C2110 *UNIX and programming*

Lesson 6 / Module 2

PS / 2020 Distance form of teaching: Rev3

Petr Kulhanek

kulhanek@chemi.muni.cz

National Center for Biomolecular Research, Faculty of Science
Masaryk University, Kamenice 5, CZ-62500 Brno

Input/Output

Output: Command echo

Command **echo** used for unformatted output to standard output stream.

Syntax:

```
echo [options] [string2] [string2] ...
```

The command prints the strings in the specified order, separated by a space.

Useful choices:

-n does not line out the output

Examples:

```
$ echo "Horse"        "jumps" "high."
Horse jumps high.
$ echo "Horse    jumps    high."
Horse    jumps    high.
$ echo -n "Enter color: "
$ A=5
$ echo "Value of variable A is $A."
Value of variable A is 5.
```

Exercise I

1. In interactive mode, try the examples from the previous page.

Input: Command read

Command **read** is used to **read text** from the standard input (**i.e. for interactive loading of input**) and its storage in variables. The command always reads the whole line, the first word is stored in the first variable, ..., the rest of the line is stored in the last variable.

Syntax:

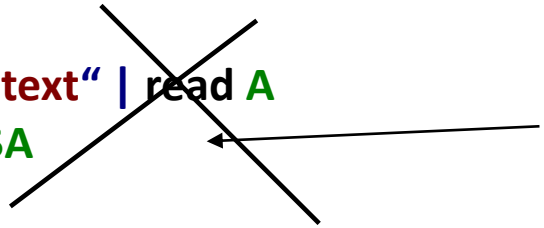
```
read A     # the whole line is stored in variable A
read A B   # the first word is stored in variable A
           # the rest of the line in variable B
```

Example:

```
echo -n "Enter value:"
read A
echo "Entered value is: $A"
```

Attention: do not use the command read in combination with pipes

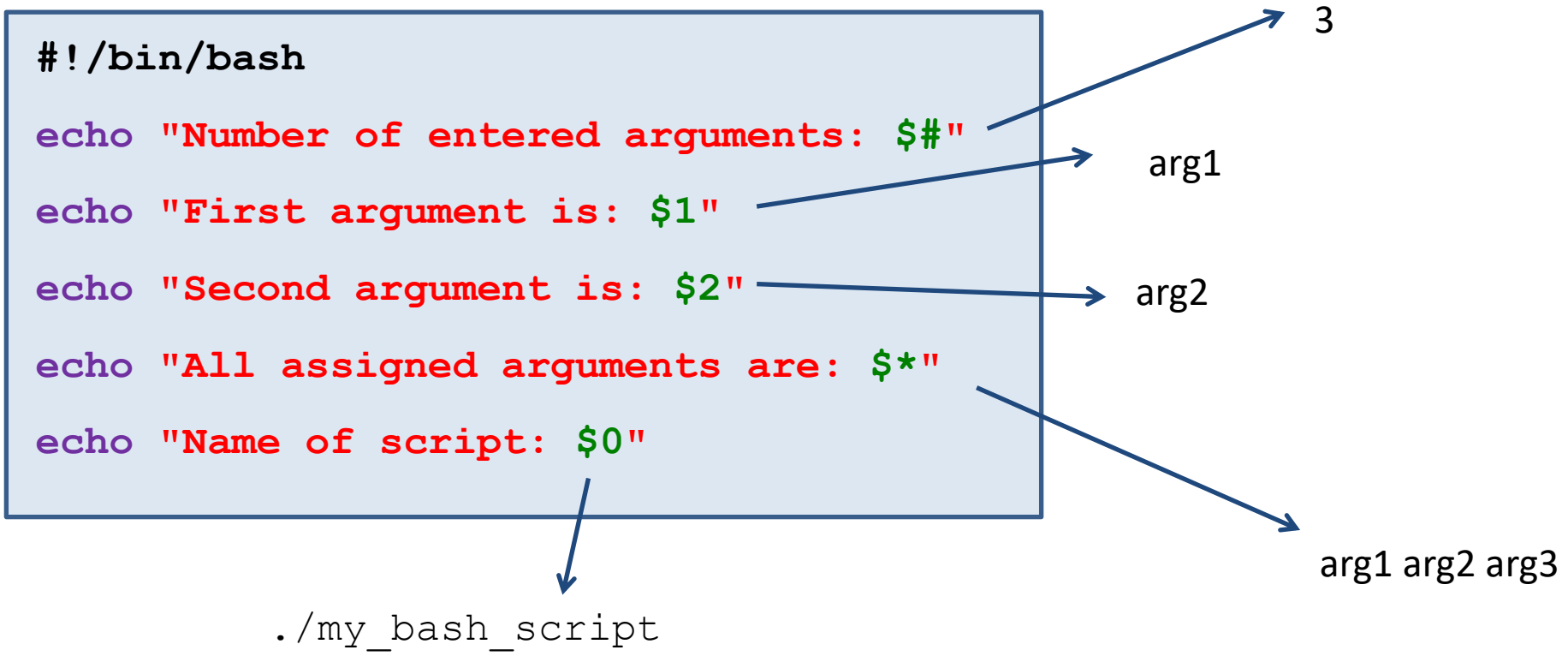
```
echo "text" | read A
echo $A
```



A won't contain the value "text". See the validity of variables in child processes.

