Notes

Look for my notes here. (Click or hover.)

Notes

william smith

jamf | @talkingmoose

Code snippets

jamf.it/asg

Notes

Notes

apple-smart-battery.zsh

Shell Script | Language | Run | Stop | Run Settings... | Back/Forward | View

apple-smart-battery.zsh

```
1  #!/bin/zsh
2
3  ioreg -r -c "AppleSmartBattery"
```

Filter | Program Output

```
    "MaxCapacity" = 100
    "InstantAmperage" = 0
    "PortControllerInfo" =
      ({"PortControllerLoserReason"=1,"PortCon
      trollerEvtBuffer"=<0000000000$
    "GasGaugeFirmwareVersion" = 2
    "AdapterInfo" = 0
    "Location" = 0
    "Temperature" = 3037
    "AvgTimeToEmpty" = 65535
    "BestAdapterIndex" = 0
    "DesignCapacity" = 6075
    "IsCharging" = No
    "PermanentFailureStatus" = 0
    "Voltage" = 12599
    "UserVisiblePathUpdated" = 1719852672
    "CycleCount" = 116
    "AppleRawMaxCapacity" = 5005
    "VirtualTemperature" = 3050
}
```

Run Succeeded | Time 21 ms | Peak Memory 4.3M | Symbol | Tabs: 4 | Line 3, Col

Notes

apple-smart-battery.zsh

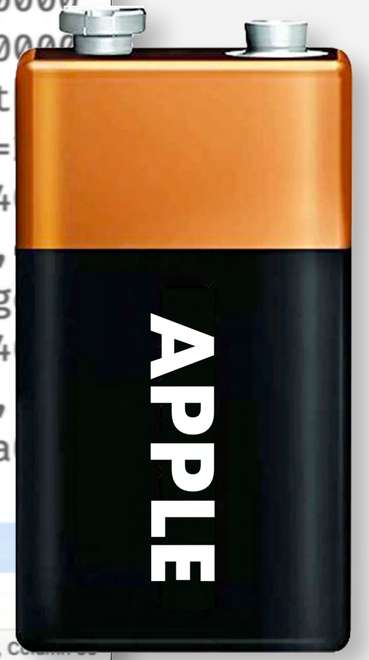
Shell Script | Language | Run | Stop | Run Settings... | Back/Forward | View

apple-smart-battery.zsh

```
1  #!/bin/zsh
2
3  ioreg -r -c "AppleSmartBattery" | grep -w
       "CycleCount"
```

Filter | Program Output

```
99,"SystemPower"=1099365920,"LifetimeDat
a"={"Raw"=<04f95a0e000099e000000000000
00004f963ac0000c3e44092000000000000250
00410ed0c7c32c427261842ea471dfde771ea47e
77100ee0005a8d50000>,"UpdateTime"=171985
2732,"ResistanceUpdatedDisabledCount"=0,
"CycleCountLastQmax"=116,"TimeAtHighSoc"
=<000000094400000ef0800000000000000000
000000000000000000000000f7000000180000
000000000000000000000000000000000000000
000902000016000000000000000000000000000
000000000017000000980c00002a0100000000
000000000000000000000000>,"Temperat
Samples"=370901,"TotalOperatingTime"=
81,"MaximumDischargeCurrent"=18446744
709546055,"MinimumPackVoltage"=10022,
ximumPackVoltage"=12996,"MaximumCharg
rrent"=6210,"AverageTemperature"=1844
4073709551598,"MinimumTemperature"=4,
ISCnt"=0,"MaximumTemperature"=37},"Ra
=56}
"CycleCount" = 116
```



Run Succeeded | Time 23 ms | Peak Memory 4.1M | Symbol | Tabs: 4 | Line 3, Col

Notes

apple-smart-battery.zsh

Shell Script | Language | Run | Stop | Run Settings... | Back/Forward | View

apple-smart-battery.zsh

```
1  #!/bin/zsh
2
3  ioreg -r -c "AppleSmartBattery" | grep -w
       "CycleCount" | awk '{ print $3 }'
```

Filter | Program Output

```
ed"=3256,"ITMiscStatus"=0,"StateOfCharge"=99
,"Ra09"=69,"GaugeFlagRaw"=224,"CycleCount"=1
16,"Voltage"=12599,"SystemPower"=1099365920,
"LifetimeData"={"Raw"=<04f95a0e0000099e00000
0000000000004f963ac0000c3e44092000000000000
025000410ed0c7c32c427261842ea471dfde771ea47e
77100ee0005a8d50000>,"UpdateTime"=1719852732
,"ResistanceUpdatedDisabledCount"=0,"CycleCo
untLastQmax"=116,"TimeAtHighSoc"=<0000000094
400000ef0800000000000000000000000000000000
0000000000f7000001800000000000000000000000
0000000000000000000000902000160000000000
0000000000000000000000017000000980c000
0100000000000000000000000000000000>,"
peratureSamples"=370901,"TotalOperatingTi
=23181,"MaximumDischargeCurrent"=18446744
709546055,"MinimumPackVoltage"=10022,"Max
mPackVoltage"=12996,"MaximumChargeCurrent
210,"AverageTemperature"=1844674407370955
8,"MinimumTemperature"=4,"RDISCnt"=0,"Max
mTemperature"=37},"Ra02"=56}

116
```

Run Succeeded | Time 24 ms | Peak Memory 4.0M | Symbol | Tabs: 4 | Line 3, Col

Notes

apple-smart-battery.zsh

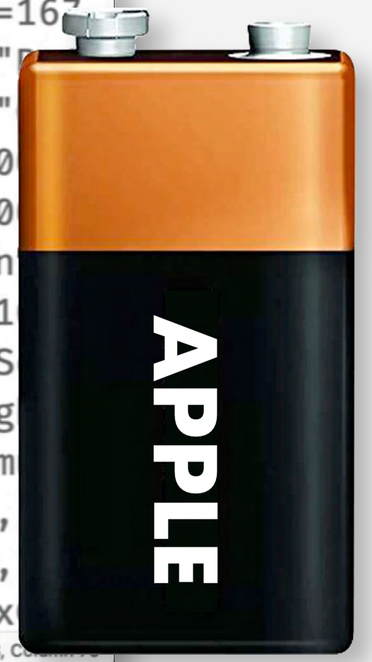
Shell Script | Language | Run | Stop | Run Settings... | Back/Forward | View

apple-smart-battery.zsh

```
1  #!/bin/zsh
2
3  ioreg -r -c "AppleSmartBattery" | grep -w
       "CycleCount" | awk '{ print $3 }'
```

Filter | Program Output

```
{"Ra03"=56,"Ra10"=69,"CellWom"=(0,0),"RaTableRaw
    "=(<000000cf00360036003400 4d0030003d003e0045
    0046004700 52006f00ff01b7>,<000000cb003800340
    034004b002f003a003b0040003f003e0056006a00fc0
    1a8>,<005500e0003a00380038005000 33003e003f00
    430045004500550074011c01d2>),"Qstart"=0,"Ada
    pterPower"=1097635302,"TrueRemainingCapacity
    "=0,"DailyMinSoc"=99,"Ra04"=80,"CurrentSense
    MonitorStatus"=0,"Ra11"=85,"CellVoltage"=(42
    00,4199,4199),"PackCurrentAccumulator"=18446
    744073709551240,"PassedCharge"=0,"Flags"=167
    77729,"PresentDOD"=(10,10,10),"Ra05"=51,"R
    2"=116,"MiscStatus"=136,"FccComp1"=5511,"
    mID"=20882,"iMaxAndSocSmoothTable"=<00000
    0000000000000000000000000000000000000
    000000000>,"FccComp2"=5005,"PackCurren
    cumulatorCount"=76698,"DOD0"=(1648,1648,1
    ),"Dod0AtQualifiedQmax"=0,"Ra06"=62,"ResS
    e"=0,"Ra13"=284,"FilteredCurrent"=0,"Weig
    dRa"=(68,67,72),"RSS"=0,"CellCurrentAccum
    torCount"=0,"Serial"="F8Y144209JRQ1LTA4",
    taFlashWriteCount"=9118,"DailyMaxSoc"=99,
    teOfFirstUse"=0,"Ra07"=63,"Ra14"=466,"Max
```



Run Succeeded | Time 24 ms | Peak Memory 4.0M | Symbol | Tabs: 4 | Line 3, Col

Notes

apple-smart-battery.zsh

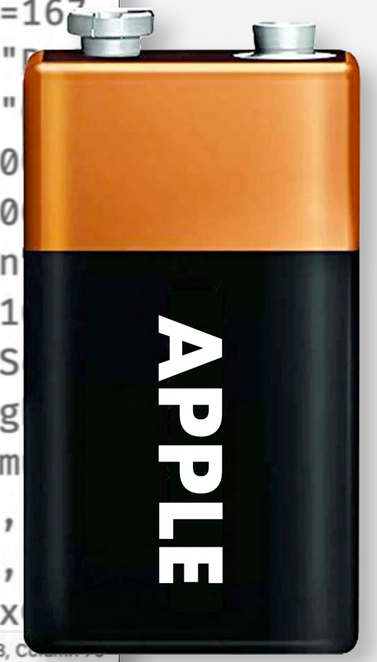
Shell Script | Language | Run | Stop | Run Settings... | Back/Forward | View

apple-smart-battery.zsh

```
1  #!/bin/zsh
2
3  ioreg -r -c "AppleSmartBattery" | grep -w
       "CycleCount" | awk '{ print $3 }'
```

Filter | Program Output

{"Ra03"=56,"Ra10"=69,"CellWom"=(0,0),"RaTableRaw"=<000000cf00360036003400 4d0030003d003e0045004600470052006f00ff01b7>,<000000cb003800340034004b002f003a003b0040003f003e0056006a00fc01a8>,<005500e0003a003800380050003300 3e003f00430045004500550074011c01d2>),"Qstart"=0,"AdapterPower"=1097635302,"TrueRemainingCapacity"=0,"DailyMinSoc"=99,"Ra04"=80,"CurrentSenseMonitorStatus"=0,"Ra11"=85,"CellVoltage"=(4200,4199,4199),"PackCurrentAccumulator"=18446744073709551240,"PassedCharge"=0,"Flags"=16377729,"PresentDOD"=(10,10,10),"Ra05"=51,"R2"=116,"MiscStatus"=136,"FccComp1"=5511,"mID"=20882,"iMaxAndSocSmoothTable"=<00000000000000000000000000000000000000000000000000000000000000000000000000000000>,"FccComp2"=5005,"PackCurrencumulatorCount"=76698,"DOD0"=(1648,1648,1),"Dod0AtQualifiedQmax"=0,"Ra06"=62,"ResSe"=0,"Ra13"=284,"FilteredCurrent"=0,"WeigdRa"=(68,67,72),"RSS"=0,"CellCurrentAccumtorCount"=0,"Serial"="F8Y144209JRQ1LTA4",taFlashWriteCount"=9118,"DailyMaxSoc"=99,teOfFirstUse"=0,"Ra07"=63,"Ra14"=466,"Max



Run Succeeded | Time 24 ms | Peak Memory 4.0M | Symbol | Tabs: 4 | Line 3, Col

Notes

apple-smart-battery.zsh

Shell Script | Language | Run | Stop | Run Settings... | Back/Forward | View

apple-smart-battery.zsh

```
1  #!/bin/zsh
2
3  ioreg -r -c "AppleSmartBattery" | grep -w
       "CycleCount" | awk '{ print $3 }' |
       sed /{.*}/d
```

Filter | Program Output

```
116
```

Run Succeeded | Time 27 ms | Peak Memory 3.9M | Symbol | Tabs: 4 | Line 3, Col

Notes

apple-smart-battery.zsh

Shell Script | Language | Run | Stop | Run Settings... | Back/Forward | View

apple-smart-battery.zsh

```
1  #!/bin/zsh
2
3  ioreg -r -c "AppleSmartBattery" | grep -w
     "CycleCount" | awk '{ print $3 }' |
     sed 's/{.*}//d'
```

Filter | Program Output

```
116
```

Run Succeeded | Time 27 ms | Peak Memory 3.9M | Symbol | Tabs: 4 | Line 3, Col

Notes

apple-smart-battery.zsh

Shell Script | Language | Run | Stop | Run Settings... | Back/Forward | View

apple-smart-battery.zsh

```
1  #!/bin/zsh
2
3  ioreg -r -c "AppleSmartBattery" | grep -w
       "CycleCount" | awk '{ print $3 }' |
       sed /{.*}/d
```

Filter | Program Output

```
116
```

Run Succeeded | Time 27 ms | Peak Memory 3.9M | Symbol | Tabs: 4 | Line 3, Col

Notes

```
ioreg -r -c "AppleSmartBattery" | grep -w "CycleCount" | awk '{ print $3 }' | sed /{.*}/d
```

Notes

```
ioreg -r -c "AppleSmartBattery" | grep -w "CycleCount" | awk '{ print $3 }' | sed /{.*}/d
```

just awk:

```
ioreg -r -c "AppleSmartBattery" | awk -F ' = ' '/"CycleCount" = / { print $2 }'
```

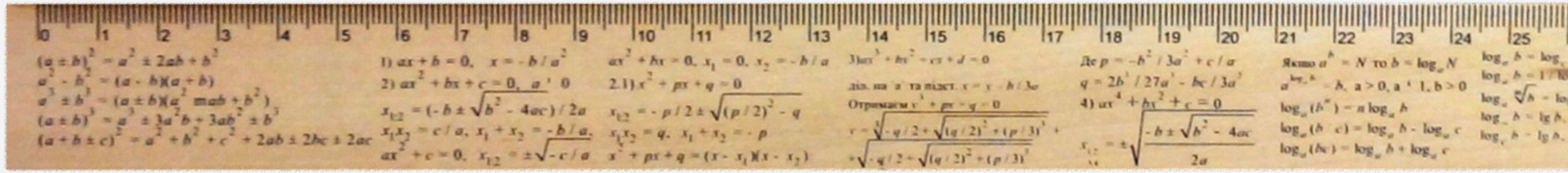
just sed:

```
ioreg -r -c "AppleSmartBattery" | sed -e '/"CycleCount" =/!d' -e 's/.* = //'
```

just grep:

```
ioreg -r -c "AppleSmartBattery" | grep -e "\"CycleCount\" = " | grep -o "\d*"
```

```
ioreg -l | grep -e "\"CycleCount\" = " | grep -o "\d*"
```



