



# Bash Scripting Part 4

P. TenHoopen - WMLUG



# Command Line Parameters

Bash scripts can be given parameters when they are run to pass in options and values.

The script is called with the parameters on the command line. For example:

```
myscript -d -f value1
```



# Command Line Parameters

The parameters are read using the `getopts` command in a while loop, and are easily processed with a case statement.

You can specify that some options require a value by putting a colon after the option name in the `getopts` statement.

For example, “`ab:c`” would require option `b` to have a value.



# Parsing Options

```
# parse command line options

while getopts "df:hv" options; do
  case $options in
    d ) DEBUG=1;;
    f ) FILE=$OPTARG;;
    h ) fnUsage
        exit 0;;
    v ) VERBOSE=1;;
    \? ) fnUsage
        exit 1;;
    * ) fnUsage
        exit 1;;
  esac
done
```



# Functions

A function is a block of code that is given a name, can be called/used multiple times, and may or may not return an exit value/error code.

```
SayHello()  
{  
    echo "Hello"  
}
```



# Simple Help Function

```
# define function to show usage of script
```

```
fnUsage( )
```

```
{
```

```
echo ""
```

```
echo "getopts Example"
```

```
echo ""
```

```
echo "Options:"
```

```
echo "    -d      Turn debugging on"
```

```
echo "    -f      Specify file to work with"
```

```
echo "    -h      Show this message"
```

```
echo "    -v      Verbose"
```

```
}
```



# Putting it all together

Demo