

COMMAND	BASH	/MINISHELL
BUILTS SIMPLE		
<code>echo</code>	/n	/n
<code>echo n</code>	return prompt	return prompt
<code>echo -nnnn</code>	return prompt	return prompt
<code>echo This is a test</code>	This is a test\n	This is a test\n
<code>echo -n This is a test</code>	This is a testarepsa@...	This is a test[MINISHELL]:...
<code>echo -nnnn This is a test</code>	This is a testarepsa@...	This is a test[MINISHELL]:...
<code>echo This is a test -nnan</code>	This is a test -nnnn\n	This is a test -nnnn\n
<code>echo \$?</code>		
<code>env</code>	prints environment variables	prints environment variables
<code>env PATH=</code>	env: 'PATH': No such file or directory	env: 'PATH': No such file or directory
<code>cd /</code>	arepsa@c1r15e9~\$	[MINISHELL]~\$
<code>cd .</code>	stay in the same directory	stay in the same directory
<code>cd ..</code>	\nCommand 'cd': not found, did you mean:...	command not found
<code>cd .ls</code>	bash: cd: too many arguments	minishell: cd: too many arguments
<code>cd ./././.</code>	arepsa@c1r15e9/\$	[MINISHELL]/\$
<code>cd /usr/bin</code>	arepsa@c1r15e9:/usr/bin\$	[MINISHELL]/usr/bin\$
<code>cd nonexistent</code>	minishell: cd: nonexistent: No such file or directory	minishell: cd: nonexistent: No such file or directory
<code>unset</code>	return prompt	return prompt
<code>unset nonexistentvariable</code>	return prompt	return prompt
<code>unset HOME</code>	deletes HOME, returns prompt	deletes HOME, returns prompt
<code>cd (after unsetting HOME)</code>	bash: cd: HOME not set	minishell: cd: HOME not set
<code>unset USER</code>	deletes USER from env	deletes USER from env
<code>export USER=bubu (do after unset)</code>	adds to env new user bubu	adds to env new user bubu
<code>export =</code>	bash: export: '=': not a valid identifier	minishell: export: '=': not a valid identifier
<code>export ABC=\$USER</code>	changes ABC=username	changes ABC=username
<code>export novalue</code>	returns prompt, keeps "novalue" in export print, but doesn't add to env	returns prompt, keeps "novalue" in export print, but doesn't add to env
<code>export=zzzzz</code>	returns promt, doesn't add any variables	returns promt, doesn't add any variables
<code>export _</code>	doesn't export _ (special case)	doesn't export _ (special case)
<code>fdsdfhb=</code>	returns prompt	returns prompt
<code>unset blabla OLDPWD</code>	deletes OLDPWD, returns prompt	deletes OLDPWD, returns prompt
<code>unset ALL VARIABLES</code>	unsets all, returns prompt	unsets all, returns prompt
<code>env (perform after unsetting all variables)</code>	bash: env: No such file or directory	returns prompt (and does enter the env builtin...)