Notes

NASA

Guidance computer

Sent mankind to the moon on Apollo 11 spacecraft in **1969**

12,250 flops/sec

Notes

Cray-2

Supercomputer

The most power computer built in **1985**

1.9 billion flops/sec

Notes

Smartphone

Pocket computer

Today’s most ubiquitous computer platform

2 teraflops/sec

Notes

Macbook Pro

Laptop computer

Today’s high end Apple product for designers, scientists, and engineers in **2024**

4.6 teraflops/sec

Notes

Xbox Series X

Gaming system

Microsoft's premium out-of-the-box gaming system introduced **2020**

12 teraflops/sec

Notes

Frontier

Modern supercomputer

HPE's Cray EX supercomputer rated as the fastest computer in the world in **2022**

1.102 exaflops/sec

Notes

flops / sec

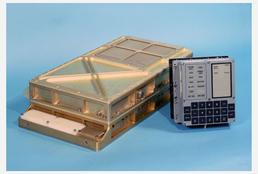
12000000000000

9000000000000

6000000000000

3000000000000

0



NASA
Guidance Computer
1969

Cray-2
Supercomputer
1985

Smartphone
2024

MacBook Pro
2024

Xbox
Series X
2024

Notes

```
ioreg -r -c "AppleSmartBattery" | grep -w "CycleCount" | awk '{ print $3 }' | sed /{.*}/d
```

Notes

- Origins
- What they have in common
- When to use each
- Syntax

Notes

ed

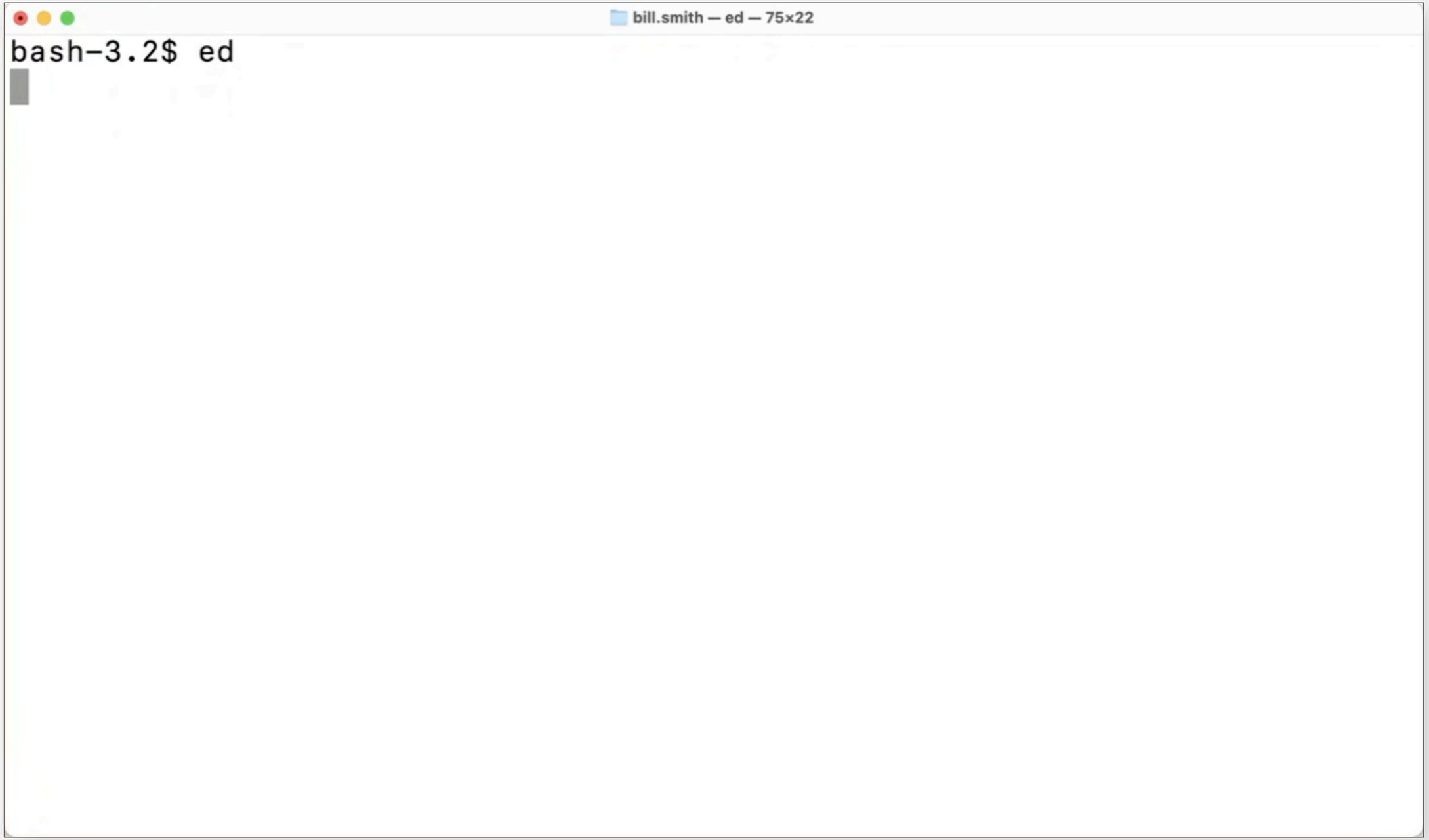
‘The most user-hostile editor ever created’

