Case statement

- what is a case statement
- case statement usage

Case statement

Case statement is used to make complex conditionals with multiple choices. You can totally use nested **if** statements instead of **case** ones, but this will be less readable and harder to maintain.

Case will compare the variable against every pattern one by one and will stop once it finds a match.

Also note, that the **case** statement is used when you need to match a variable against several values or patterns, so if you need to do more complex comparisons – you'll have to use **if-else** statement with **elif** extensions.

```
case $VARIABLE in
   PATTERN_1)
      STATEMENT_1
      ;;
   PATTERN_2)
      STATEMENT_2
      ;;
...
   PATTERN_N)
      STATEMENT_N
      ;;
esac
```

Case statement example

```
bash-3.2$ cat oranges.sh
#!/bin/bash

echo "Do you like apples or oranges?: "
read fruit

case $fruit in
  apples)  echo "Well, they're tasty!" ;;
  oranges) echo "Wow, me too!" ;;
  *)       echo "I didn't understand :(" ;;
esac
bash-3.2$
```

Code example

```
bash-3.2$ ./oranges.sh
Do you like apples or oranges?:
apples
Well, they're tasty!
bash-3.2$ ./oranges.sh
Do you like apples or oranges?:
oranges
Wow, me too!
bash-3.2$ ./oranges.sh
Do you like apples or oranges?:
pizza
I didn't understand :(
bash-3.2$
```

Execution exapmple

Case statement example

```
bash-3.2$ cat example.sh
#!/bin/bash

echo "You've entered the number $1"

case $1 in
  *[0,2,4,6,8])
    echo "the number is even"
    ;;
  *[1,3,5,7,9])
    echo "the number is odd!"
    ;;
esac
bash-3.2$
```

Code example

```
bash-3.2$ ./example.sh 50
You've entered the number 50
the number is even
bash-3.2$ ./example.sh 821
You've entered the number 821
the number is odd!
bash-3.2$ ./example.sh -590
You've entered the number -590
the number is even
bash-3.2$
```

Execution exapmple

Case statement practical example

```
bash-3.2$ cat prompt.sh
#!/bin/bash

echo -n "Do you like the Bash course? [yes/no]:"
read answer
case $answer in

        [Yy] | [Yy][Ee][Ss] )
                echo "That's right!"
                ;;

        [Nn] | [Nn][Oo] )
                echo ":(";
                exit 1
                ;;
        *) echo "Enter Y[es] or N[o]"
            ;;
esac
```

Code example

```
bash-3.2$ ./prompt.sh
Do you like the Bash course? [yes/no]:No
:(
bash-3.2$ ./prompt.sh
Do you like the Bash course? [yes/no]:YES
That's right!
bash-3.2$ ./prompt.sh
Do you like the Bash course? [yes/no]:Hmm
Enter Y[es] or N[o]
bash-3.2$
```

Execution exapmple

Case statement advanced practical example

```
bash-3.2$ cat mysql_runner.sh
#!/bin/bash

# create directory for data if it doesn't exist
mkdir -p `pwd`/data

case $1 in
  run)
    docker run --name mysql-server -v `pwd`/data:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=my-secret-pw -d mysql:8.0.23
    ;;
  shell)
    docker exec -it mysql-server bash
    ;;
  logs)
    docker logs mysql-server
    ;;
  terminate)
    docker stop mysql-server && docker rm mysql-server
    ;;
  help)
    echo "Run a container with mysql. Commands are ( run | shell | logs | terminate )"
    ;;
  *)
    echo "Command unknown. Known commands are in ( run | shell | logs | terminate )"
    ;;
esac
```

Case statement advanced practical example

```
bash-3.2$ ./mysql_runner.sh help
Run a container with mysql. Commands are ( run | shell | logs | terminate )
bash-3.2$ ./mysql_runner.sh run
a4bda8f5a0759c69829d42cf69f0a1de21ef07a22433d25937e74852691e318e
bash-3.2$ ./mysql_runner.sh shell
root@a4bda8f5a075:/# mysql --help
mysql  Ver 8.0.23 for Linux on x86_64 (MySQL Community Server - GPL)
Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Usage: mysql [OPTIONS] [database]
  -?, --help          Display this help and exit.
  -I, --help          Synonym for -?
  --auto-rehash       Enable automatic rehashing. One doesn't need to use
                      'rehash' to get table and field completion, but startup
                      and reconnecting may take a longer time. Disable with
                      --disable-auto-rehash.
                      (Defaults to on; use --skip-auto-rehash to disable.)
  -A, --no-auto-rehash
                      No automatic rehashing. One has to use 'rehash' to get
                      table and field completion. This gives a quicker start of
                      mysql and disables rehashing on reconnect.
  --auto-vertical-output
```

Thanks for watching!