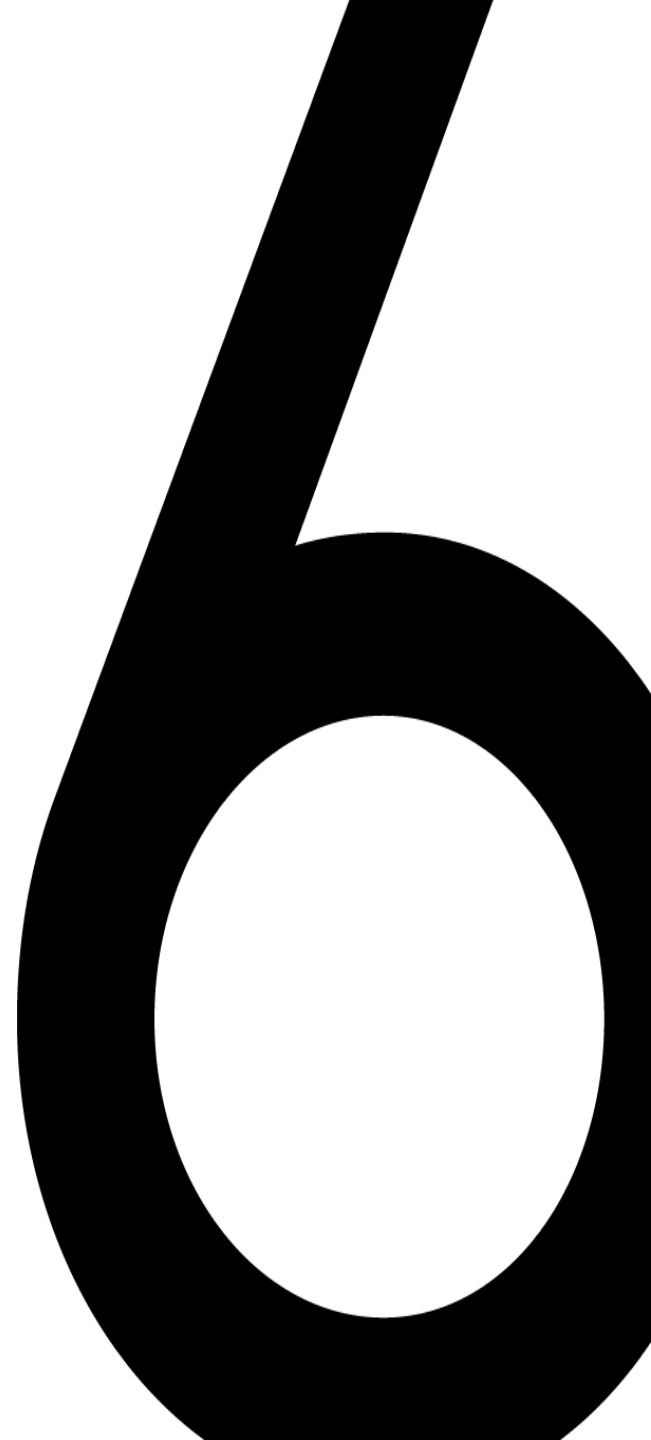


---

# Bash Regex



---

# REGEX

- soorten
- expressies
- =~
- groups
- extglob

# Soorten regex

---

## ■ POSIX2.0 Regex

- ◆ Extended Regex (ERE)

- `egrep/bash`

- ◆ Oud: Basic Regex(BRE)

- escapen van ? + { | ( en )
- standaard grep (zie CS1...)