– Peter H. Salus, computer historian

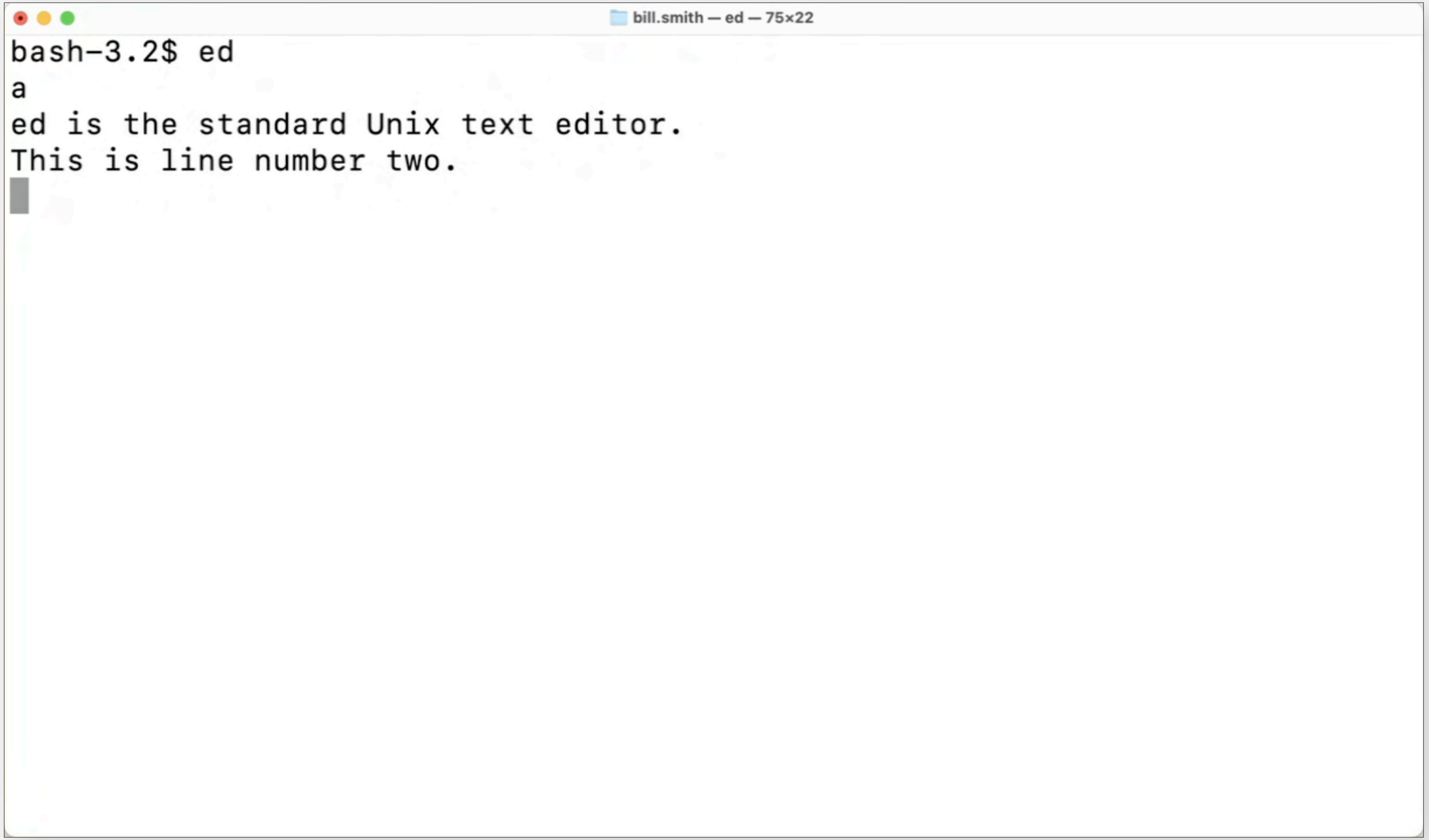
Notes

Notes

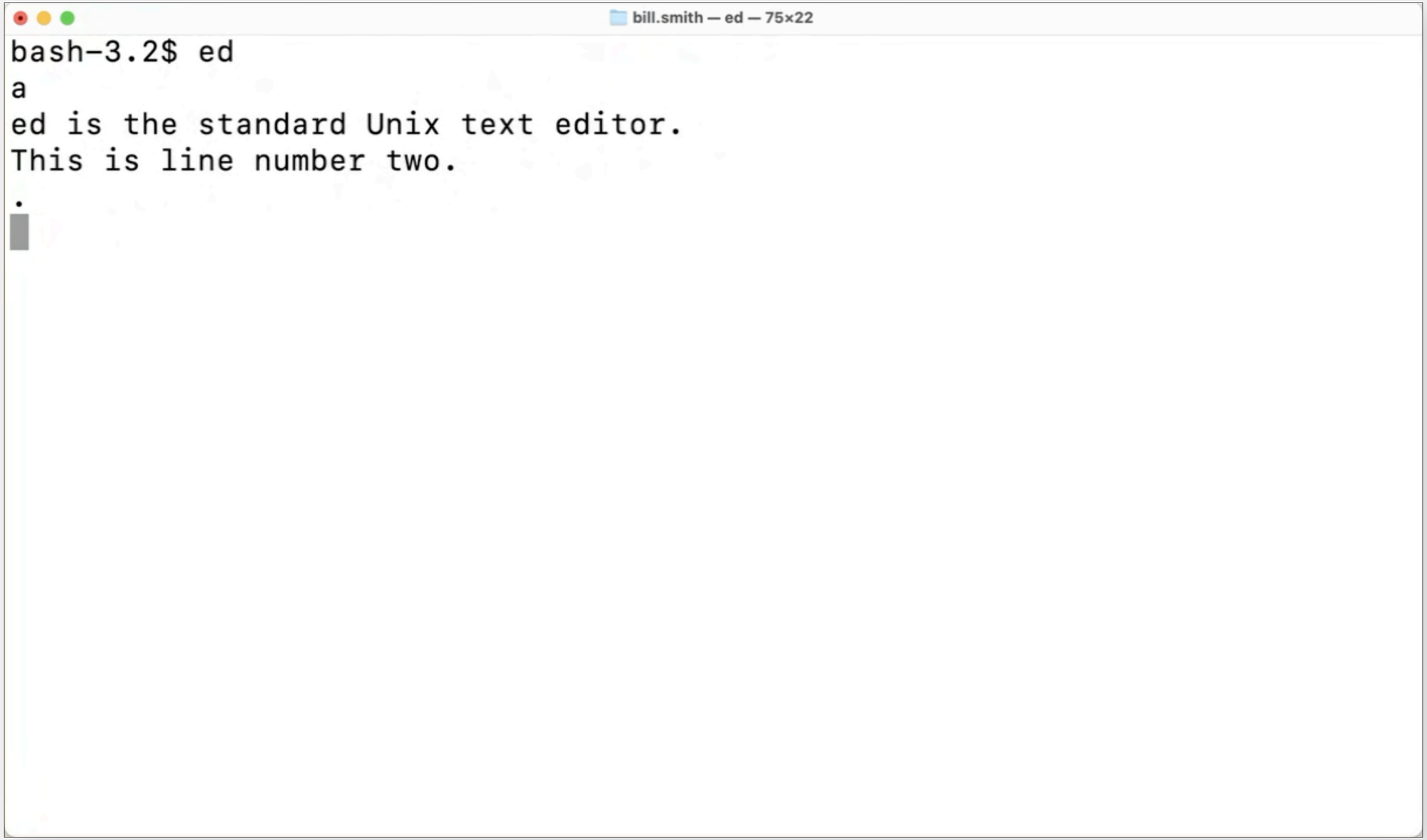
Notes



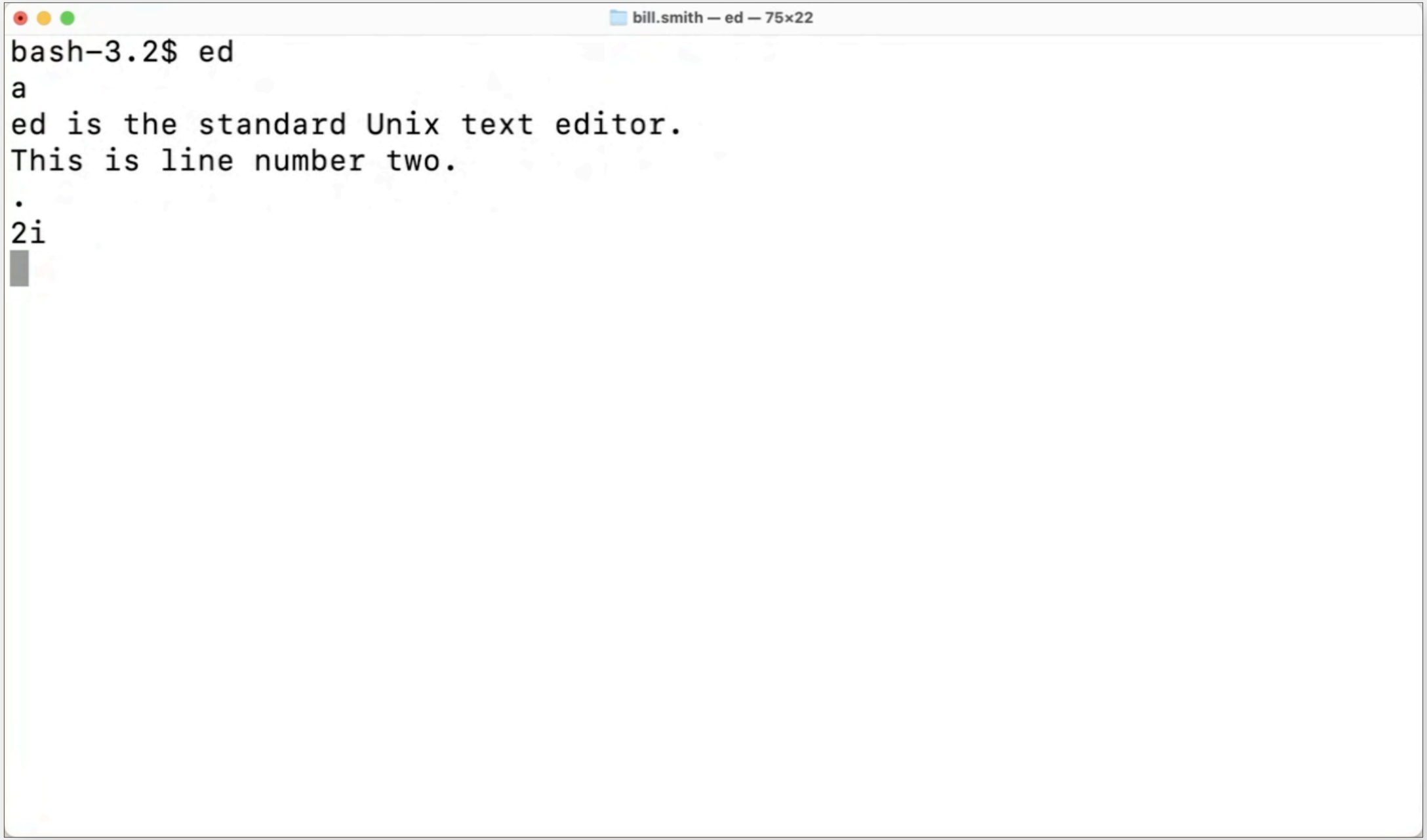
Notes


bill.smith — ed — 75×22
bash-3.2$ ed
a
ed is the standard Unix text editor.
This is line number two.

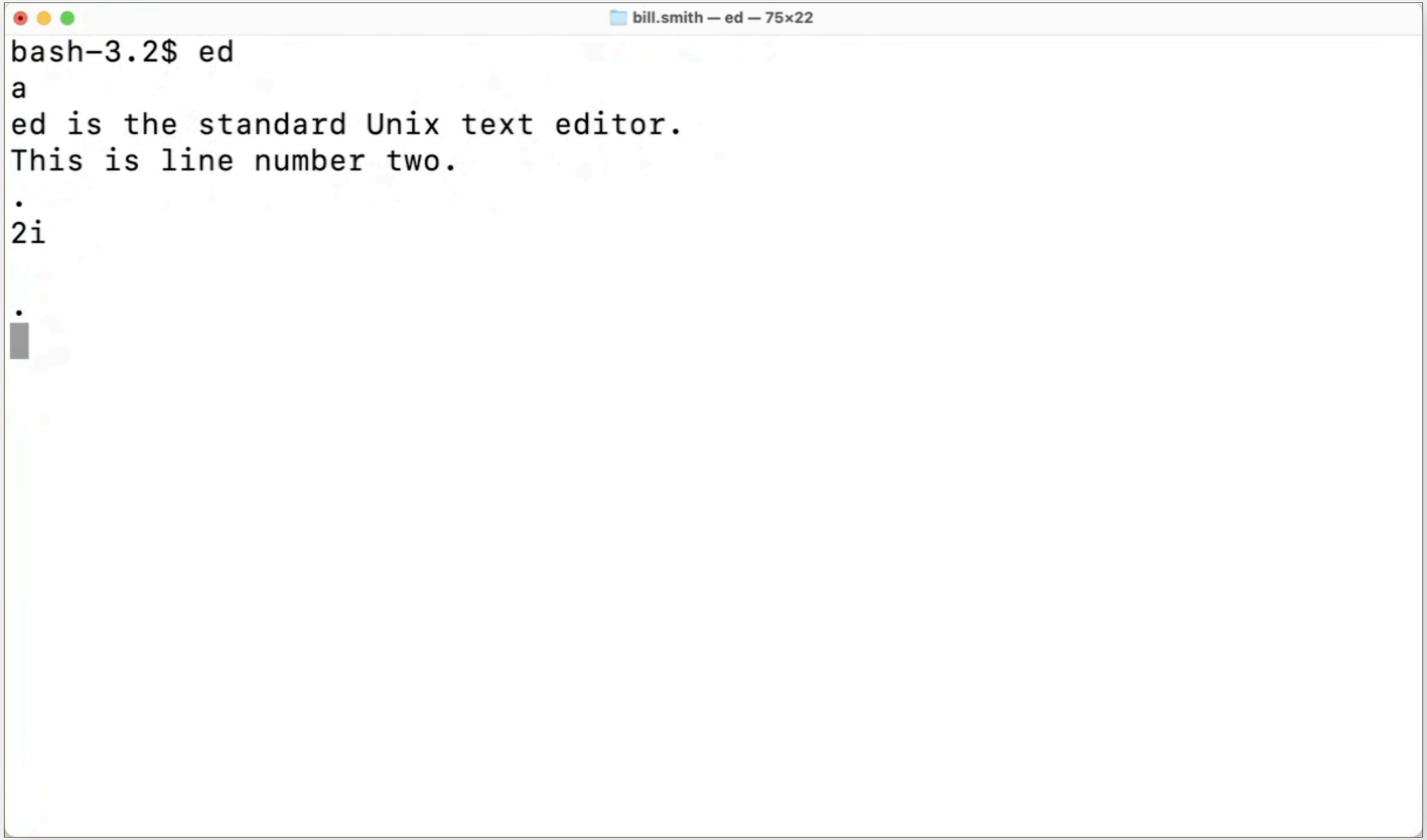
Notes


bill.smith — ed — 75×22
bash-3.2$ ed
a
ed is the standard Unix text editor.
This is line number two.
.

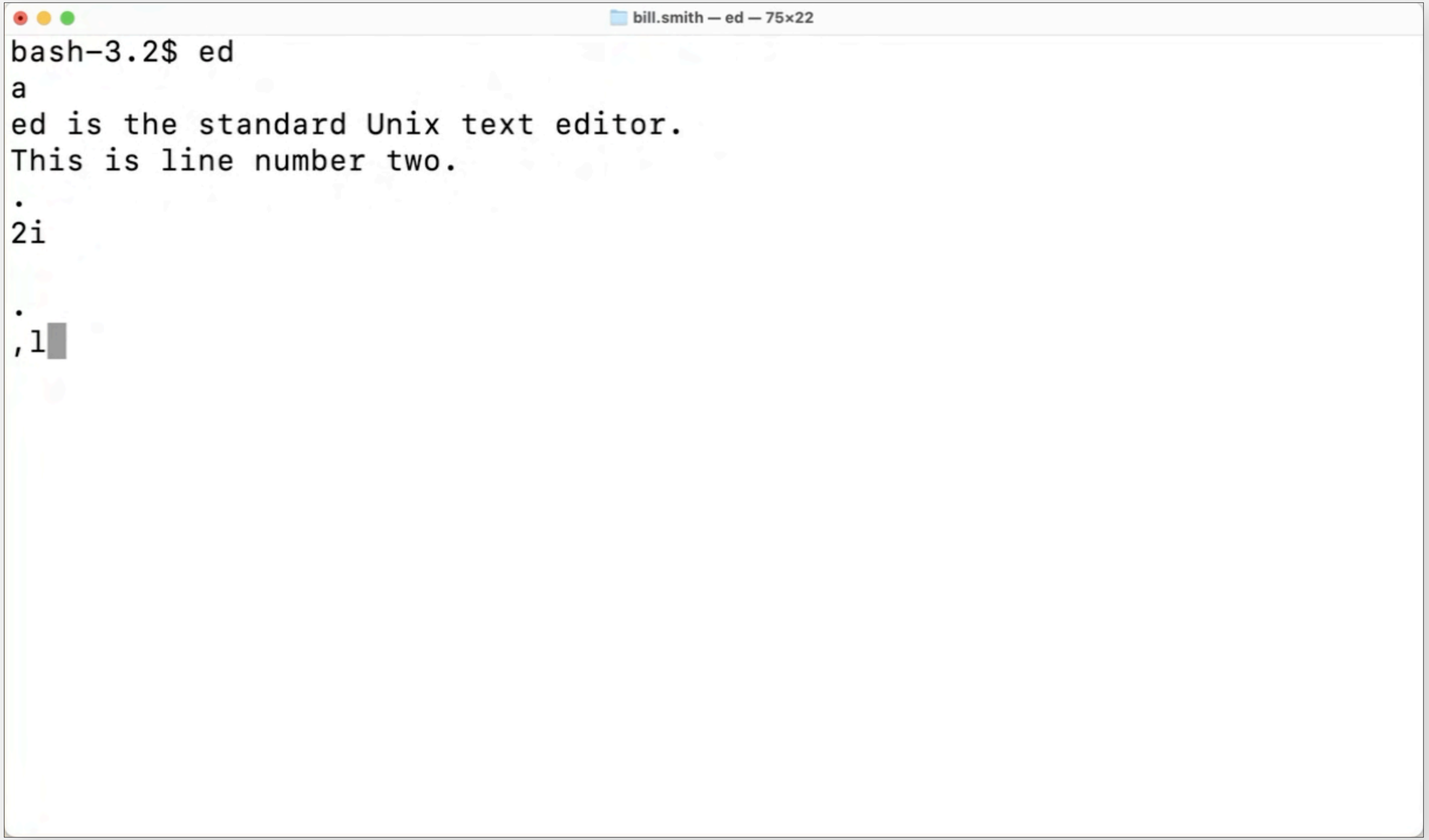
Notes


bill.smith — ed — 75×22
bash-3.2$ ed
a
ed is the standard Unix text editor.
This is line number two.
.
2i

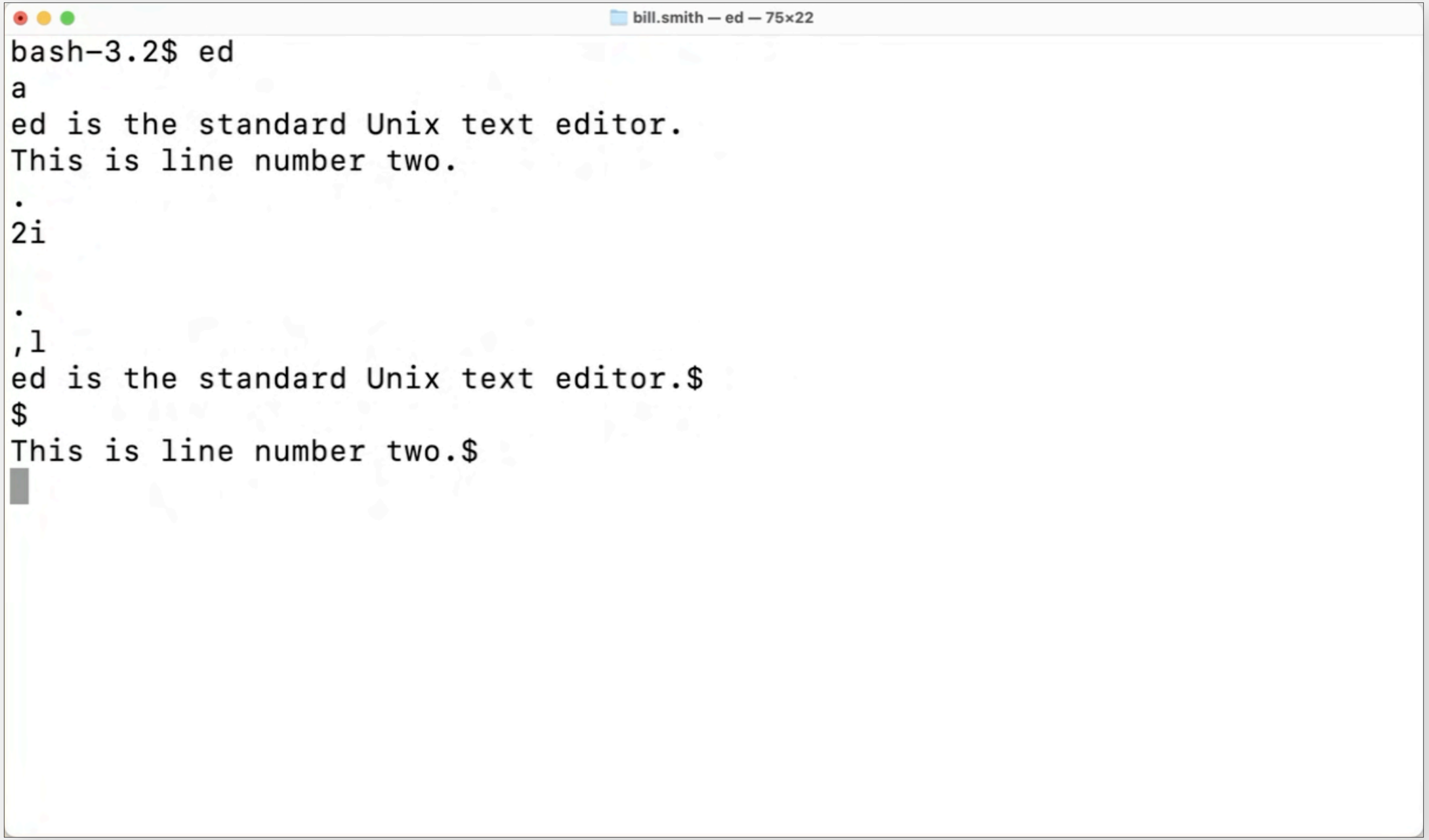
Notes


bill.smith — ed — 75×22
bash-3.2$ ed
a
ed is the standard Unix text editor.
This is line number two.
.
2i
.

