AWK & SED Cheat Sheet
======================

AWK – Text Processing & Pattern Scanning

AWK extracts, filters, and transforms data from text files.

Key Concepts:

- Each row = a line
- Each column = a field ($1, $2, $3…)

AWK Variables:

- $0 : Entire line
- $1,$2,$3 : Fields
- NF : Number of fields
- NR : Line number
- FS : Input Field Separator
- OFS : Output Field Separator

Syntax:

```
awk 'pattern { action }' filename
```

Examples (students.txt):

```
John 85
Anita 90
Rahul 70
Megha 95
```

1. Print first column:

```
awk '{print $1}' students.txt
```

2. Print second column:

```
awk '{print $2}' students.txt
```

3. Print names scoring above 80:

```
awk '$2 > 80 {print $1}' students.txt
```

4. Print line number + line:

```
awk '{print NR, $0}' students.txt
```

5. Print lines matching condition:

```
awk '$1 == "Rahul"' students.txt
```

6. NF – Number of fields:

```
awk '{print $0, "=> Fields:", NF}'
```

7. NR – Line number:

```
awk '{print "Line:", NR, "-", $0}'
```

FS – Field Separator

----

data.txt:

```
Alice,24,Developer
Bob,30,Tester
```

Split by comma:

```
awk -F "," '{print $1, $3}' data.txt
```

Split passwd by colon:

```
awk -F ":" '{print $1, $6}' /etc/passwd
```

OFS – Output Field Separator

Default:

```
Alice 24
```

Set OFS to dash:

```
awk 'BEGIN{OFS="-"} {print $1, $2}' data.txt
```

Output: Alice-24

Set OFS to comma:

```
awk 'BEGIN{OFS=","} {print $1, $2}' data.txt
```

Output: Alice,24

SED – Stream Editor

==================

SED modifies files using search, replace, delete, insert, append.

Syntax:

```
sed 'command' filename
```

file.txt:

```
apple
```

```
banana
grape
banana
mango
```

1. Replace first occurrence:

```
sed 's/banana/orange/' file.txt
```

2. Replace all occurrences:

```
sed 's/banana/orange/g' file.txt
```

3. Delete line 3:

```
sed '3d' file.txt
```

4. Delete lines containing word:

```
sed '/banana/d' file.txt
```

5. Print only matching lines:

```
sed -n '/grape/p' file.txt
```

6. Insert line BEFORE pattern:

```
sed '/banana/i FRUIT LIST'
```

7. Append line AFTER pattern:

```
sed '/banana/a This is yellow fruit'
```

8. Replace only on line 2:

```
sed '2s/banana/orange/' file.txt
```

9. Delete range (2 to 4):

```
sed '2,4d' file.txt
```

10. In■place replace:

```
sed -i 's/banana/orange/g' file.txt
```