

## 8. Test

```
test() {  
    # ...  
    test "$@" -eq 0  
    # ...  
}
```

```
$ type [  
[ is a shell builtin  
$ which [  
/usr/bin/[  
$ ls -l /usr/bin/[  
lrwxrwxrwx 1 root root 4 Mar 27 2000 /usr/bin/[ -> test
```

☐ , '[' ☐ ☐☐☐ ls ☐ ☐ ☐☐☐ ☐☐☐ ☐☐☐☐☐ ☐☐☐ ☐☐☐ :

```
if [ $foo = "bar" ]
```

```

1  test$foo = "bar" ]
2  .
3  'SPACE'
4  ,
5  'SPACE'
6  :

```

```
if SPACE [ SPACE "$foo" SPACE = SPACE "bar" SPACE ]
```

□□ : □□ □□ □□ □□ "=="□ □□□□ , □□ □□□ □□□ □□□□ □□ "=="□ □□□□ □□□ □□ "eq"□ □□□ □□□ .

test 0000 00 00 00000 . 00 00 0000 "man test" 0000 . 000000 0 00 0000 0000 00 00000 .

```
test if while  do  done  break  continue . test  done
done  do  done  do  do  done  done  done  done  !
if...then...else...  do  done  done  :
```

```
if [ ... ] then
    # if-code
else
    # else-code
fi
```

```
fi
# do something
esac
.
if [ ... ]
then
# do something
fi
:
```

```
if [ ... ]; then
# do something
fi
```

```
elif
:
```

```
if [ something ]; then
echo "Something"
elif [ something_else ]; then
echo "Something else"
else
echo "None of the above"
fi
```

```
[something ]
echo "Something"
, [ something_else ]
.
[something_else ]
echo "Something else"
.
"None of the above"
```

```
X
(-1, 0, 1, hello, bye
).
(
- 1
Dave
):
```

```
$ X=5
$ export X
$ ./test.sh
... output of test.sh ...
$ X=hello
$ ./test.sh
... output of test.sh ...
$ X=test.sh
$ ./test.sh
... output of test.sh ...
```

```
$X
( : /etc/hosts)
```

```
#!/bin/sh
if [ "$X" -lt "0" ]
```

```
then
    echo "X is less than zero"
fi

if [ "$X" -gt "0" ]; then
    echo "X is more than zero"
fi

[ "$X" -le "0" ] && \
    echo "X is less than or equal to  zero"

[ "$X" -ge "0" ] && \
    echo "X is more than or equal to zero"

[ "$X" = "0" ] && \
    echo "X is the string or number \"0\""

[ "$X" = "hello" ] && \
    echo "X matches the string \"hello\""

[ "$X" != "hello" ] && \
    echo "X is not the string \"hello\""

[ -n "$X" ] && \
    echo "X is of nonzero length"

[ -f "$X" ] && \
    echo "X is the path of a real file" || \
    echo "No such file: $X"

[ -x "$X" ] && \
    echo "X is the path of an executable file"

[ "$X" -nt "/etc/passwd" ] && \
    echo "X is a file which is newer than /etc/passwd"
```

1. 在代码块中，使用 `if` 语句来检查变量 `is_valid` 是否为 `True`。如果为 `True`，则打印 "Valid"；否则，打印 "Invalid"。

```

is_valid = True

if is_valid:
    print("Valid")
else:
    print("Invalid")

```

□ □□□ □ □ □□ test□ □□ , □□□ □ □ □ □□ □□ □□ □□□ □□ □ □□□□ .

-a, -e( " " ), -S( ), -nt( ), -ot( ), -ef( ) -O( ) ( : Solaris, AIX, HPUX /bin/sh) Aaron .

```
if (a < b) {
    a = b;
} else {
    b = a;
}
```

```
#!/bin/sh
[ $X -ne 0 ] && echo "X isn't zero" || echo "X is zero"
```

```
[ $X -ne 0 ] && echo "X isn't zero" || echo "X is zero"
```

```
[ -f $X ] && echo "X is a file" || echo "X is not a file"
[ -n $X ] && echo "X is of non-zero length" || \
    echo "X is of zero length"
```

❶ ❷ ❸ ❹ [ ❺ ❻ (❼ Ⓜ Ⓝ )Ⓜ Ⓜ ⓂⓂⓂ , Ⓜ ⓂⓂ testⓂ ⓂⓂⓂⓂ . ⓂⓂⓂ  
 Ⓜ ⓂⓂⓂ ⓂⓂⓂⓂ ⓂⓂⓂⓂ Ⓜ ⓂⓂⓂ ⓂⓂⓂ Ⓜ Ⓜ ⓂⓂⓂⓂ ⓂⓂⓂⓂ . if...then...else... ⓂⓂⓂ ⓂⓂ  
 Ⓜ ⓂⓂⓂⓂ ⓂⓂⓂⓂ . ⓂⓂⓂ ⓂⓂⓂ ⓂⓂⓂⓂ ⓂⓂⓂⓂⓂ ⓂⓂⓂ Ⓜ ⓂⓂⓂ ⓂⓂ ⓂⓂⓂⓂ [...]