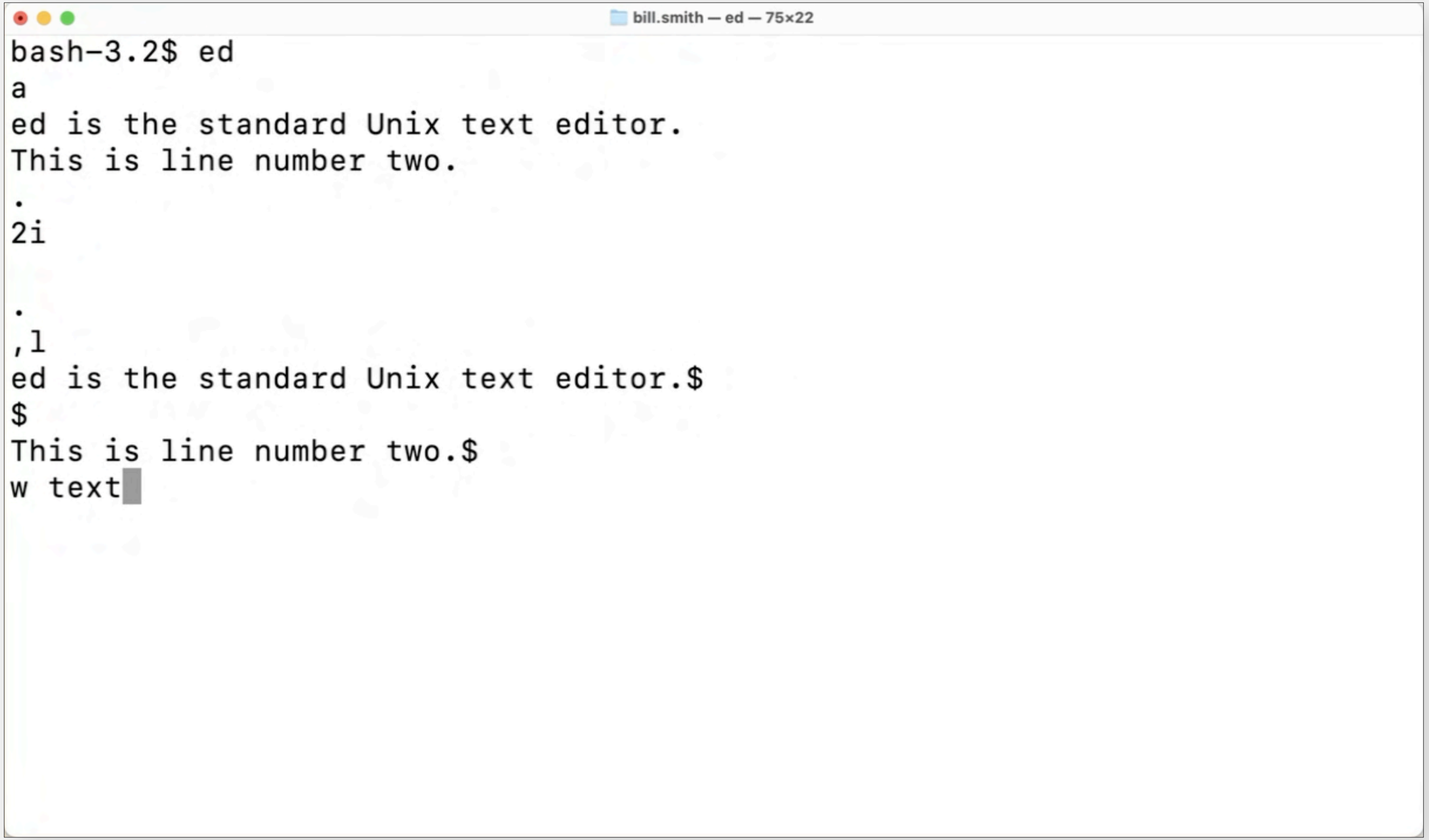
Notes


bill.smith — ed — 75×22
bash-3.2$ ed
a
ed is the standard Unix text editor.
This is line number two.
.
2i
.
,l

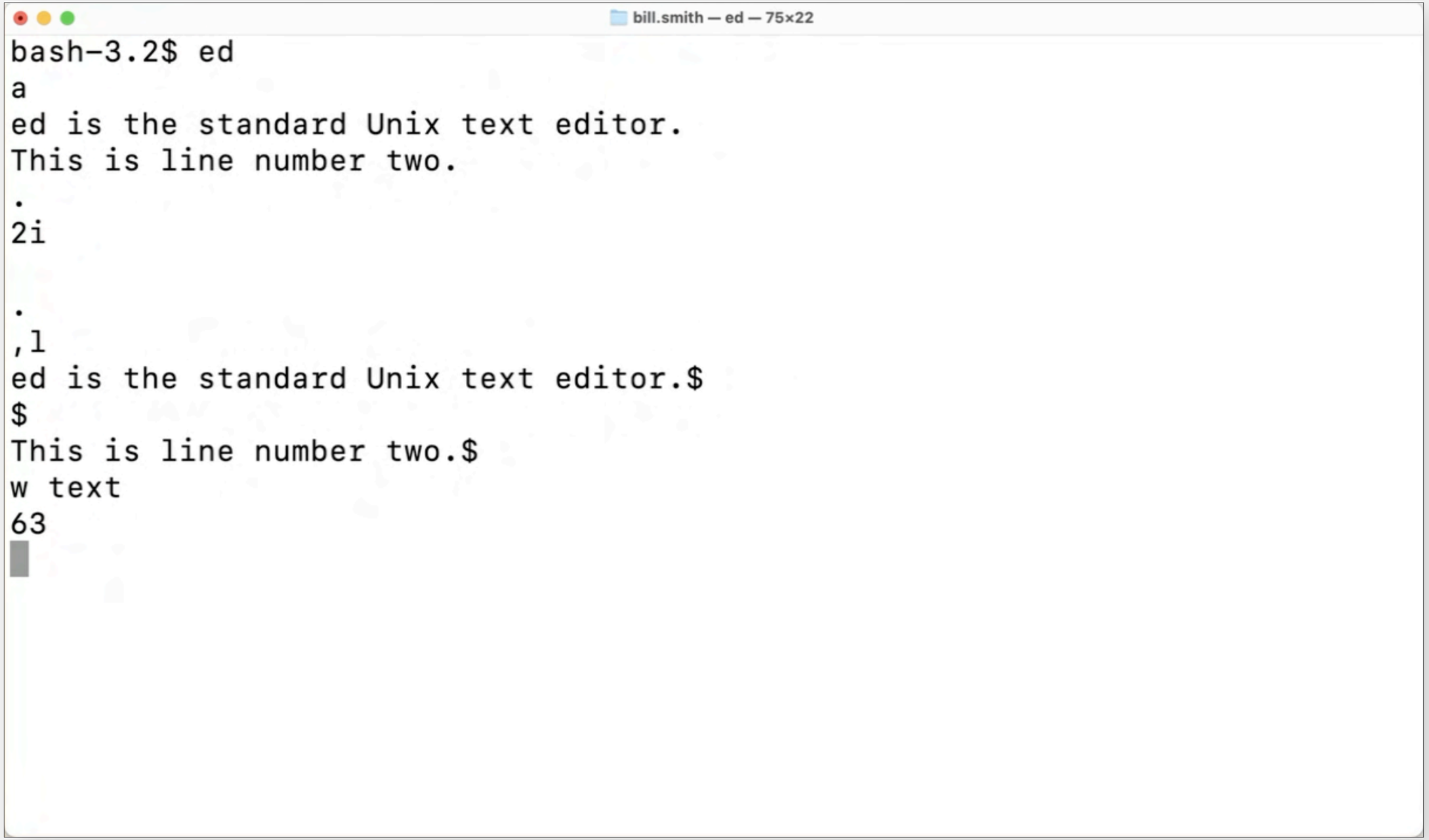
Notes


bill.smith — ed — 75×22
bash-3.2$ ed
a
ed is the standard Unix text editor.
This is line number two.
.
2i
.
,l
ed is the standard Unix text editor.$
$
This is line number two.$

Notes

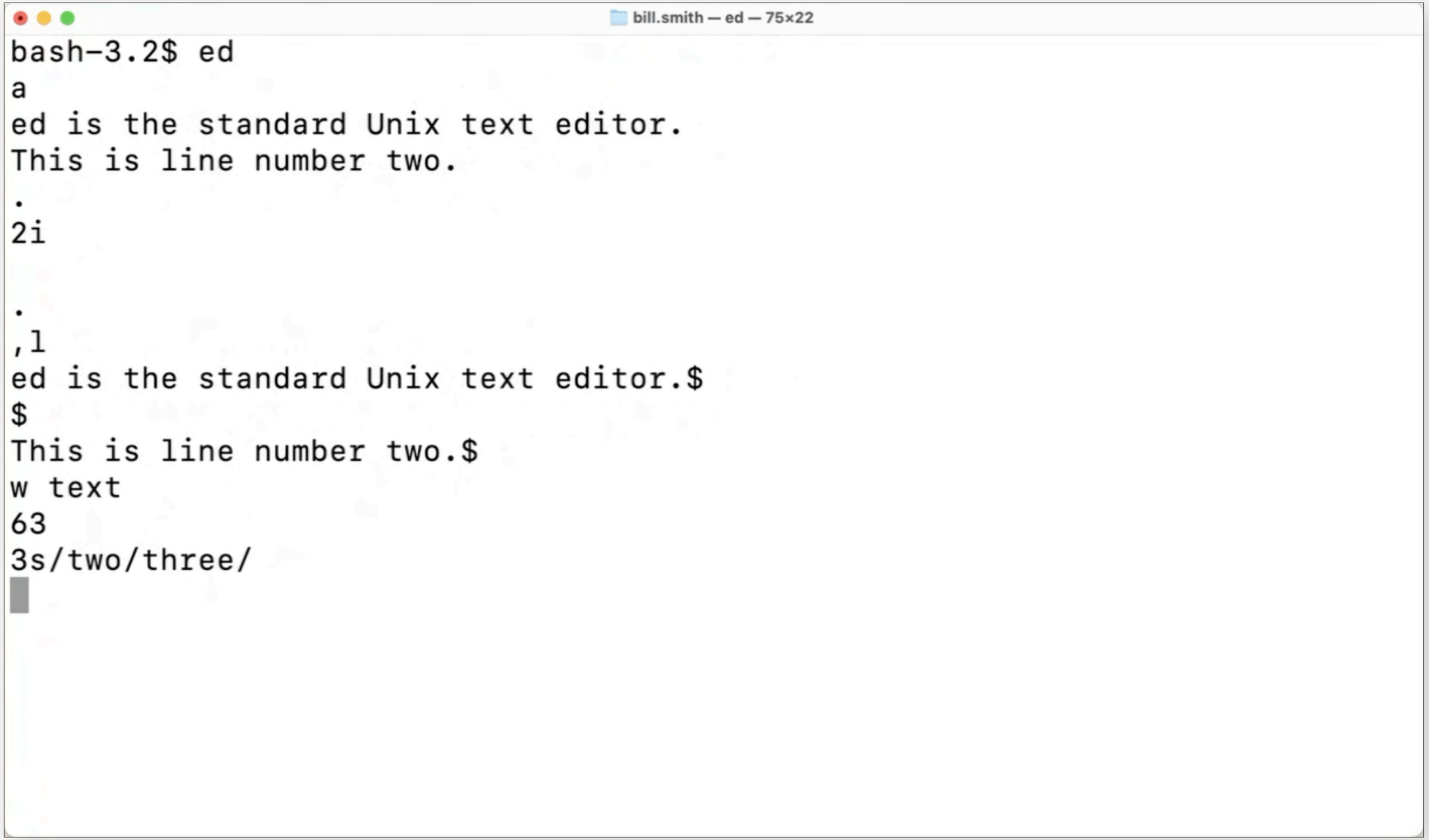

bill.smith — ed — 75×22
bash-3.2$ ed
a
ed is the standard Unix text editor.
This is line number two.
.
2i
.
,l
ed is the standard Unix text editor.$
$
This is line number two.$
w text

Notes


bill.smith — ed — 75×22
bash-3.2$ ed
a
ed is the standard Unix text editor.
This is line number two.
.
2i

