



BridgeLabz

Employability Delivered

Programming
Constructs -
Dictionaries

6. Dictionaries

Dictionary / associative arrays / hash map are very useful data structures

Dictionary are key value map similar to HashMap in Java or the one in python

Dictionary Example

```
#!/usr/local/bin/bash -x
#Note using Latest Bash Version 5.0

declare -A sounds
sounds[dog]="bark"
sounds[cow]="moo"
sounds[bird]="tweet"
sounds[wolf]="howl"

echo "Dog Sound " ${sounds[dog]} # Dog's sound
echo "All Animal Sound " ${sounds[@]} # All values
echo "Animal " ${!sounds[@]} # All keys
echo "Number of Animals " ${#sounds[@]} # Number of elements
unset sounds[dog] # Delete dog
```

```
+ declare -A sounds
+ sounds[dog]=bark
+ sounds[cow]=moo
+ sounds[bird]=tweet
+ sounds[wolf]=howl
+ echo 'Dog Sound ' bark
Dog Sound bark
+ echo 'All Animal Sound ' tweet bark moo howl
All Animal Sound tweet bark moo howl
+ echo 'Animal ' bird dog cow wolf
Animal bird dog cow wolf
+ echo 'Number of Animals ' 4
Number of Animals 4
+ unset 'sounds[dog]'
```



UC 9

Store the Day and
the Daily Wage
along with the Total
Wage

Storing Day & Daily Wage

```
#!/bin/bash -x

# CONSTANTS FOR THE PROGRAM
IS_PART_TIME=1;
IS_FULL_TIME=2;
MAX_HRS_IN_MONTH=4;
EMP_RATE_PER_HR=20;
NUM_WORKING_DAYS=20;

# VARIABLES
totalWorkHours=0;
totalWorkingDays=0;

function getWorkingHours() {
    case $1 in
        $IS_FULL_TIME)
            workHours=8
            ;;
        $IS_PART_TIME)
            workHours=4
            ;;
        *)
            workHours=0
            ;;
    esac
    echo $workHours
}

function calcDailyWage() {
    local workHrs=$1
    wage=$((workHrs*EMP_RATE_PER_HR))
    echo $wage
}

while [[ $totalWorkHours -lt $MAX_HRS_IN_MONTH &&
        $totalWorkingDays -lt $NUM_WORKING_DAYS ]]
do
    ((totalWorkingDays++))
    workHours=$((getWorkingHours $((RANDOM%3))) )
    totalWorkHours=$((totalWorkHours+workHours))
    empDailyWage["$totalWorkingDays"]=$((calcDailyWage $workHours ))
done

totalSalary=$((calcDailyWage $totalWorkHours ))
echo "Daily Wage " ${empDailyWage[@]}
echo "All Keys " ${!empDailyWage[@]}
```

```
+ IS_PART_TIME=1
+ IS_FULL_TIME=2
+ MAX_HRS_IN_MONTH=4
+ EMP_RATE_PER_HR=20
+ NUM_WORKING_DAYS=20
+ totalWorkHours=0
+ totalWorkingDays=0
+ [[ 0 -lt 4 ]]
+ [[ 0 -lt 20 ]]
+ (( totalWorkingDays++ ))
++ getWorkingHours 2
++ case $1 in
++ workHours=8
++ echo 8
+ workHours=8
+ totalWorkHours=8
++ calcDailyWage 8
++ local workHrs=8
++ wage=160
++ echo 160
+ empDailyWage["$totalWorkingDays"]=160
+ [[ 8 -lt 4 ]]
++ calcDailyWage 8
++ local workHrs=8
++ wage=160
++ echo 160
+ totalSalary=160
+ echo 'Daily Wage ' 160
Daily Wage 160
+ echo 'All Keys ' 1
All Keys 1
```



Dictionary Practice Problems

1. Write a program in the following steps
 - a. Roll a die and find the number between 1 to 6
 - b. Repeat the Die roll and find the result each time
 - c. Store the result in a dictionary
 - d. Repeat till any one of the number has reached 10 times
 - e. Find the number that reached maximum times and the one that was for minimum times
2. Write a Program to generate a birth month of 50 individuals between the year 92 & 93. Find all the individuals having birthdays in the same month. Store it to finally print.



BridgeLabz

Employability Delivered

Thank
You