

Associative array !

Πίνακας περιεχομένων

1. Associative array, dictionary or map	1
1.1. Exercise	1
1.2. create	5
1.3. for	5
1.4. create from array	5
1.5. test	6
1.6. indexing	6
1.7. unset key	6
1.8. unset all	6



NOTE

Assuming you're already logged in

1. Associative array, dictionary or map

1.1. Exercise

- vim testfile.sh
- copy-paste
- chmod +x ./testfile.sh
- ./testfile.sh

```
#!/bin/bash

bash --version # Must be at least version 4 to have associative arrays

declare -A array=(
  [1,0]="first name"
  [1,1]="second name"
  [2,0]="third name"
  [3,0]="foo bar"
  [test 1]="test bar"
)

var=$(cat <<EOF
declare -A array=(
```

```

[1,0]="first name"
[1,1]="second name"
[2,0]="third name"
[3,0]="foo bar"
[test 1]="test bar"
)
EOF
)

echo "$var"

echo ' count with ${#array[@]} result:'
echo ${#array[@]}

echo ""
echo ""
cat <<'EOF'
-----
    for with "${!array[@]}"
-----
for i in "${!array[@]"; do
    echo "$i => ${array[$i]}";
done

    "result: "
EOF

for i in "${!array[@]"; do
    echo "$i => ${array[$i]}";
done

echo ""
echo ""

cat <<'EOF'
-----
    for with ${!array[@]}
-----
for i in ${array[@]}; do
    echo "$i => ${array[$i]}";
done

    "result: "
EOF

for i in ${array[@]}; do
    echo "$i => ${array[$i]}";
done

echo ""

```

```

echo ""

cat <<'EOF'
-----
    for with "${!array[*]}"
-----
for i in "${!array[*]}"; do
    echo "$i => ${array[$i]}";
done

    "result: "
EOF

for i in "${!array[*]}"; do
    echo "$i => ${array[$i]}";
done

echo ""
echo ""

cat <<'EOF'
-----
    for with ${!array[*]}
-----

for i in ${!array[*]}; do
    echo "$i => ${array[$i]}";
done

    "result: "
EOF

for i in ${!array[*]}; do
    echo "$i => ${array[$i]}";
done

echo ""
echo ""

cat <<'EOF'
-----
    we can get the value of the '1,0' key as array
-----

    declare -a names=(${array[1,0]})
EOF

declare -a names=(${array[1,0]})

echo ""
echo ""
cat <<EOF

```

Variable names now contains array:

```
echo \ $names  
result: $names
```

```
echo \ ${names[1]}  
result: ${names[1]}
```

EOF

```
echo ' count with ${#names[@]} result:'  
echo ${#names[@]}
```

```
echo ""  
echo ""
```

```
cat <<'EOF'
```

```
-----  
Testing whether a value is missing from an associative array  
-----
```

```
if [ ${array[1,0]+_} ]; then  
    echo "Found";  
else  
    echo "Not found";  
fi
```

```
EOF  
if [ ${array[1,0]+_} ]; then  
    echo "Found";  
else  
    echo "Not found";  
fi
```

```
echo ""  
echo ""
```

```
echo "-----"  
echo "indexing"  
echo "-----"
```

```
echo 'KEYS=("${!array[@]}") # Make a normal array containing all the keys in the  
associative array'
```

```
KEYS=("${!array[@]}")  
echo ""
```

```
echo 'echo ${KEYS[0]} # Find a key via an index'  
echo ${KEYS[0]}  
echo ""
```

```
echo 'echo ${array[${KEYS[0]}]} # Find a value via an index'
```

```

echo ${array[${KEYS[0]}]}

echo ""
echo "-----"
echo "remove key"
echo "-----"
echo ""
echo 'unset array["test 1"]    # You must quote keys containing spaces when you unset
in an associative array'
unset array["test 1"]
echo ""
echo ' count with ${#array[@]} result:'
echo ${#array[@]}
echo ""

echo "-----"
echo ' remove all with "unset array" '
echo "-----"
unset array

echo ' count with ${#array[@]} result:'
echo ${#array[@]}

```

1.2. create

```

declare -A array=(
  [1,0]="first name"
  [1,1]="second name"
  [2,0]="third name"
  [3,0]="foo bar"
  [test 1]="test bar"
)

```

1.3. for

```

for i in "${!array[@]}"; do
  echo "$i => ${array[$i]}";
done

```

1.4. create from array

```

declare -a names=(${array[1,0]})

```

1.5. test

```
if [ ${array[1,0]+_} ]; then
    echo "Found";
else
    echo "Not found";
fi
```

1.6. indexing

```
KEYS=("${!array[@]}")
echo ${array[${KEYS[0]}]}
```

1.7. unset key

```
unset array["test 1"]
```

1.8. unset all

```
unset array
```