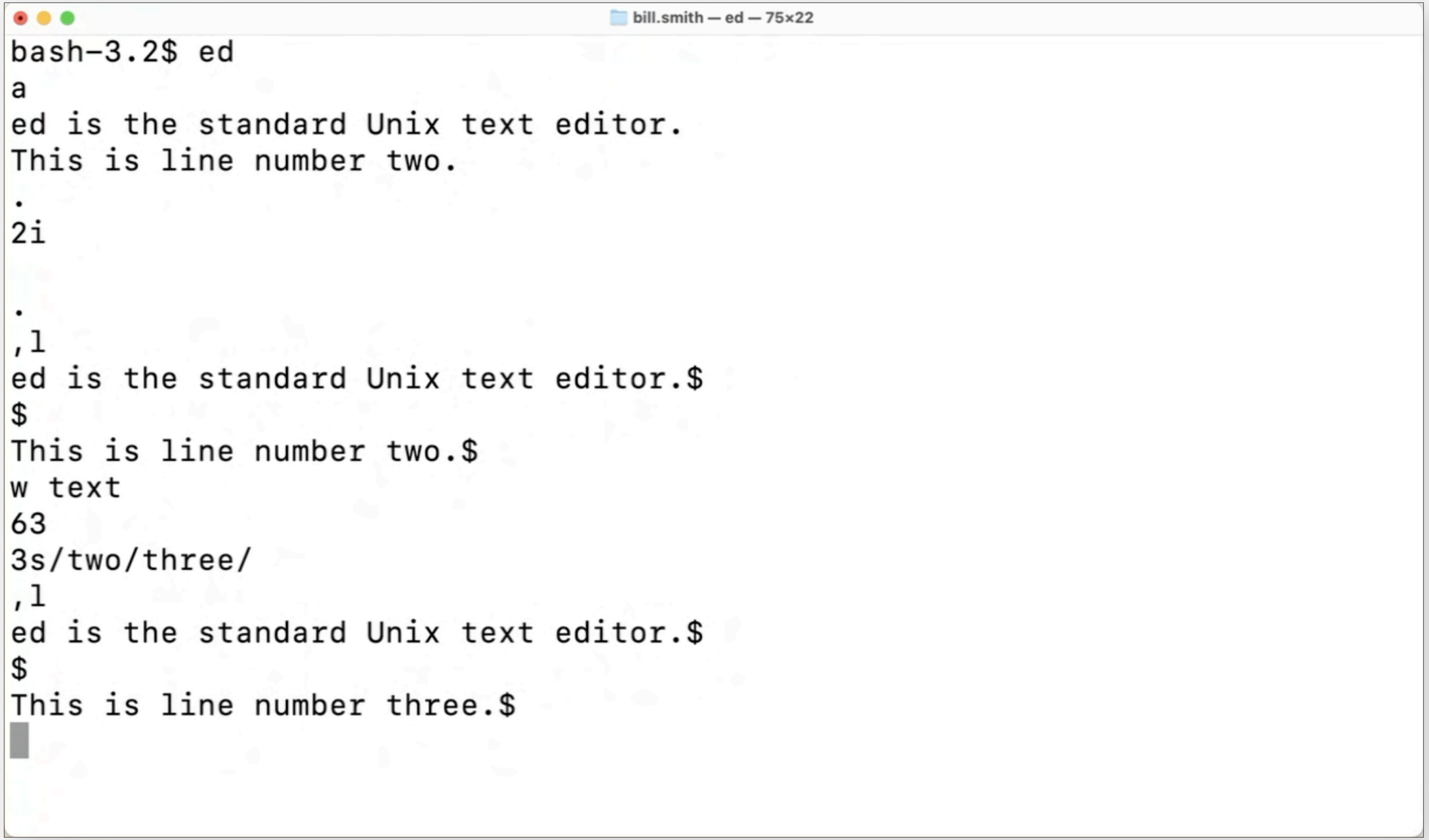
.
,l
ed is the standard Unix text editor.$
$
This is line number two.$
w text
63

Notes


bill.smith — ed — 75×22
bash-3.2$ ed
a
ed is the standard Unix text editor.
This is line number two.
.
2i

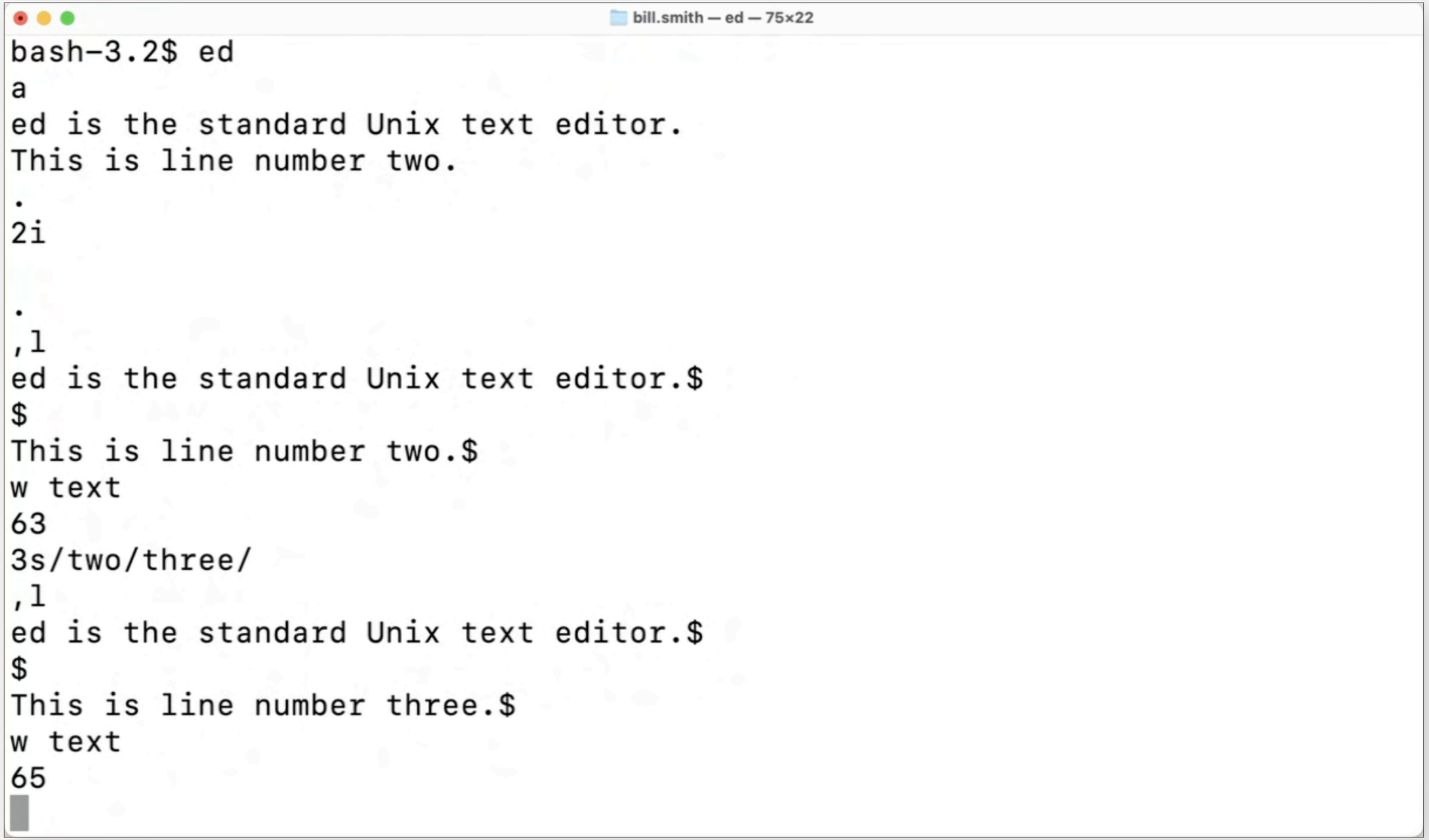
.
,l
ed is the standard Unix text editor.$
$
This is line number two.$
w text
63
3s/two/three/

Notes


bill.smith — ed — 75×22
bash-3.2$ ed
a
ed is the standard Unix text editor.
This is line number two.
.
2i

.
,l
ed is the standard Unix text editor.$
$
This is line number two.$
w text
63
3s/two/three/
,l
ed is the standard Unix text editor.$
$
This is line number three.$

Notes


bill.smith — ed — 75×22
bash-3.2$ ed
a
ed is the standard Unix text editor.
This is line number two.
.
2i

.
,l
ed is the standard Unix text editor.$
$
This is line number two.$
w text
63
3s/two/three/
,l
ed is the standard Unix text editor.$
$
This is line number three.$
w text
65

Notes

? = error

‘The experienced user will know what is wrong.’


JANUS 71

Notes

non-interactive

‘Watch your step.’


ZORK is a registered trademark of Infocom, Inc.
Revision 88 / Serial number 840726
West of House
You are standing in an open field west of a white house, with a boarded front door.
There is a small mailbox here.
>open mailbox
West of House
Score: 0
Moves: 1
Opening the small mailbox reveals a leaflet.
>read leaflet
West of House
Score: 0
Moves: 2
(Taken)
"WELCOME TO ZORK!
ZORK is a game of adventure, danger, and low cunning. In it you will explore some of the most amazing territory ever seen by mortals. No computer should be without one!"
>
TeleVideo

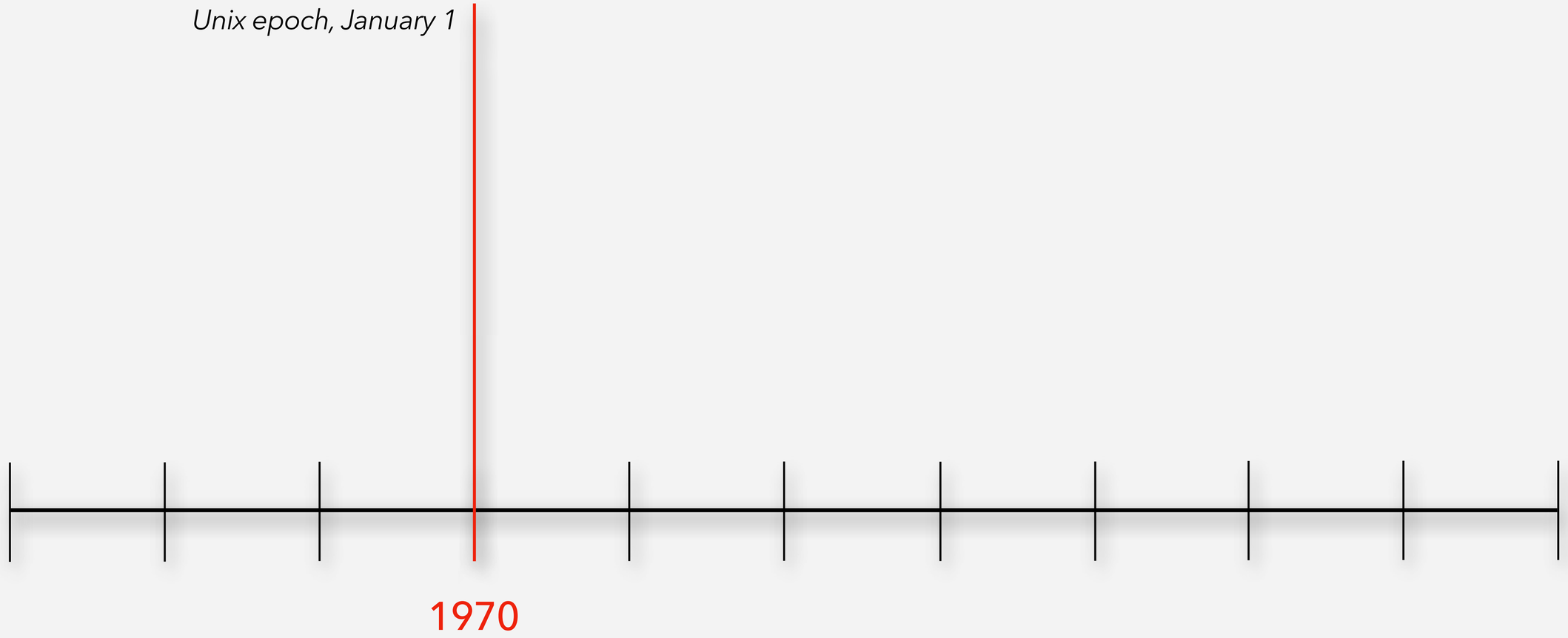

LIBRARY Creates and alters object and macro libraries
LINK Produces an executable program
LOAD Makes a device handler permanently resident in memory
MACRO Invokes the macro assembler
PRINT Prints files on the line printer
R Loads and executes a memory image file
REENTER Starts a program at its reentry address
REMOVE Removes a device handler from the system
RENAME Changes the name of a file
RESET Causes a general system reset
RESUME Resumes execution of a foreground or system job
RUN Loads and starts a program
SAVE Writes memory areas to a file
SET Controls various system options
SHOW Displays system hardware and software status
SQUEEZE Rearranges disk files to collect unused file space
SRUN Loads and starts a system job
START Initiates the program in memory
SUSPEND Stops execution of the foreground or system job
TIME Sets or displays the system time
TYPE Outputs files to the terminal
UNLOAD Removes a resident device handler from memory
.
SET-UP
ESC
TAB
BACK SPACE
BREAK
DELETE

Notes

interactive

‘The world is your burrito.’

