LESSON 5

report.awk

```
if ($2 >= 90)
  grade = "A"
else
  grade = "F"
```

AWK

if · for · next

Learn it all @MyWebUniversity



A FREE MYWEBUNIVERSITY COURSE

AWK SCRIPTING

LESSON 5

Control Flow

if, else, while, for, next and exit

if and else

grade.awk

```
{
    if ($2 >= 90)      grade = "A"
    else if ($2 >= 75) grade = "B"
    else if ($2 >= 60) grade = "C"
    else               grade = "F"
    print $1, grade, ($2 >= 60 ? "pass" : "fail")
}
```

- Inside an action, AWK has the same if and else as C. Put the condition in parentheses and …
- For short choices, the ternary operator is handy: condition, question mark, value if true, …

01

while and for

rowsum.awk

```
{
    sum = 0
    for (i = 1; i <= NF; i++)
        sum += $i
    print "line", NR, "sum =", sum
}

# 10 20 30  ->  line 1 sum = 60
```

- Most of the time AWK loops over lines for you, but sometimes you loop over fields. A for …
- while and do-while work too. Here we add up all the numbers on each line, however many …

02

next and exit

next_exit.awk

```
/^#/ { next }                    # skip comments
NR == 1 { next }                 # skip header
$1 == "STOP" { exit }           # stop reading
{ count++ }
END { print count, "data lines" }
```

- next stops processing the current line and jumps to the next one. Use it to skip comments …
- exit stops reading input completely, but the END block still runs. It is useful when you …

03

Multi-Line Programs

terminal

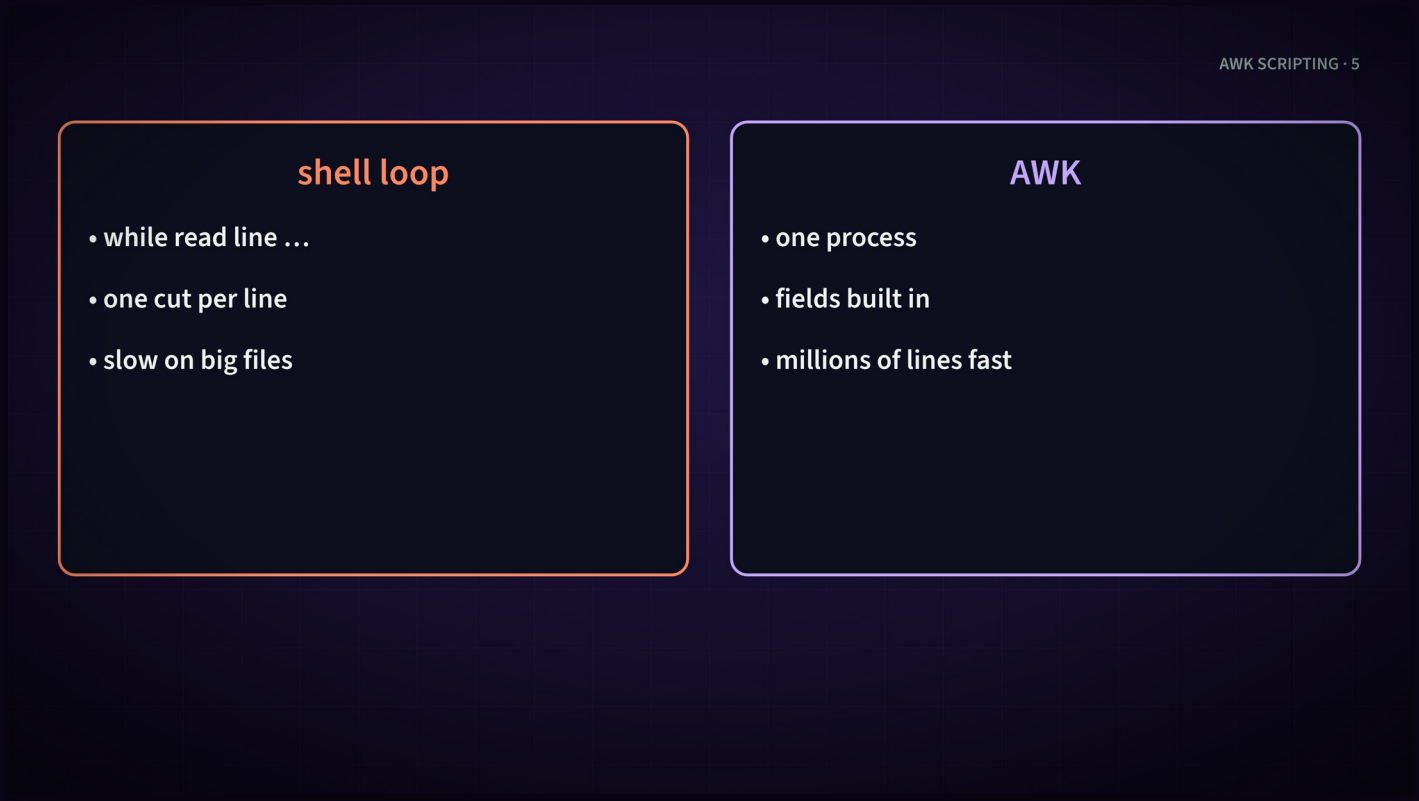
```
$ cat report.awk
# report.awk — totals per line
NR == 1 { next }
{
    total += $2
}
END { printf "Total: %d\n", total }

$ awk -f report.awk scores.txt
```