<code>pwd</code>	prints home, returns prompt	prints home, returns prompt
<code>pwd args</code>	prints home, returns prompt	prints home, returns prompt
<code>export</code>	prints ordered env variables with "declare -x" before	prints ordered env variables with "declare -x" before
<code>env</code>	prints env variables	prints env variables
<code>exit</code>	exits with exit code 0	exits with exit code 0
<code>exit 300</code>	exits with exit code 44	exits with exit code 44
<code>exit -300</code>	exits with exit code 212	exits with exit code 212
<code>exit +300</code>	exits with exit code 44	exits with exit code 44
<code>exit 2 3</code>	bash: exit: too many arguments (and does not exit)	bash: exit: too many arguments (and does not exit)
<code>exit abc</code>	bash: exit: abc: numeric argument required + exits program	minishell: exit: abc: numeric argument required + exits program
<code>exit 9999999999999999999999999999999</code>	bash: exit: 9999999999999999999999999999999: numeric argument required	bash: exit: 9999999999999999999999999999999: numeric argument required
<code>unset -n HOME</code>	does nothing	unsets HOME
QUOTES		
<code>echo "Hello World"</code>	Hello World	Hello World
<code>echo 'Hello World'</code>	'Hello World'	'Hello World'
<code>"echo Hello World"</code>	echo Hello World: command not found	echo Hello World: command not found
<code>echo Hello World</code>	echo Hello World: command not found	echo Hello World: command not found
<code>echo "</code>	unclosed quotes	mensagem de erro de: unclosed quotes
<code>`bin/lis`</code>	execute ls	execute ls
<code>_minishell</code>	execute minishell	execute minishell
<code>"ls"</code>	execute ls	execute ls
<code>"ls"</code>	ls': command not found	ls': command not found
<code>"ls -l"</code>	execute ls with details	execute ls with details
<code>"c"ho' Hello</code>	Hello	Hello
<code>`pw"d</code>	prints home directory	prints home directory
<code>export USER=bubu LAST="quote'</code>	updates user and creates LAST="quote	updates user and creates LAST="quote
<code>echo '\$HOME e \$CASA e \$USER</code>	\$HOME e \$CASA e \$USER	(\$HOME e \$CASA e \$USER
<code>echo '\$HOME e \$CASA e \$USER'</code>	(expand)\$HOME e e (expand)USER	(expand)(HOME e e (expand)USER
<code>echo '\$HOME e \$CASA e \$USER'</code>	(expand)\$HOME e \$CASA e (expand)USER	(expand)(HOME e \$CASA e (expand)USER
<code>echo `cama`</code>	unclosed quotes	mensagem de erro de: unclosed quotes
<code>echo "\$HOME echo \$USER</code>	unclosed quotes	mensagem de erro de: unclosed quotes
<code>'ls' wc`</code>	ls ' wc: command not found	ls ' wc: command not found
<code>'ls' wc`</code>	unclosed quotes	mensagem de erro de: unclosed quotes
<code>ls grep "md" wc -l</code>	unclosed quotes	mensagem de erro de: unclosed quotes
<code>ls grep "md" wc -l</code>	0	0
<code>ls grep "md" wc -l</code>	1	1
<code>echo "md" wc</code>	1 1 3	1 1 3
<code>echo '\$HOME echo \$USER'</code>	(expand)\$HOME echo '(expand)USER	(expand)(HOME echo '(expand)USER
<code>echo "carro" echo `cama`</code>	carro echo cama	carro echo cama
<code>echo 'fred' echo `cama`</code>	fred echo `cama	'nao imprime nada'
<code>echo `cat lol.c cat > lol.c`</code>	cat lol.c cat > lol.c	'nao imprime nada'
<code>echo \$USER</code>	prints \$USER	prints \$USER
<code>export WWW="-aaa "</code>	saves variable WWW="-aaa "	saves variable WWW="-aaa "
<code>export ZZ='Y-aaa'bb</code>	exports variable ZZY="aaabb"	exports variable ZZY="aaabb"
<code>echo Hello</code>	echo Hello: command not found	echo Hello: command not found
<code>unset MAIL</code>	unsets MAIL	unsets MAIL
<code>echo 42 *2</code>	42 42	42 42
<code>echo a'b'c'd'e'f'g'h'i'j'k'l'm'n'o'p'q'r's't</code>	abcdefghijklmnopqrstuvwxyz	abcdefghijklmnopqrstuvwxyz
<code>echo \$USER\$USER</code>	damachad\$USER	damachad\$USER
<code>cd -</code>	returns prompt	returns prompt
<code>export teste="a b c"; echo \$teste</code>	a b c	a b c
EXPANDER		
<code>echo \$USER</code>	prints username	prints username
<code>export ABC=\$USER</code>	creates var ABC=username	creates var ABC=username
EXIT CODES		
<code>cat nonexistent file or folder; echo \$?</code>	exit code 1	exit code 1
