Week2 – Day3

----Array

Declaring arrays --

```
# Method 1: Using parentheses (Recommended)
fruits=("Apple" "Banana" "Cherry")

# Method 2: Assigning values individually
fruits[0]="Apple"
fruits[1]="Banana"
fruits[2]="Cherry"
```

Accessing elements --

```
echo "${fruits[0]}"   # Apple
echo "${fruits[1]}"   # Banana
echo "${fruits[2]}"   # Cherry
```

Printing entire Array --

```
echo "${fruits[*]}"
```

----Operators.sh

Operator	Description	Example
`+`	Addition	`echo $((5 + 3))` → `8`
`-`	Subtraction	`echo $((10 - 2))` → `8`
`*`	Multiplication	`echo $((4 * 3))` → `12`
`/`	Division	`echo $((8 / 2))` → `4`
`%`	Modulus (Remainder)	`echo $((10 % 3))` → `1`
`**`	Exponentiation (Bash 4+)	`echo $((2 ** 3))` → `8`

Operator	Description	Example
`-eq`	Equal to	`[ "$a" -eq "$b" ]`
`-ne`	Not equal to	`[ "$a" -ne "$b" ]`
`-gt`	Greater than	`[ "$a" -gt "$b" ]`
`-lt`	Less than	`[ "$a" -lt "$b" ]`
`-ge`	Greater than or equal to	`[ "$a" -ge "$b" ]`
`-le`	Less than or equal to	`[ "$a" -le "$b" ]`

Used for string comparisons.

Operator	Description	Example
==	Equal to	`[ "$str1" == "$str2" ]`
!=	Not equal to	`[ "$str1" != "$str2" ]`
-z	String is empty	`[ -z "$str" ]`
-n	String is not empty	`[ -n "$str" ]`

Used for combining conditions.

Operator	Description	Example
&&	And	`[ "$a" -gt 5 ] -a [ "$b" -lt 15 ]`
||	Or	`[ "$a" -gt 5 ] -o [ "$b" -lt 15 ]` `[[ "$a" -gt 5 \|\| "$b" -lt 15 ]]`
!	Not	! $withdrawal

Reading User Input (read command)

- `read -p "message" variable1 variable2` → Reads multiple inputs.
- `read -s` → Hides input (for sensitive data like passwords).

Case Statement (Menu Selection)

- Used for handling multiple conditions.
- Ideal for menu-based choices.

```
read -p "Enter selection [1-3]: " selection
case $selection in
   1) accounttype="checking"; echo "You selected Checking";;
   2) accounttype="saving"; echo "You selected Saving";;
   3) accounttype="current"; echo "You selected Current";;
   *) accounttype="random"; echo "Random selection";;
esac
```