- One-liners grow. When they do, put the program in a file and run it with awk dash f. …
- Indent like any other language. A readable AWK file is easy to keep and share with your team.

04

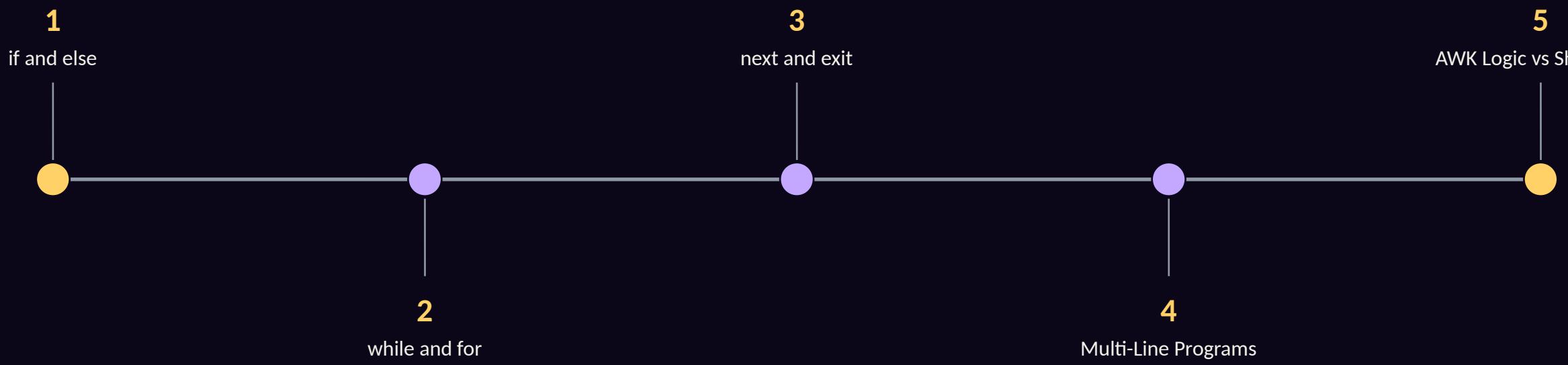
AWK Logic vs Shell Logic



- A shell while-read loop runs a separate program for every line and is slow. One AWK process …
- So when your shell script loops over lines and cuts columns, it is usually time to switch …

05

Lesson Map


1
if and else
2
while and for
3
next and exit
4
Multi-Line Programs
5
AWK Logic vs Shell Logic

Sources and Further Reading

grade.awk

```
{
    if ($2 >= 90)      grade = "A"
    else if ($2 >= 75) grade = "B"
    else if ($2 >= 60) grade = "C"
    else               grade = "F"
    print $1, grade, ($2 >= 60 ? "pass" : "fail")
}
```


next_exit.awk

```
/^#/ { next }                    # skip comments
NR == 1 { next }                 # skip header
$1 == "STOP" { exit }            # stop reading
{ count++ }
END { print count, "data lines" }
```

Thank you for watching

MyWebUniversity.com

Free technical education since 2000 — English and Dari

@MyWebUniversity · @MyWebUniversity-Dari