XⓂ ⓂⓂⓂ ⓂⓂ ⓂⓂⓂ ⓂⓂⓂⓂ ⓂⓂ Ⓜ Ⓜ ⓂⓂⓂⓂ ⓂⓂⓂⓂ ⓂⓂⓂⓂⓂ :

```
test.sh: [: integer expression expected before -lt
test.sh: [: integer expression expected before -gt
test.sh: [: integer expression expected before -le
test.sh: [: integer expression expected before -ge
```

ⓂⓂ -lt, -gt, -le Ⓜ -ge ⓂⓂⓂ ⓂⓂ ⓂⓂⓂⓂ ⓂⓂⓂⓂ ⓂⓂⓂⓂⓂⓂ ⓂⓂⓂⓂ ⓂⓂ ⓂⓂⓂⓂⓂ . !=Ⓜ ⓂⓂ  
 ⓂⓂⓂ ⓂⓂⓂ "5"Ⓜ ⓂⓂⓂⓂ ⓂⓂⓂⓂⓂⓂ "Hello"Ⓜ ⓂⓂⓂ ⓂⓂⓂⓂ ⓂⓂⓂⓂ ⓂⓂⓂⓂ ⓂⓂⓂⓂ ⓂⓂ  
 ⓂⓂⓂ ⓂⓂⓂ ⓂⓂⓂⓂⓂ . Ⓜ ⓂⓂⓂⓂⓂⓂ Ⓜ ⓂⓂⓂⓂⓂ ⓂⓂⓂⓂⓂ ⓂⓂⓂ ⓂⓂⓂⓂⓂⓂ ⓂⓂ ⓂⓂⓂⓂ ⓂⓂ  
 ⓂⓂⓂ ⓂⓂⓂ ⓂⓂⓂⓂⓂ ⓂⓂⓂ :

```
echo -en "Please guess the magic number: "
read X
echo $X | grep "[^0-9]" > /dev/null 2>&1
if [ "$?" -eq "0" ]; then
    # If the grep found something other than 0-9
    # then it's not an integer.
    echo "Sorry, wanted a number"
else
    # The grep found only 0-9, so it's an integer.
    # We can safely do a test on it.
    if [ "$X" = "7" ]; then
        echo "You entered the magic number!"
    fi
fi
```

ⓂⓂⓂ ⓂⓂ ⓂⓂⓂⓂⓂⓂ Ⓜ ⓂⓂ ⓂⓂ ⓂⓂⓂⓂ ⓂⓂⓂⓂ ⓂⓂⓂⓂ ⓂⓂⓂ Ⓜ ⓂⓂⓂⓂ . ⓂⓂⓂ ⓂⓂⓂⓂ 'ⓂⓂ  
 - 2Ⓜ '(10Ⓜ )ⓂⓂ ⓂⓂⓂⓂⓂ , grepⓂ ⓂⓂⓂⓂ ⓂⓂⓂⓂ ⓂⓂⓂⓂ (0~9)Ⓜ ⓂⓂ ⓂⓂⓂⓂ ⓂⓂⓂⓂ  
 ⓂⓂ ⓂⓂⓂⓂ grep [^0~9]Ⓜ ⓂⓂ (^)Ⓜ ⓂⓂⓂⓂⓂⓂ ⓂⓂⓂⓂ ⓂⓂ ⓂⓂⓂⓂⓂⓂ . ⓂⓂⓂ Ⓜ ⓂⓂⓂⓂ ⓂⓂ (

```

[ ] [ ] while [ ] [ ] test [ ] [ ] [ ] [ ] :

```

```
#!/bin/sh
X=0
while [ -n "$X" ]
do
    echo "Enter some text (RETURN to quit)"
    read X
    echo "You said: $X"
done
```

```

$ return $(cat /dev/urandom | tr -dc 'a-z0-9' | fold -n 10 | xargs | shuf | sed 's/ /_/' | tr -d '\n')
$ echo $(cat /dev/urandom | tr -dc 'a-z0-9' | fold -n 10 | xargs | shuf | sed 's/ /_/' | tr -d '\n')
Justin_Heath_0000000000
$ echo $(cat /dev/urandom | tr -dc 'a-z0-9' | fold -n 10 | xargs | shuf | sed 's/ /_/' | tr -d '\n')
Justin_Heath_0000000000
$ echo $(cat /dev/urandom | tr -dc 'a-z0-9' | fold -n 10 | xargs | shuf | sed 's/ /_/' | tr -d '\n')
Justin_Heath_0000000000
$ echo $(cat /dev/urandom | tr -dc 'a-z0-9' | fold -n 10 | xargs | shuf | sed 's/ /_/' | tr -d '\n')
Justin_Heath_0000000000

```

```
$ ./test2.sh
Enter some text (RETURN to quit)
fred
You said: fred
Enter some text (RETURN to quit)
wilma
You said: wilma
Enter some text (RETURN to quit)
```

\$

```
#!/bin/sh

X=0

while [ -n "$X" ]
do
    echo "Enter some text (RETURN to quit)"
    read X
    if [ -n "$X" ]; then
        echo "You said: $X"
    fi
done
```