Input: Script Arguments

```
$ bash my_bash_script arg1 arg2 arg3
$./my_bash_script arg1 arg2 arg3
```



The use and meaning of arguments is determined by the author of the script.

Script Arguments - Variables

Script arguments (special variable names):

#	number of arguments with which the script was run
0	name of the running script
1 … 9	values of arguments 1 to 9 with which the script was run
*****	all arguments with which the script was run

Advanced work with arguments:

"$@"	all arguments with which the script was run enclosed in quotation marks (handles input where arguments contain spaces) Different behavior towards **$*** or **"$*"** !

If we need to pass more than nine arguments, we must use the command **shift**. The command removes the first argument from the argument list.

```
for ((I=1; I<=$#; I++)); do
      echo $1
      shift
done
```

Sequentially lists entered script arguments.

Exercise II

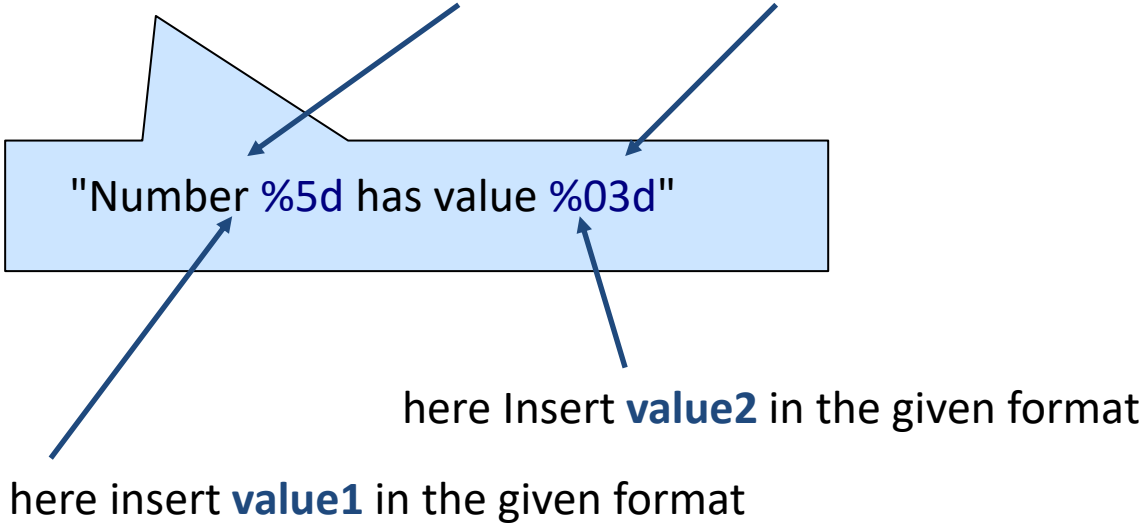
1. Write a script that asks the user for his favorite color and then writes it into the terminal.
2. Write a script that prints the number of entered arguments and the value of the first argument.

Output: command printf

Command **printf** used to print formatted texts and numbers.

Syntax:

```
printf [format] [value1] [value2] ...
```

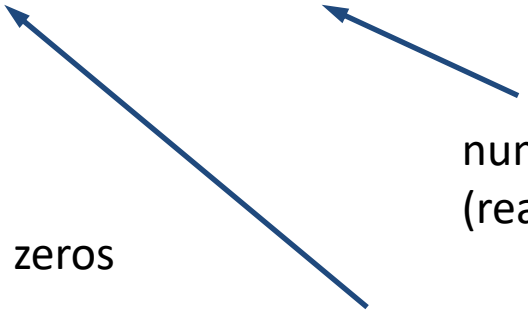


Further information: man bash, man printf

Command printf, Format

[] - optional part

%[flag][length][.accuracy]type



number of decimal places (real numbers)

field for total length

Flags:

-	align left
0	fill the blank space with zeros
+	always indicate the sign

Type:

d	integer
s	string (text)
f	real number

Special characters:

\n	end of line
\r	return to beginning of line
%%	% sign

More information: man bash, man printf

Command printf, Examples

```
$ I=10
$ B=12.345

$ printf "Value of variable I is %d\n" $I
Value of variable I is 10

$ printf "Entered number B is %10.4f\n" $B
Entered number B is    12.3450

$ printf "Entered number B is %010.4f\n" $B
Entered number B is 00012.3450

$ printf "Entered number B is %+010.4f\n" $B
Entered number B is +0012.3450

$ printf "Number I is %-5d and number B is %.1f\n" $I $B
The number I is 10    and the number B is 12.3
```

Exercise III

1. Write a script that asks the user for his favorite color and then writes it into the terminal. Print the query so that the color you enter is on the same line as the query.
2. Practice the command printf by executing the commands listed in the examples.
3. Write a script that prints the first script argument entered in %4d format.
4. Write a script that reads a number from standard input and prints it as follows: a sign will be given, five places will be used for the output, empty spaces will be filled with zeros:

   Entered number is: +0003
5. What happens if you submit the number 123456 to the script from Task 4?
6. Write a script to which takes two numbers as arguments. The script prints these numbers and then prints their sum.
7. Write a script which in turns asks the user for two numbers. Then it lists their product.