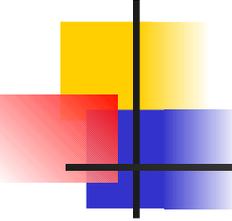


Week 10 – Lesson 2: **awk & print, printf**

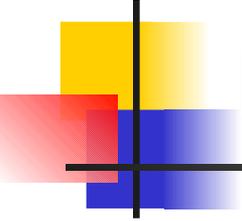


Chapter Objectives

In this chapter, you will:

Learn additional shell scripting tools including:

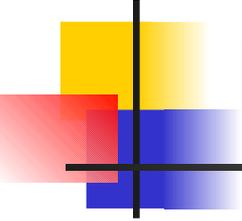
- More on awk Utility
 - Displaying Standard Output
 - **Print** command
 - **Printf** command
 - Generating Reports with **Headers** and **Footers**



Introduction to awk Utility

Definition:

- **awk** is a general purpose programming language that is designed for processing text-based data, either in files or used as filters in pipeline commands.
- *awk* has many useful applications:
 - Text manipulation
 - Report Generation from Databases
 - Floating point decimal calculations



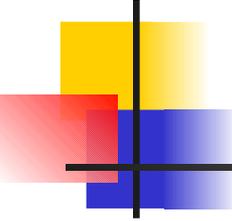
Introduction to awk Utility

SYNTAX:

```
awk [options] '/re/ {execution}' filename
```

Options:

```
-f scriptfilename (execute from script)  
-F" ; " (sets ; as default delimiter)
```

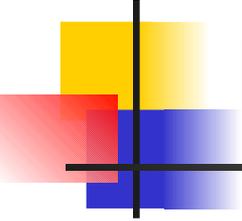


Introduction to awk Utility

Variables

- The following is a list of common variables used with awk:

\$0 Entire Record
\$n *Field number* “n” in Record (eg. \$1, \$2, \$3)
NF Number of fields in record
NR Record number of current record
FS Input Field Separator (default space / tab)
OFS Output Field Separator (default space)
RS Input Record Separator (default new line)
ORS Output Record Separator (default new line)
FILENAME Name of current input file



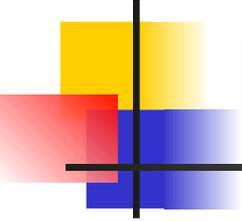
Introduction to awk Utility

Execution Commands

- Here are some common commands that can be used in the execution of awk (contained in braces { }):

print Can use variables like \$1,\$2, etc. When using those types of variables separate with a comma (no spaces). The comma represents the default output field separator. The variable for the default output separator is **OFS**

printf very similar to print but provides formatting options for the display of values (eg. # of decimal places) (refer to examples in Sample Script section of this week's resources...)



Introduction to awk Utility

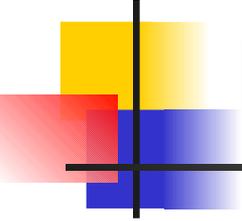
printf

- This command, by default, does not print the newline character.
- Escape sequences beginning with the backslash symbol can be used:

`\n` - new line

`\t` - tab

`\\` - the `\` symbol



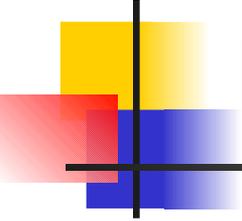
Introduction to awk Utility

printf

- This command can be used to place values of variables within the printf statement...
- The obvious example:

```
var="hello"
```

```
printf "${var}, how are you?\n\n"
```



Introduction to awk Utility

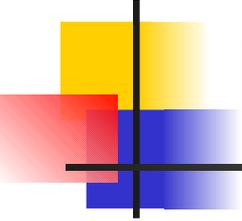
printf

- In the `awk` command, values can be inserted using format specifiers to explain their purpose:
 - `%s` - string
 - `%i` - integer number
 - `%f` - floating point decimal number

- Example:

```
var="hello"
```

```
printf "%s, how are you?\n\n", $var
```



Introduction to awk Utility

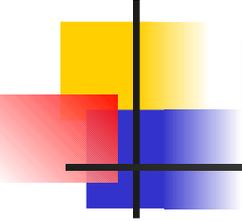
printf

- Example:

```
cat data.txt
```

```
Murray;23;professor
```

```
awk -F";" '{printf "Occupation:\t\nCategory:\t%i\nName:%s\n\n",\n$3,$2,$1}' data.txt
```



Introduction to awk Utility

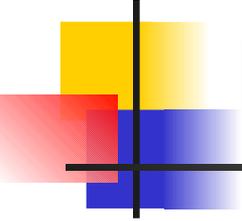
printf

- Instead of using `\t` (tabs) to separate output, the format specifiers can be used to align and space results.

`%20s` - 20 positions (right-aligned)

`%-10i` – 10 positions (left-aligned)

You will be seeing this in lab #9...



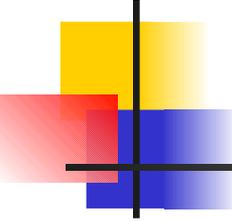
Introduction to awk Utility

printf

- The BEGIN and END statements in awk can be used to perform only one operation ONCE – at the beginning of the report, and at the ending of the report....

- Example:

```
awk 'BEGIN {printf "title\n\n"} $2 ~ /chevy {print}  
    END {printf "\nEnd of Report\n\n"}' input-file
```



Summary

There are a limited number of commands that the **awk** command uses, but they are very powerful.

The **print** and **printf** commands are used to display output. The **printf** command is used to display formatted output.

The **awk** command can use **BEGIN** to perform a single operation at the start of the report, and use **END** to perform a single operation at the conclusion of the report