Notes



Notes

Unix epoch, January 1

ed

Ken Thompson

First Unix editor

1969 1970

Notes

Unix epoch, January 1

ed

Ken Thompson

First Unix editor

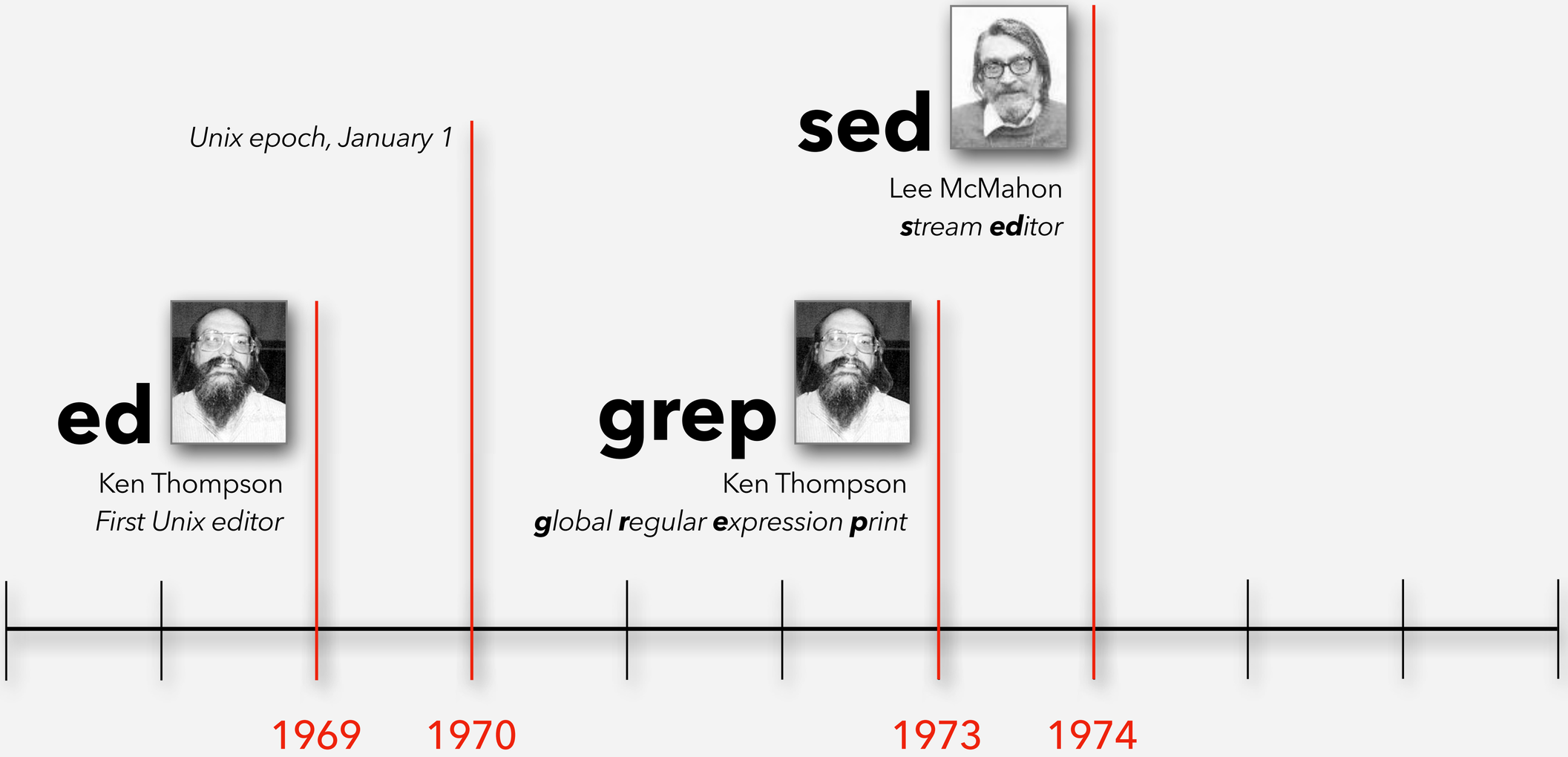
grep

Ken Thompson

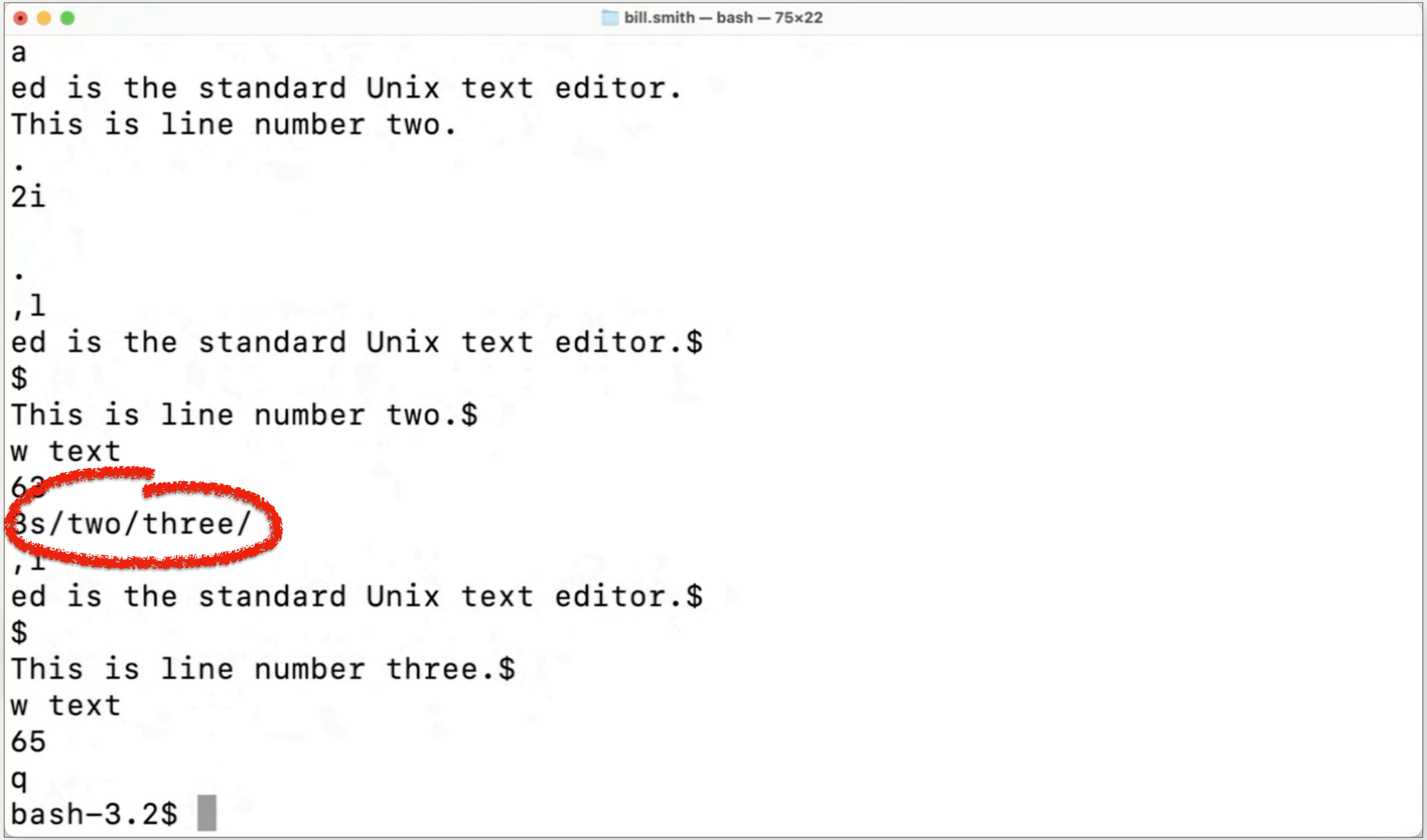
***g**lobal **r**egular **e**xpression **p**rint*

1969 1970 1973

Notes


Unix epoch, January 1
sed
Lee McMahon
stream editor
ed
Ken Thompson
First Unix editor
grep
Ken Thompson
global regular expression print
1969
1970
1973
1974

Notes


bill.smith — bash — 75×22
a
ed is the standard Unix text editor.
This is line number two.
.
2i
.
,l
ed is the standard Unix text editor.$
$
This is line number two.$
w text
63
3s/two/three/
,l
ed is the standard Unix text editor.$
$
This is line number three.$
w text
65
q
bash-3.2$

Notes

Unix epoch, January 1

ed

Ken Thompson

First Unix editor

sed

Lee McMahon

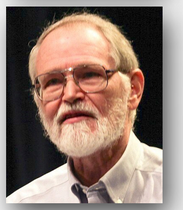
***s**tream **ed**itor*

grep

Ken Thompson

***g**lobal **r**egular **e**xpression **p**rint*

awk



***A**ho, **W**einberger, and **K**ernighan*

1969 1970 1973 1974 1977

Notes

Unix epoch, January 1

ed

Ken Thompson

First Unix editor

Search

grep

Ken Thompson

***g**lobal **r**egular **e**xpression **p**rint*

Search and replace

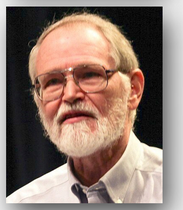
sed

Lee McMahon

***s**tream **ed**itor*

Text Processor

awk



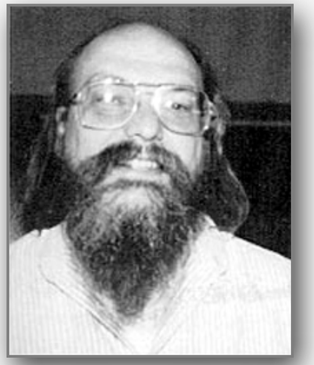
***A**ho, **W**einberger, and **K**ernighan*

1969 1970 1973 1974 1977

Notes

ed | awk
sed
grep

Unix philosophy



An approach to software development that emphasizes minimalism, modularism, and reusability. It emphasizes code that can be extended and maintained by someone other than its creators.

It is antithetical to monolithic design.

- Write programs that do one thing and do it well.
- Write programs to work together.
- Write programs to handle text streams, because that is a universal interface.

Notes

- Origins
- What they have in common
- When to use each
- Syntax

Notes

Similarities and differences in function

	ed	grep	sed	awk
plain text				
numbers and calculations				
file argument				
one-letter commands				
substitution				
line-based editing				
regular expressions				
addressing				
global by default				

Notes

Similarities and differences in function

	ed	grep	sed	awk
plain text	✔	✔	✔	✔
numbers and calculations				
file argument				
one-letter commands				
substitution				
line-based editing				
regular expressions				
addressing				
global by default				

Notes

Similarities and differences in function

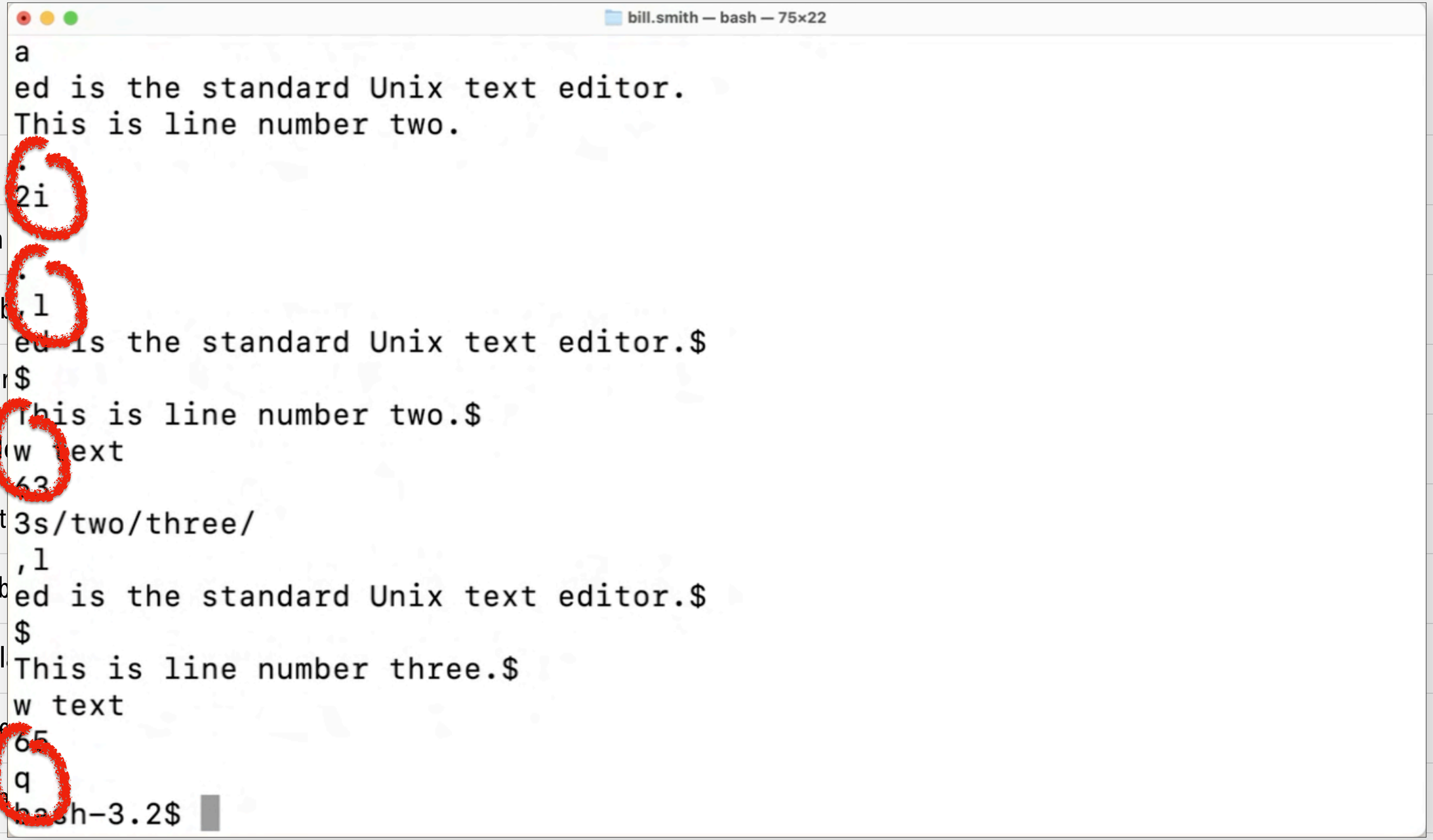
	ed	grep	sed	awk
plain text	✔	✔	✔	✔
numbers and calculations	✘	✘	✘	✔
file argument				
one-letter commands				
substitution				
line-based editing				
regular expressions				
addressing				
global by default				

Notes

Similarities and differences in function

	ed	grep	sed	awk
plain text	✔	✔	✔	✔
numbers and calculations	✘	✘	✘	✔
file argument	✔	✔	✔	✔
one-letter commands				
substitution				
line-based editing				
regular expressions				
addressing				
global by default				

Notes


bill.smith — bash — 75×22
a
ed is the standard Unix text editor.
This is line number two.
.
2i
.
,l
ed is the standard Unix text editor.$
$
This is line number two.$
w text
63
3s/two/three/
,l
ed is the standard Unix text editor.$
$
This is line number three.$
w text
65
q
bash-3.2$

Notes

Similarities and differences in function

	ed	grep	sed	awk
plain text	✔	✔	✔	✔
numbers and calculations	✘	✘	✘	✔
file argument	✔	✔	✔	✔
one-letter commands	✔	✘	✔	✘
substitution	✔	✘	✔	✔
line-based editing				
regular expressions				
addressing				
global by default				

Notes

Similarities and differences in function

	ed	grep	sed	awk
plain text	✔	✔	✔	✔
numbers and calculations	✘	✘	✘	✔
file argument	✔	✔	✔	✔
one-letter commands	✔	✘	✔	✘
substitution	✔	✘	✔	✔
line-based editing				
regular expressions				
addressing				
global by default				