<code>enter /.minishell; exit 300</code>	exit code 44	exit code 44
<code>./not_executable_file</code>	exit code 126	exit code 126
<code><not_existing_file</code>	exit code 1	exit code 1
<code> echo 1</code>	exit code 2	exit code 2
<code>ls not_existing_file</code>	exit code 2	exit code 2
PIPES		
<code>"ls" wc`</code>	execute ls and wc	execute ls and wc
<code>unset USER env grep USER</code>	USER=arepsa (doesn't unset variable)	USER=arepsa (doesn't unset variable)
<code>unset OLDPWD ls</code>	performs ls, doesn't unset variable	performs ls, doesn't unset variable
<code>echo USER export USER=bubu</code>	echo doesn't perform, USER is not changed, returns prompt	echo doesn't perform, USER is not changed, returns prompt
<code>echo Hello World grep a</code>	bash: syntax error near unexpected token ` '	minishell: syntax error near unexpected token
<code>cat cat ls</code>	performs ls and exits after 2 enters	performs ls and exits after 2 enters
<code>cat /dev/urandom head -c 100</code>	returns 100 random characters	returns 100 random characters
REDIRECTS		
<code>echo hello > outfile</code>		

ls "\$"ola	ls: cannot access '\$': No such file or directory	ls: cannot access '\$': No such file or directory		
< inaccessible_file cat -e > output	ls: cannot access 'ola': No such file or directory creates empty output file	ls: cannot access 'ola': No such file or directory		
<	bash: syntax error near unexpected token 'newline'	minishell: syntax error near unexpected token		
>	bash: syntax error near unexpected token 'newline'	minishell: syntax error near unexpected token		
> test	creates empty output file	creates empty output file		
< existingfile	Does nothing	Does nothing		
< testfile > testfile2	creates empty output file "testfile2"	creates empty output file "testfile2"		
HEREDOCs				
<<	bash: syntax error near unexpected token 'newline'	minishell: syntax error near unexpected token		
>>	bash: syntax error near unexpected token 'newline'	minishell: syntax error near unexpected token		
cat << end wc -l	count lines of heredoc file	count lines of heredoc file		
cat << end grep e	returns heredoc lines that contain e	returns heredoc lines that contain e		
<< end grep e	returns prompt	returns prompt		
<< EOF cat	expands any variable inside it	expands any variable inside it		
<< "EOF" cat	Does not expand	Does not expand		
<< 'EOF' cat	Does not expand	Does not expand		
<< "EOF" cat	unclosed quotes	mensagem de erro de: unclosed quotes		
DENTRO de HEREDOCs				
\$USER e \$CASA e "\$ HOME" e "\$ HOME" "\$HOME" "\$HOME" \$HOME	(expand)USER e e "\$ HOME e "\$ HOME" "(expand)HOME" "(expand)HOME" (expand)HOME (expand)HOME	(expand)USER e e "\$ HOME e "\$ HOME" "(expand)HOME" "(expand)HOME" (expand)HOME (expand)HOME		
\$CASA \$ CAMA	\$ CAMA	\$ CAMA		
"\$CASA" "\$ CAMA" "\$USER" "\$HOME" \$USER \$ "\$ HOME" << HOME	" "\$ CAMA" (expand)"USER" (expand)HOME (expand)USER \$ "\$ HOME" << HOME	" "\$ CAMA" (expand)"USER" (expand)HOME (expand)USER \$ "\$ HOME" << HOME		
COMMANDS				
ls \$jrgsdhgb	Executes ls	Executes ls		
unset PATH; ls	bash: ls: No such file or directory	minishell: ls: command not found		
/dev/urandom	bash: /dev/urandom: Permission denied	minishell: /dev/urandom: Permission denied		
\$<something that does not exist>	returns prompt	returns prompt		
TO SOLVE				
export test="ls -l"; \$test	runs ls -l	runs ls -l		
---	Command " not found, but can be installed with:	minishell: : command not found		
<< eof; Ctrl + C	returns prompt	returns prompt		
\$PWD	bash: <PWD>: Is a directory	execve: Permission denied		
cat /bin/ls				
/bin/ls wc				
ping 8.8.8.8				
export test; export test=hello; echo \$test	hello	hello		
---	Command " not found	minishell: : command not found		
"" ls	executes ls, and shows "command not found"	shows "command not found" then executes ls		
./srcs (execute existing folder)	bash: ./srcs: Is a directory	minishell: ./srcs: is a directory		
a= ; echo \$?	exit code 0	exit code 0		
export us\$er=blabla				
unset weird"%chars				
cd FILENAME WITH SPACE				
echo hello > FILENAME WITH SPACE				