순서

- Bash Shell Programming
  - Quoting 보충
  - Looping
  - Per-Case Execution
  - Array
  - Example
  - Miscellaneous

- Q&A
## Examples of Quoting Rules

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$person</td>
<td></td>
</tr>
<tr>
<td>&quot;$person&quot; [or] &quot;$person&quot; [or] &quot;$person &quot;</td>
<td></td>
</tr>
<tr>
<td>$person</td>
<td></td>
</tr>
<tr>
<td>'$person'</td>
<td></td>
</tr>
<tr>
<td>&quot;'$person'&quot; [or] &quot;$person&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;&quot;$person&quot; [or] &quot;$person&quot; [or] &quot;$person&quot;&quot;</td>
<td></td>
</tr>
<tr>
<td>''&quot;$person'</td>
<td></td>
</tr>
<tr>
<td>&quot;&quot;$person&quot;</td>
<td></td>
</tr>
</tbody>
</table>
Looping

```bash
for arg in "$@"; do  # correct
    echo $arg
done
for arg in "$*"; do  # wrong
    echo $arg
done
```

**Result**

```
martini:~$ loop.sh 1 "2 2" 3
1
2 2
3
1 2 2 3
```
Looping (계속)

```
martini:~$ ls
backup.C backup.c backup1.C backup1.c
```

```
martini:~$ more for.sh
for cfilename in *$1*?.[cC]: do
  echo $cfilename
done
```

**Results**

```
martini:~$ for.sh a
backup.C
backup.c
```

```
martini:~$ for.sh c
backup.C
backup.c
```
Looping (계속)

```bash
i=0
while [ $i -lt 3 ]; do
  echo $i
  i=`expr $i + 1`
done
```

Result

```
martini:~$ while.sh
0
1
2
```
Per-Case Execution

```bash
Per-Case Execution

```

```bash
echo Enter your command \(\text{(who, list, or quit)}\)
while read command; do
  case $command in
    who) who; echo Done with running who;;
    list) ls; echo Done with running ls;;
    quit) break;;
    *) echo Enter who, list, or quit;;
  esac
done

```

```bash
Result

```

```bash
martini:~$ case.sh
Enter your command (who, list, or quit)
who
net001 pts/2 Mar 31 13:26 ... (omitted)
Done with running who
list
... Done with running ls
date
Enter who, list, or quit
quit
martini:~$

```

```bash
Result

```
Array (\textit{bash} Only)

- Variable Containing Multiple Values
  - No Maximum Limit to the Size
  - No Requirements That Member Variables Be Indexed or Assigned Contiguously
  - Zero-Based
Array (계속)

```bash
a1=(one two three)
echo ${a1[*]}
echo a1[*]
echo ${a1[@]}
echo a1[@]
echo ${a1[0]}
echo a1[0]
echo ${a1[3]}
unset a1[3]
unset a1[*]
echo ${a1[0]}
```

Result

```
one two three
a1[*]
one
two
three
a1[@]
one
a1[0]
```

```bash
# unset
i=0
echo $i
unset i
echo $i
--------
0
```
i=0
status=0
if [ $# -lt 1 ]; then
    echo Usage: $0 [-b base1 base2...] [-c expo1 expo2...]
    exit 1
fi
for arg in "@"; do
    if [ -n "echo $arg | grep -i" ]; then
        if [ $arg = -b ]; then
            status=1
        elif [ $arg = -c ]; then
            status=2
        fi
    else
        case $status in
        1 ) base[$i]=$arg; i=`expr $i + 1`;;
        2 ) exponent[$i]=$arg; i=`expr $i + 1`;;
        * ) echo Usage: $0 [-b base1 base2...] [-c expo1 expo2...]; exit 1;;
        esac
    fi
done
# to be continued on the next slide
Bash Script Example (계속)

```bash
if ! [ -d OUTPUT ]; then
    mkdir OUTPUT
    if [ $? -gt 0 ]; then  # if the exit status of the previous command indicates a failure
        echo Remove/Rename the OUTPUT File
        exit 1
    fi
fi
for i in ${base[@]}; do
    for j in ${exponent[@]}; do
        if ! [ -e ./OUTPUT/output.$i.$j ]; then
            ./pow $i $j >> ./OUTPUT/output.$i.$j
        fi
    done
done
```

**Execution**

```
martini:~$powsim.sh -b 3 4 5 -c 12 14 16
```
Miscellaneous

- Stream Editor: `sed`
  - Global Search and Replace
- Character Translator: `tr`
- Regular Expressions
  - for Editors (Including `sed`) and `grep`
- Option Interpretation: `getopts`
- Functions
- Signals
- ...