

Bash Lecture

Astronomy Laboratory at UniTS

Giuliano Taffoni – Giuseppe Murante

BASH

- Bash is a Unix shell and command language
 - Execute commands
 - “scripting” language

```
#!/bin/bash
```

```
echo Hello World
```

```
#!/bin/bash
```

```
ls
```

Variables

Are Untyped!!!!

```
a=375
```

```
hello=$a
```

```
hello="A B C D"
```

```
a=`ls -l`
```

```
arch=$(uname -m)
```

Variables

- `a=2334`
- `Array=(2 4 5 6 "otto")`
- `let "a = $a+ 1"`
- `a=$((a+1))`
- *Local variables*
- *Environmental variables*
- *Positional parameters \$0, \$1, \$*, @\$*
- *unset → free!*

Conditions

```
#!/bin/bash
```

```
if [ "foo" = "foo" ]; then
```

```
    echo expression evaluated as true
```

```
else
```

```
    echo expression evaluated as false
```

```
fi
```

loops

```
#!/bin/bash
for i in $( ls ); do
    echo item: $i
done
```

```
#!/bin/bash
for i in `seq 1 10`; do
    echo $i
done
```

```
#!/bin/bash
COUNTER=0
while [ $COUNTER -lt 10 ]; do
    echo The counter is $COUNTER
    let COUNTER=COUNTER+1
done
```

```
#!/bin/bash
for (( i = 0 ; i <= 20 ; i += 2 )) ;
do
    echo "i ha valore $i"
done
```

```
#!/bin/bash
COUNTER=20
until [ $COUNTER -lt 10 ]; do
    echo COUNTER $COUNTER
    let COUNTER-=1
done
```

Conditional statements

```
if [ -z "$a" ]
then
    echo "a not set"
    exit 1
fi

while [ $a -le "$LIMIT" ]
do
    a=$((a+1))

    if [ "$a" -eq 3 ] || [ "$a" -eq 11 ] # Excludes 3 and 11.
    then
        continue # Skip rest of this particular loop iteration.
    fi

    echo -n "$a " # This will not execute for 3 and 11.
done
```

Conditional Statements: files and dir

```
If [ -e $file_name]; then  
    echo exists  
fi
```

```
If [ -d $file_name]; then  
    echo is a dir  
fi
```

```
If [ -f $file_name]; then  
    echo exists and is a file  
fi
```

```
If [ !-d $file_name]; then  
    echo is not a dir  
fi
```


Functions

```
#!/bin/bash
```

```
JUST_A_SECOND=1
```

```
fun ()  
{ # A somewhat more complex function.  
  local i=0  
  REPEATS=30  
  sleep $JUST_A_SECOND    # Hey, wait a second!  
  while [ $i -lt $REPEATS ]  
  do  
    echo "<-----FUN----->"  
    let "i+=1"  
  done  
}
```

```
fun  
exit $?
```

Complex functions

```
#!/bin/bash
```

```
JUST_A_SECOND=1
```

```
fun ()  
{ # A somewhat more complex function.  
  local i=0  
  REPEATS=30  
  sleep $1      # Hey, wait a second!  
  while [ $i -lt $REPEATS ]  
  do  
    echo "<-----FUN----->"  
    let "i+=1"  
  done  
}
```

```
fun $JUST_A_SECOND  
exit $?
```

Read files

```
read -r riga < $file  
echo $riga
```

```
#!/bin/bash  
while IFS='' read -r line || [[ -n "$line" ]];  
do  
    echo "Text read from file: $line"  
done < "$1"
```

```
for word in $(cat $file); do  
    echo $line  
done
```