

OEF REGEX

1. help

Schrijf het script `backupregex.sh`. Uitgaande van het script `backup.sh` (HFST3) implementeer je ook een `help` functie.
Gebruik een regex van `$@` om een `--help` of een `-h` argument te detecteren.
Als dat het geval is, toont het script een help tekst en het stopt.

2. paswoord complexiteit

Schrijf het script `checkpassword.sh` dat aan de gebruiker een paswoord vraagt.

Met een regex kijk je na of het paswoord voldoet aan de volgende voorwaarden:

- beginnen met een hoofdletter
- minimum 8 karakters lang zijn
- eindigen met minstens twee getallen

Het script geeft als output. "Paswoord geldig" of "Paswoord ongeldig"

3. agrep

DOEL: Bij een google search mag je een afwijking ingeven
Als je bv. op "gekent" zoekt, geeft hij "Did you mean gekend?"
Gebruik `agrep` om dezelfde soort afwijkingen toe te laten.

NODIG: `sudo apt install agrep dictionaries-common`

Schrijf het script `agrep.sh` dat als eerste argument een woord verwacht.
Dit woord zoek je met `agrep` op in het woordenboek `/usr/share/dict/words`
Als je bij `agrep` de optie `-1` gebruikt staat die 1 afwijking toe.
Afwijkingen kunnen foute characters zijn maar ook langere of kortere woorden.

Als het woord in het woordenboek staat, toon je het woord met de melding "FOUND"

Als het woord niet in de woordenboek staat geef je "Did you mean:" en de antwoorden die `agrep` geeft

VOORBEELD:

```
./agrep.sh proffessor
Did you mean:
professor
processor
professors
```

```
./agrep.sh incredible
FOUND:
incredible
```

```
./agrep.sh incredibel
Did you mean:
incredible
incredibly
```

4. Kleuren output

Doel: Kleuren van output commando ps

Hint: Met het commando shopt -q -s extglob schakel je herkenning van regex patronen aan, met shopt -q -u extglob schakel je dit terug uit.

Schrijf het script *kleurps.sh*. Dit script voert de volgende functies uit:

- Het leest met een for lus de output van ps -eo pid,args
- Het kleurt met behulp van parameter substitutie en regex de eerste kolom rood en de tweede kolom blauw.

5. cut regex

Schrijf het script *cutregex.sh*. Dit script voert dezelfde functies uit als cut.sh. Alleen gebruik je nu regex groups om de commandline parameters in te lezen

6. poorten regex

Schrijf het script *poortenregex.sh*. Dit script voert dezelfde functies uit als *poorten.sh*. Alleen gebruik je nu een regex group om de poort uit te lezen

7. nslookup regex

Schrijf het script *nslookupregex.sh*. Dit script voert dezelfde functies uit als *nslookup.sh*. Gebruik hier een regex group om enkel de naam uit te lezen uit de output van *nslookup*

8. regex configuratiebestand

Schrijf het script *servername.sh*.

- a) Het script zoekt in het configuratiebestand */etc/apache2/apache2.conf* naar de ingestelde parameter *ServerName*. bv *ServerName mijnlaptop*
- b) Wanneer de parameter is ingesteld, dan laat je deze staan, wanneer deze nog niet is ingesteld voeg je deze toe aan het configuratiebestand.