

Agile

Containers

DevOps
Automation

DevOps Linux

DevOps Services

Languages

Network

Scripting

Security

Solutions

Virtualization

Windows

Glib Examples

Go Examples

Javascript
Examples

D4D Examples

APIs

**Bash Associative
Array**

Bash Functions

Bash Regex

Bash *

Color Distance [↗](#)Shell Problems [↗](#)

Shell-Scripting

awk

packages.json [↗](#)

sed



Bash Associative Array Cheat Sheet

[Edit Cheat Sheet ↗](#)See also [Bash](#)

Indexed Arrays

Unlike hashes indexed arrays require no declaration

```
# Assigning a fixed list
arr=("string 1", "string 2", "string 3")

# Assigning with indizes, allows sparse lists
arr=[1]="string 1", [2]="string 2", [4]="string 4")

# Adding single elements by index
arr[4]="string 4"
```

Check below under "Hashes" for accessing the different properties of an array.

Hashes

Can be used since Bash v4

```
# Hashes need declaration!
declare -A arr

# Assigning values to associative arrays
arr[my key]="my value"
arr["my key"]="my value"
arr[$my_key]="my value"

# Fetching values
echo ${arr[my key]}
echo ${arr["my key"]}
echo ${arr[$my_key]}

# Accessing the array
${arr[@]}      # Returns all indizes and their items (doesn't work with associative arrays)
${arr[*]}      # Returns all items
${!arr[*]}     # Returns all indizes
${#arr[*]}     # Number elements
```

CHEAT SHEETS

Agile

Containers

DevOps
Automation

DevOps Linux

DevOps Services

Languages

Network

Scripting

Security

Solutions

Virtualization

Windows

Glib Examples

Go Examples

Javascript
Examples

DHD Examples

APIs

**Bash Associative
Array**

Bash Functions

Bash Regex

Bash *

Color Distance [🔗](#)Shell Problems [🔗](#)

Shell-Scripting

awk

packages.json [🔗](#)

sed

```
 ${#arr[$n]}      # Length of $nth item
```

```
 # Pushing to array  
 arr+=("new string value", "another new value")
```

Iterating Hashes

```
 # Print = lines...  
 for k in "${!array[@]}"  
 do  
     printf "%s\n" "$k=${array[$k]}"  
 done
```

[Comment on Disqus](#)