Arrays:

In shell scripting, you can use arrays to store multiple values in a single variable. The syntax for declaring and using arrays can vary slightly depending on the shell you're using (e.g., Bash, Zsh). Here's how to do it in Bash, which is one of the most commonly used shells.

1. **Declaring an Array:**

   You can declare an array using parentheses ( ):

   ```
   # Declare an array
   my_array=(value1 value2 value3)
   ```

2. **Accessing Array Elements**

   You can access elements of the array using the index (starting from 0):

   ```
   # Accessing elements
   echo ${my_array[0]}  # Output: value1
   echo ${my_array[1]}  # Output: value2
   ```

3. **Adding Elements to an Array**

   You can add elements to an existing array:

   ```
   # Adding an element
   my_array+=("value4")  # Now my_array has 4
   elements
   ```

4. **Getting the Length of an Array**

   You can get the length (number of elements) of an array using the # operator:

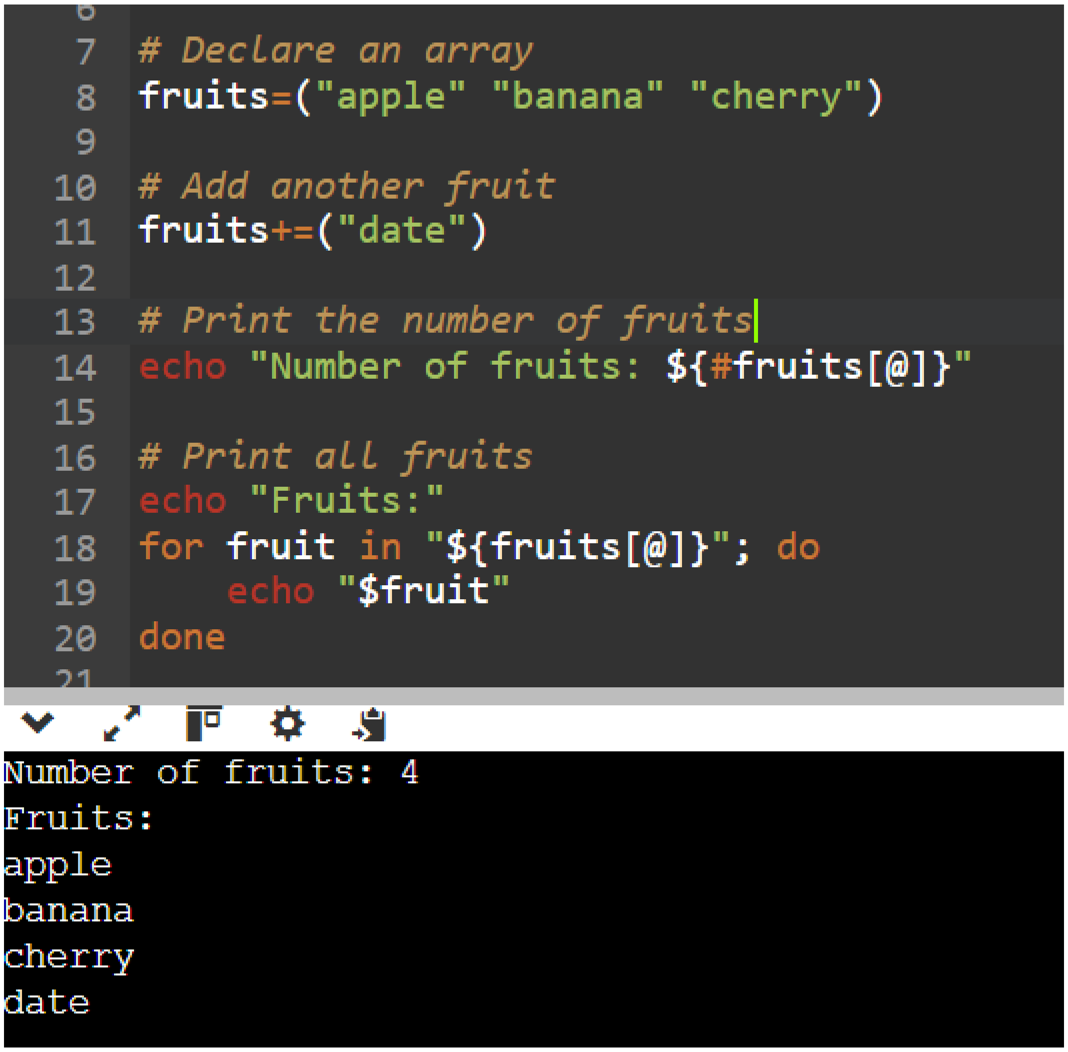
```
# Getting the length of the array
echo ${#my_array[@]}  # Output: 4
```

5. Looping Through an Array

You can loop through the elements of an array using a `for` loop:

```
# Looping through the array
for value in "${my_array[@]}"; do
        echo $value
Done
```

Example:



```bash
# Declare an array
fruits=("apple" "banana" "cherry")

# Add another fruit
fruits+=("date")

# Print the number of fruits
echo "Number of fruits: ${#fruits[@]}"

# Print all fruits
echo "Fruits:"
for fruit in "${fruits[@]}"; do
    echo "$fruit"
done
```

```
Number of fruits: 4
Fruits:
apple
banana
cherry
date
```