KEYPHRENE'S BLOG

Introduction to the Unix Shell

Introduction to the Unix Shell

Shell Variables

```
a=1
# Display
echo $a

# global variable
export b='abcd'
echo $b

# get return command into variable with back-tics
c=`date +%Y-%m-%d`
echo $c
```

Shell Arithmetic

```
# bash
cpt=$[$cpt-1]
# ksh
cpt=$(expr $cpt + 1)
# ash
cpt=$(($cpt-2))

echo $cpt

expr 1 + 3
expr 2 - 1
expr 10 / 2
expr 20 % 3
expr 10 \* 3
```

Substring and Split

```
TEST=/etc/machin:/etc/truc:/etc/bidule
FIRST=${TEST%%:*}
LAST=${TEST##*:}
echo $FIRST
echo $LAST
echo ${TEST:1:3}
```

Shell functions

```
# Simple bash function
hello() {
   echo Hello!
}

# function return
hello() {
   local a=$1
   echo value:$a
}
a=`hello 1`
echo $a

# Simple ksh function
foo()
{
    echo "Argument 1: $1"
    echo "Argument 2: $2"
    echo "All agruments: $*"
}
# Function Call
foo arg1 arg2
```

Conditions

Control flow - If

```
# Number example
a=1
if [ $a -eq 1 ]; then
 echo "[ OK - 1 ]"
elif [ $a -eq 2 ]; then
 echo "[ OK - 2 ]"
```

```
else
 echo "[ KO ]"
fi

# String example
a='abcd'
if [ $a = 'abcd' ]; then
 echo "[ OK - 1 ]"
elif [ "$a" = '2' ]; then
 echo "[ OK - 2 ]"
else
 echo "[ KO ]"
fi
```

Control flow - Case

```
days=1
case `expr $days % 7` in
 0)
  echo 'Monday';;
 1)
  echo 'Tuesday';;
 *)
  echo 'All days';;
esac

# Another example
prog=freeplayer
case "$1" in
 start)
  # start
  ;;
 stop)
  # stop
  ;;
 *)
  echo $"usage: $prog"
  ;;
esac
```

Loop

Control flow - For

```
# Read word
for i in `cat file.txt`
do
    echo $i
done

# Loop (1 - 10)
for i in `seq 1 10`
do
 echo $i
done
```

Control flow - While

```
while command
do
   commands
done
```

Tests

```
#  Files :
if [ -f fichier ] # if file exists
if [ -d fichier ] # if directory exists


# String :
if [ -z "$s" ] # empty string
if [ -n "$s" ] # not empty string
if [ $var ]
if [ $word = "coucou" ]
if [ $var != "string" ]

# Number :
if [ $# -eq 3 ] # Equal. If number of parameters is equal to 3
if [ $1 -ne 1 ] # NotEqual. If first parameter is different of 1
if [ $chiffre -gt 4 ] # GreaterThan >
if [ $chiffre -ge 5 ] # GreaterorEqual >=
if [ $chiffre -lt 5 ] # LessThan <
if [ $chiffre -le 5 ] # LessorEqual <=
```

Shell Arguments

```
# ex: test.sh arg1 arg2
```

```
# Script filename
echo $0
# First argument
echo $1
# Second argument
echo $2
# number of arguments
echo $#
# All arguments
echo $@
# Process ID
echo $$
# Process ID the last command
echo $!
# Return code (exit status)
echo $?

# Example to test return code
cat test.txt
if [ $? -eq 0 ]; then
 echo "[ OK ]"
else
 echo "[ KO ]"
fi
```

Input output redirection

```
# Output Standard redirection
ls -l > file.log

# Append Output Standard redirection
ls -l >> file.log

# Output Standard and Error redirection
ls -l > file.log 2>&1

# Append Output Standard and Error redirection
ls -l >> file.log 2>>&1

# Output Standard redirection with no file
ls -l > /dev/null

# Output Standard and Error redirection with no file
ls -l > /dev/null 2>&1

# Output Standard and Error redirection in file.log
```

```
ls -l 2>&1 | tee -a file.log

# Create empty file
> test.txt
```

Include file script

```
# include file script env.sh
. env.sh

echo true
```

Debug

```
set -x
# or
sh -x test.sh
```