Search

Search and replace

Notes

Similarities and differences in function

	ed	grep	sed	awk
plain text	✔	✔	✔	✔
numbers and calculations	✘	✘	✘	✔
file argument	✔	✔	✔	✔
one-letter commands	✔	✘	✔	✘
substitution	✔	✘	✔	✔
line-based editing	✔	✘	✔	✔
regular expressions				
addressing				
global by default				

Search

Search and replace

Notes

Similarities and differences in function

	ed	grep	sed	awk
plain text	✔	✔	✔	✔
numbers and calculations	✘	✘	✘	✔
file argument	✔	✔	✔	✔
one-letter commands	✔	✘	✔	✘
substitution	✔	✘	✔	✔
line-based editing	✔	✘	✔	✔
regular expressions	✔	✔	✔	✔
addressing				
global by default				

An Introduction to

(re.ex|re+gex|re?gex|re*gex){1}

https://www.youtube.com/watch?v=Wc8Kpw0nEww

YouTube

Notes

Similarities and differences in function

	ed	grep	sed	awk
plain text	✔	✔	✔	✔
numbers and calculations	✘	✘	✘	✔
file argument	✔	✔	✔	✔
one-letter commands	✔	✘	✔	✘
substitution	✔	✘	✔	✔
line-based editing	✔	✘	✔	✔
regular expressions	✔	✔	✔	✔
addressing	✔	✔	✔	✔
global by default				

Notes

```
function checkResponseCode() {
  httpStatusCodes="000 No HTTP code received
200 Request successful
201 Request to create or update object successful
400 Bad request
401 Authentication failed
403 Invalid permissions
404 Object/resource not found
409 Conflict
500 Internal server error"

  responseCode=${1: -3}
  code=$( grep "$responseCode" <<< "$httpStatusCodes" )

  echo "$code"
}
```

Notes

Similarities and differences in function

	ed	grep	sed	awk
plain text	✔	✔	✔	✔
numbers and calculations	✘	✘	✘	✔
file argument	✔	✔	✔	✔
one-letter commands	✔	✘	✔	✘
substitution	✔	✘	✔	✔
line-based editing	✔	✘	✔	✔
regular expressions	✔	✔	✔	✔
addressing	✔	✔	✔	✔
global by default	✘	✔	✔	✔

Notes

```
poem="Mary had a little lamb.
Its fleece was white as snow.
And everywhere that Mary went,
The lamb was sure to go."

grep 'Mary' <<< "$poem"

sed -n '/Mary/p' <<< "$poem"

awk '/Mary/ { print $0 }' <<< "$poem"
```

```
Mary had a little lamb.
And everywhere that Mary went,
```

Notes

```
poem="Mary had a little lamb.
Its fleece was white as snow.
And everywhere that Mary went,
The lamb was sure to go."

grep 'Mary' <<< "$poem"

sed -n '/Mary/p' <<< "$poem"

awk '/Mary/ { print $0 }' <<< "$poem"
```

```
Mary had a little lamb.
And everywhere that Mary went,
```

Notes

```
poem="Mary had a little lamb.
Its fleece was white as snow.
And everywhere that Mary went,
The lamb was sure to go."

grep 'Mary' <<< "$poem"

sed -n '/Mary/p' <<< "$poem"

awk '/Mary/ { print $0 }' <<< "$poem"
```

```
Mary had a little lamb.
And everywhere that Mary went,
```

Notes

```
poem="Mary had a little lamb.
Its fleece was white as snow.
And everywhere that Mary went,
The lamb was sure to go."

grep 'Mary' <<< "$poem"

sed -n '/Mary/p' <<< "$poem"

awk '/Mary/ { print $0 }' <<< "$poem"
```

```
Mary had a little lamb.
Mary had a little lamb.
Its fleece was white as snow.
And everywhere that Mary went,
And everywhere that Mary went,
The lamb was sure to go.
```

Notes

```
poem="Mary had a little lamb.
Its fleece was white as snow.
And everywhere that Mary went,
The lamb was sure to go."

grep 'Mary' <<< "$poem"

sed -n '/Mary/p' <<< "$poem"

awk '/Mary/ { print $0 }' <<< "$poem"
```

```
 $1   $2 $3  $4    $5
Mary had a little lamb.
And everywhere that Mary went,
```

Notes

Similarities and differences in function

	ed	grep	sed	awk
plain text	✔	✔	✔	✔
numbers and calculations	✘	✘	✘	✔
file argument	✔	✔	✔	✔
one-letter commands	✔	✘	✔	✘
substitution	✔	✘	✔	✔
line-based editing	✔	✘	✔	✔
regular expressions	✔	✔	✔	✔
addressing	✔	✔	✔	✔
global by default	✘	✔	✔	✔

Notes

- Origins
- What they have in common
- When to use each
- Syntax

Notes

Choosing the right tool

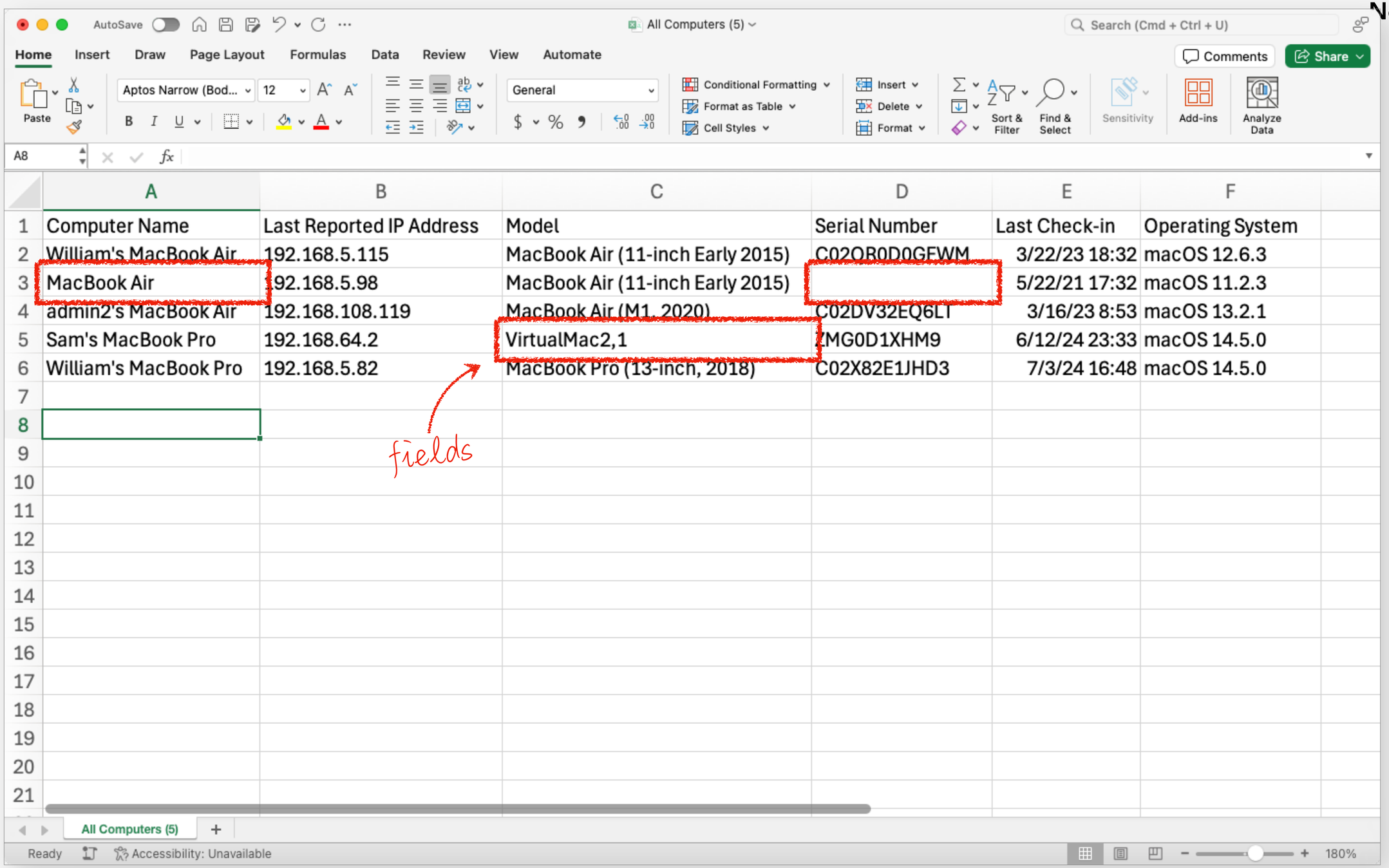
‘What data do we have and what do we want from it?’

Notes

Structured data

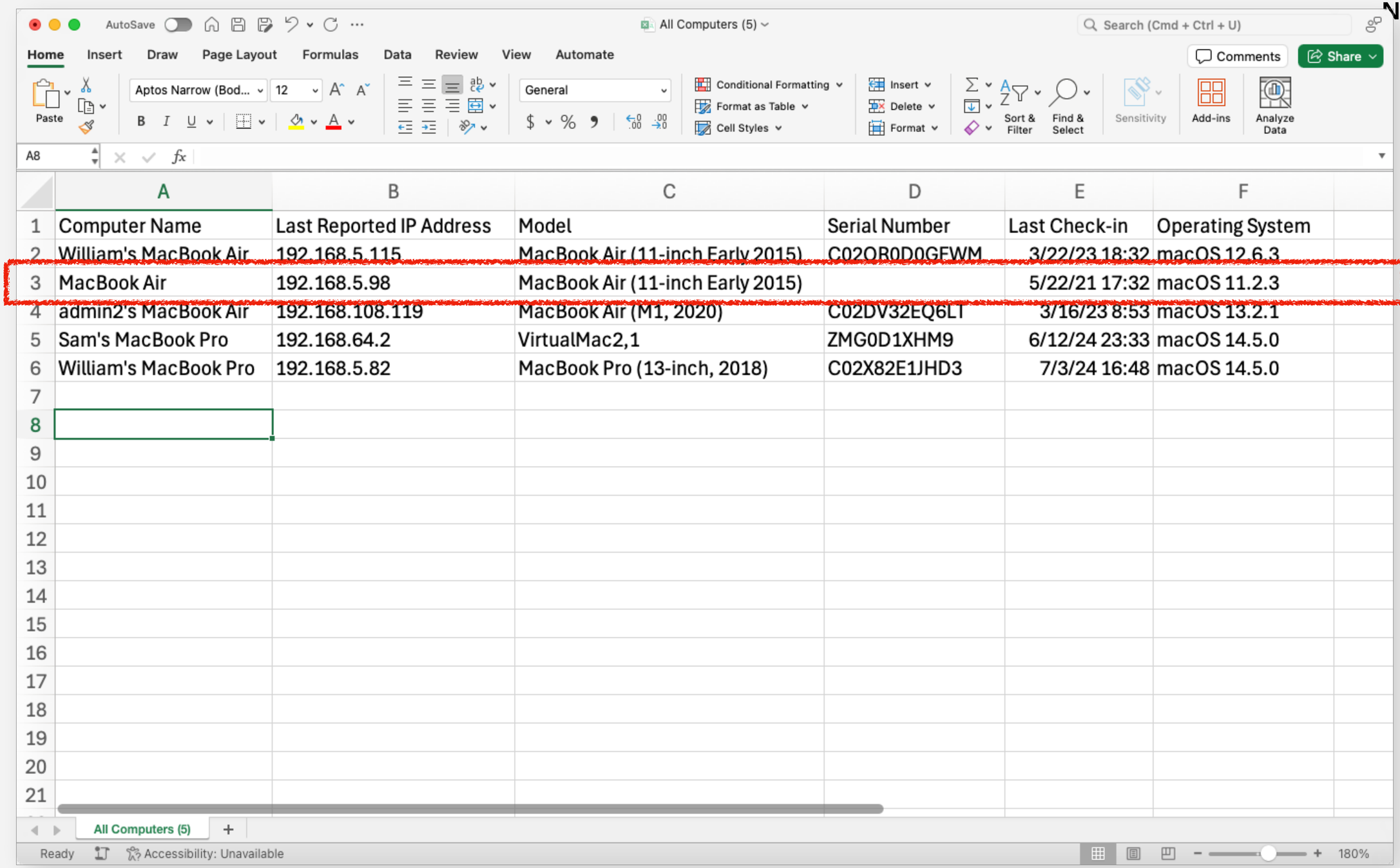
‘We really want data in a standardized format.’