## ■ Perl Compatible Regex (PCRE)

- UTF-8 en UNICODE support
- perl/python/java/javascript/c#
- grep -P

# **Extended en basic regex voorbeeld**

## ■ Extended regex met grep (egrep):

- ◆ `grep -E "groen|rood" bestand.txt`

## ■ Basic regex met grep

- ◆ `grep 'groen \| rood' bestand.txt`

# Enkele regex expressies

---

- . 1 karakter
- [a-z] karakter a tot en met z in kleine letters
- [0-9] getal 0 tot en met 9
- [ab] karakter a of karakter b
- a{4} vier keer a
- a{1,4} van één tot 4 keer a
- + 1 tot n keer
- \* 0 tot n keer
- ^ begint met
- \$ eindigt met

# **POSIX: Soorten karakters extended regex**

- `[[:digit:]]`      een cijfer
- `[[:space:]]`      spatie, tab, newl, return
- `[[:alnum:]]`      letters en cijfers
- `[[:alpha:]]`      letters
- `[[:blank:]]`      spatie en tab
- `[[:lower:]]`      lowercase letters
- `[[:print:]]`      afdrukbare karakters

# **=~ operator**

```
#!/bin/bash
```

```
content="Karel de Grote-Hogeschool, Nationalestraat 5,B-  
2000 Antwerpen"
```

```
regex="B-[0-9]{4}"
```

```
if [[ $content =~ $regex ]]
```

```
then
```

```
    echo "postnummer gevonden" ; exit 0
```

```
else
```

```
    echo "postnummer niet gevonden" >&2 ; exit 1
```

```
fi
```

OPGELET: \$regex is NOOIT met dubbele quotes!

# **`${BASH_REMATCH[0]}`**

---

- `${BASH_REMATCH[0]}` is een array met het resultaat van de match. Element 0 is de hele match

```
#!/bin/bash
```

```
# Functie: Regex voor een postnummer
```

```
content="Karel de Grote-Hogeschool, Nationalestraat 5,B-  
2000 Antwerpen"
```

```
regex="B-[0-9]{4}"
```

```
[[ $content =~ $regex ]]
```

```
echo "${BASH_REMATCH[0]}"
```

Output: B-2000



# regex voorbeeld: email adres

*jan . celis @ kdg . be*  
[[[:alnum:]]+ \. [[[:alnum:]]+ @ [[[:alnum:]]+ \. [[[:alpha:]]+]

Het eerste stuk kan ook zonder punt zijn, dus dat plaatsen we optioneel:

*jancelis @ kdg . be*  
([[[:alnum:]]+ \.){0,2} [[[:alnum:]]+ @ [[[:alnum:]]+ \. [[[:alpha:]]+]

Het tweede deel kan ook uit subdomeinen bestaan:

*jan . celis @ student . kdg . be*  
([[[:alnum:]]+ \.){0,2} [[[:alnum:]]+ @ ([[[:alnum:]]+ \.){1,3} [[[:alpha:]]+]

Het laatste domein deel is van 2 tot 3 karakters (zonder cijfers):

*jan . celis @ student . kdg . be*  
([[[:alnum:]]+ \.){0,2} [[[:alnum:]]+ @ ([[[:alnum:]]+ \.){1,3} [[[:alpha:]]{2,3}]

# regex groups

---

- Een group selecteert met haakjes een onderdeel van de regex
- Het resultaat zit in array `n` `${BASH_REMATCH[n]}`

```
content="Karel de Grote-Hogeschool, Nationalestraat 5,B-2000  
Antwerpen"
```

```
regex="([- a-zA-Z]+), ([[:alpha:]]+) ([[:digit:]]+),(.*) (.*)"
```

```
[[ $content =~ $regex ]]
```

```
echo "Naam: ${BASH_REMATCH[1]}"
```

```
echo "Straat: ${BASH_REMATCH[2]} Nr: ${BASH_REMATCH[3]}"
```

```
echo "Postcode: ${BASH_REMATCH[4]} Stad: ${BASH_REMATCH[5]}"
```

Output:

Naam: Karel de Grote-Hogeschool

Straat: Nationalestraat Nr: 5

Postcode: B-2000 Stad: Antwerpen

# Regex met parameter substitution

```
#!/bin/bash
```

```
content=" Karel de Grote-Hogeschool, Nationalestraat 5,B-2000  
Antwerpen"
```

```
shopt -q -s extglob
```

```
content2=${content#*+([[:digit:]])}  
content=${content##*+([[:digit:]])}
```

```
shopt -q -u extglob
```

```
echo "ungreedy: $content2"
```

```
echo "greedy: $content"
```

Output:

```
ungreedy: ,B-2000 Antwerpen
```

```
greedy: Antwerpen
```