |
| find |  |

|  |  |  |
| --- | --- | --- |
| **Script shizzle** | | |
| $var ~ ${var} | | inhoud variabele |
| $(function) | | returns value van de functie |
| PATH=$PATH:<workdir> | | laat toe om script elders uit te voeren |
| read var1 rest < file | | Splitst file op volgens IFS en slaat op in vars + de overschot van file gaat in rest |
| **Script-argumenten** | | |
| $1, $2.., ${13} |  | |
| ${!#} | laatste arg | |
| $# | Aantal args | |
| $\* of $@ | alle args | |
| $? | exitstatus | |
| $$ | PID | |
| set -- $(ls -a) | stel alle vars in met return values van de functie | |
| Shift n | alle vars n plaatsen naar links, wat de eerste n verwijdert | |
| **Script uitvoeren** | | |
| bash -x <script> | beschrijft hoe elk deelcommando geïnterpreteerd wordt | |
| exit 1/0 | exitcode aan je script geven | |

|  |  |
| --- | --- |
| **Parameter expansion** | |
| ${var-ding} | If var defined => return var. Else return ding |
| ${var=ding} | If var undefined => var = ding. Returns var |
| ${var?error} | Returns var unless undefined. Prints error if undefined |
| ${var+ding} | If var defined => return ding, else return null |

|  |  |
| --- | --- |
| **Testen** | |
| [[ voorwaarde ]] | {==, =>,..} -> stringvgl. {-eq, -lt,..} -> numerieke vgl |
| (( voorwaarde )) | {==, ++, <=,..} -> numerieke vgl |
| Info test | all values |
| Kortere if | [[ voorwaarde ]] &&/|| { opdracht1 ; opdracht2 ; .. ; } |

|  |  |
| --- | --- |
| **Bash v1-compatibele manipulaties** | |
| ${#*variabele*} | geeft de lengte van de waarde van de variabele in karakters |
| ${*variabele*#*pattern*} | verwijdert de **kortst** mogelijke substring vooraan de*variabele* die aan het *pattern* voldoet |
| ${*variabele*##*pattern*} | verwijdert de **langst** mogelijke substring vooraan de*variabele* die aan het *pattern* voldoet |
| ${*variabele*%*pattern*} | verwijdert de **kortst** mogelijke substring achteraan de*variabele* die aan het *pattern* voldoet |
| ${*variabele*%%*pattern*} | verwijdert de **langst** mogelijke substring achteraan de*variabele* die aan het *pattern* voldoet |
| **Vereisen minimaal Bash v2** | |
| ${*variabele*:*offset*} | geeft de deelstring vanaf positie *offset*. |
| ${*variabele*:*offset*:*length*} | geeft de deelstring vanaf positie *offset* met *length*tekens. |
| ${*variabele*/*pattern*/*string*} | geeft de string die ontstaat als je in de *variabele* de eerste en langste substring die aan *pattern* voldoet, vervangt door *string*. |
| ${*variabele*//*pattern*/*string*} | geeft de string die ontstaat als je in de *variabele* alle substrings die aan *pattern* voldoen, vervangt door *string*. |
| ${*variabele*/#*pattern*/*string*} | geeft de string die ontstaat als je de langste substring vooraan de *variabele*, die aan *pattern* voldoet, vervangt door *string*. |
| ${*variabele*/%*pattern*/*string*} | geeft de string die ontstaat als je de langste substring achteraan de *variabele*, die aan *pattern* voldoet, vervangt door *string*. |
| **Vereisen minimaal Bash v4** | |
| ${*variabele^*} | converteert de eerste letter van de string naar *uppercase* |
| ${*variabele,*} | converteert de eerste letter van de string naar *lowercase* |
| ${*variabele^^*} | converteert de volledige string naar *uppercase* |
| ${*variabele,,*} | converteert de volledige string naar *lowercase* |