Notes



	A	B	C	D	E	F
1	Computer Name	Last Reported IP Address	Model	Serial Number	Last Check-in	Operating System
2	William's MacBook Air	192.168.5.115	MacBook Air (11-inch Early 2015)	C02QR0D0GFWM	3/22/23 18:32	macOS 12.6.3
3	MacBook Air	192.168.5.98	MacBook Air (11-inch Early 2015)		5/22/21 17:32	macOS 11.2.3
4	admin2's MacBook Air	192.168.108.119	MacBook Air (M1, 2020)	C02DV32EQ6LT	3/16/23 8:53	macOS 13.2.1
5	Sam's MacBook Pro	192.168.64.2	VirtualMac2,1	ZMG0D1XHM9	6/12/24 23:33	macOS 14.5.0
6	William's MacBook Pro	192.168.5.82	MacBook Pro (13-inch, 2018)	C02X82E1JHD3	7/3/24 16:48	macOS 14.5.0
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						

Notes

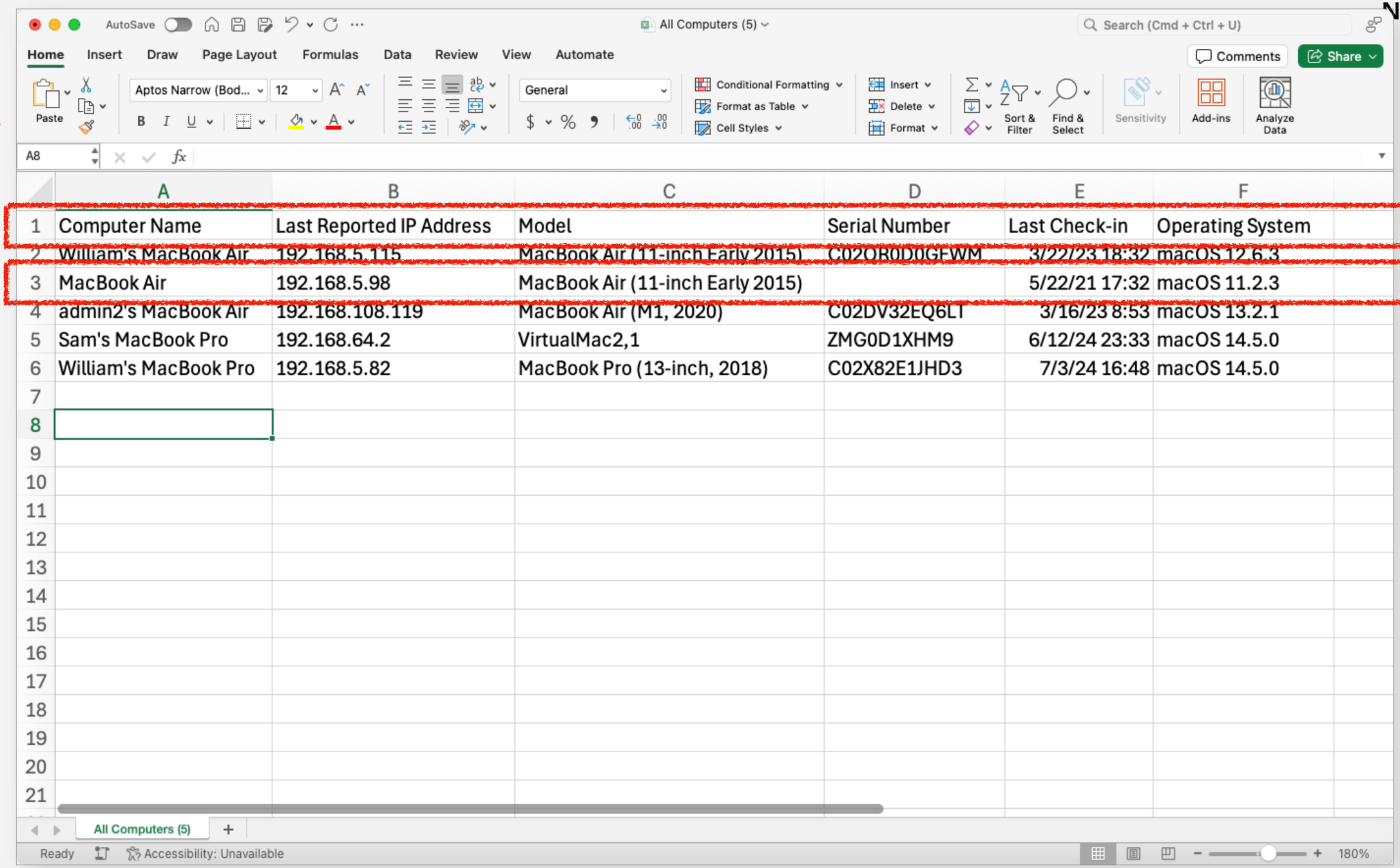


	A	B	C	D	E	F
1	Computer Name	Last Reported IP Address	Model	Serial Number	Last Check-in	Operating System
2	William's MacBook Air	192.168.5.115	MacBook Air (11-inch Early 2015)	C02QR0D0GFWM	3/22/23 18:32	macOS 12.6.3
3	MacBook Air	192.168.5.98	MacBook Air (11-inch Early 2015)		5/22/21 17:32	macOS 11.2.3
4	admin2's MacBook Air	192.168.108.119	MacBook Air (M1, 2020)	C02DV32EQ6LT	3/16/23 8:53	macOS 13.2.1
5	Sam's MacBook Pro	192.168.64.2	VirtualMac2,1	ZMG0D1XHM9	6/12/24 23:33	macOS 14.5.0
6	William's MacBook Pro	192.168.5.82	MacBook Pro (13-inch, 2018)	C02X82E1JHD3	7/3/24 16:48	macOS 14.5.0
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						

Notes

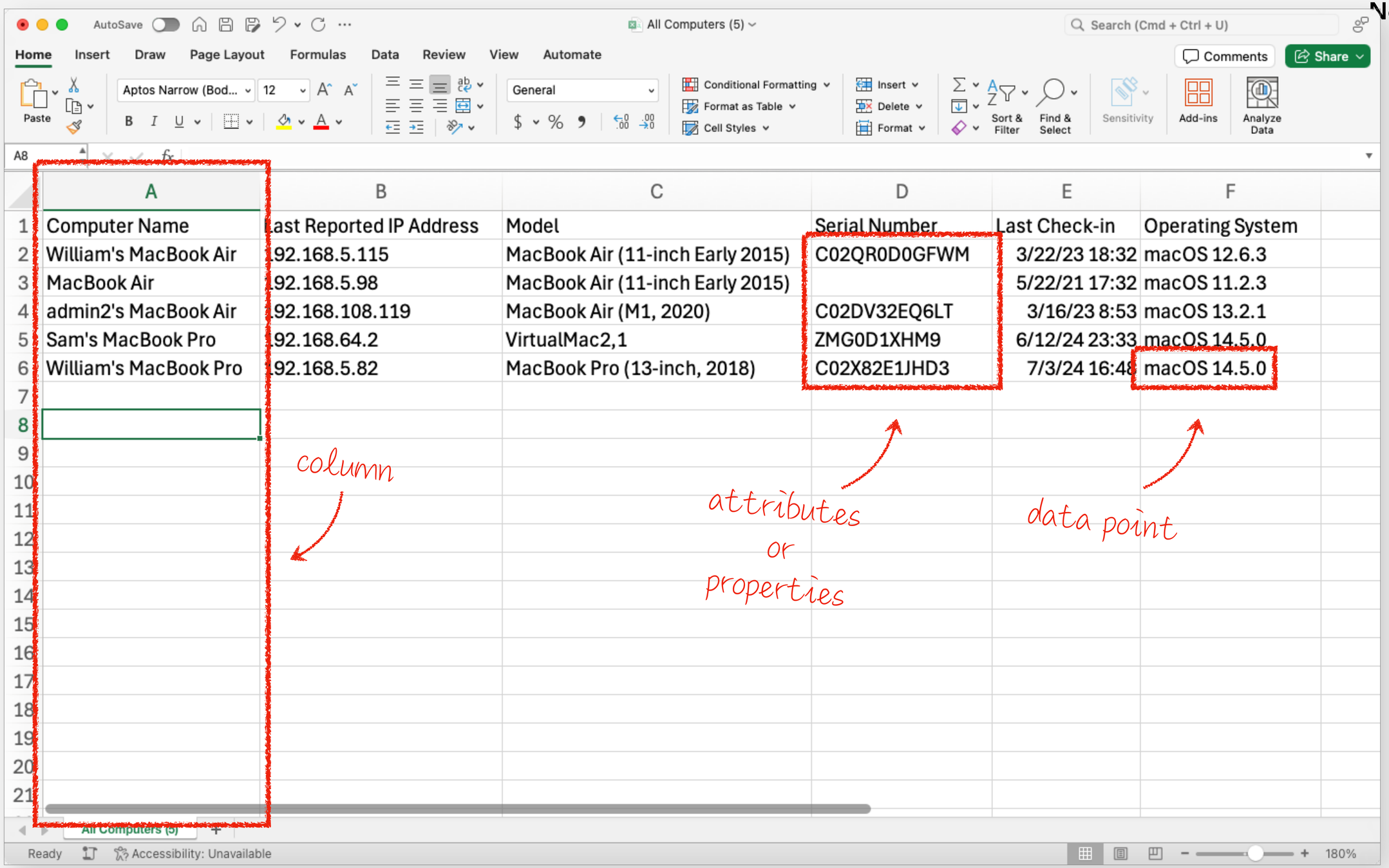
header

record

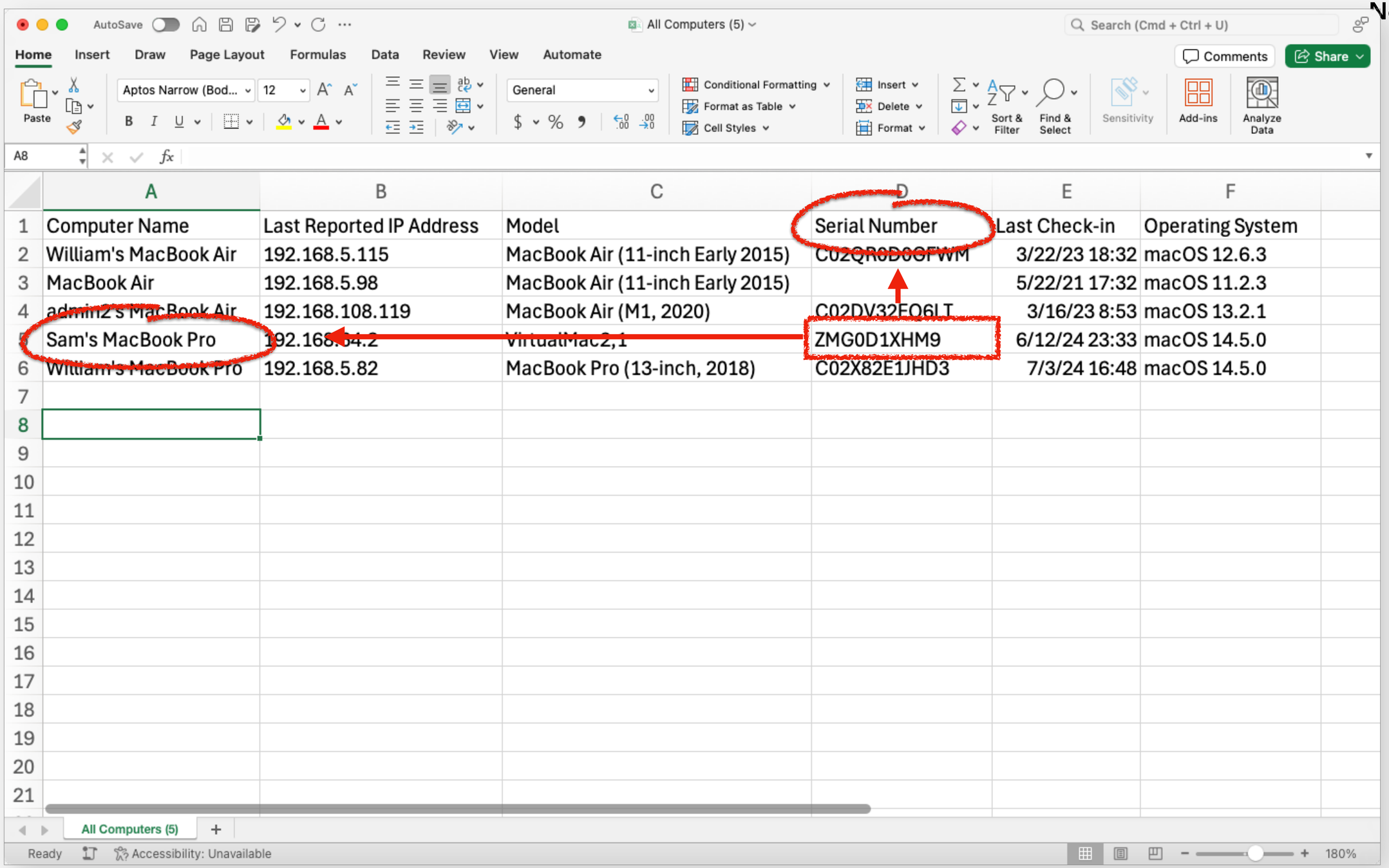


	A	B	C	D	E	F
1	Computer Name	Last Reported IP Address	Model	Serial Number	Last Check-in	Operating System
2	William's MacBook Air	192.168.5.115	MacBook Air (11-inch Early 2015)	C02QR0D0GFWM	3/22/23 18:32	macOS 12.6.3
3	MacBook Air	192.168.5.98	MacBook Air (11-inch Early 2015)		5/22/21 17:32	macOS 11.2.3
4	admin2's MacBook Air	192.168.108.119	MacBook Air (M1, 2020)	C02DV32EQ6LT	3/16/23 8:53	macOS 13.2.1
5	Sam's MacBook Pro	192.168.64.2	VirtualMac2,1	ZMG0D1XHM9	6/12/24 23:33	macOS 14.5.0
6	William's MacBook Pro	192.168.5.82	MacBook Pro (13-inch, 2018)	C02X82E1JHD3	7/3/24 16:48	macOS 14.5.0
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						

Notes


AutoSave
All Computers (5)
Search (Cmd + Ctrl + U)
Home
Insert
Draw
Page Layout
Formulas
Data
Review
View
Automate
Comments
Share
Paste
Aptos Narrow (Bod...
12
General
Conditional Formatting
Format as Table
Cell Styles
Insert
Delete
Format
Sort & Filter
Find & Select
Sensitivity
Add-ins
Analyze Data
A8
Computer Name
Last Reported IP Address
Model
Serial Number
Last Check-in
Operating System
William's MacBook Air
192.168.5.115
MacBook Air (11-inch Early 2015)
C02QR0D0GFWM
3/22/23 18:32
macOS 12.6.3
MacBook Air
192.168.5.98
MacBook Air (11-inch Early 2015)
5/22/21 17:32
macOS 11.2.3
admin2's MacBook Air
192.168.108.119
MacBook Air (M1, 2020)
C02DV32EQ6LT
3/16/23 8:53
macOS 13.2.1
Sam's MacBook Pro
192.168.64.2
VirtualMac2,1
ZMG0D1XHM9
6/12/24 23:33
macOS 14.5.0
William's MacBook Pro
192.168.5.82
MacBook Pro (13-inch, 2018)
C02X82E1JHD3
7/3/24 16:48
macOS 14.5.0
column
attributes or properties
data point
All Computers (5)
Ready
Accessibility: Unavailable
180%

Notes


All Computers (5)
Search (Cmd + Ctrl + U)
Home Insert Draw Page Layout Formulas Data Review View Automate
Comments
Share
Paste
Aptos Narrow (Bod...
12
General
Conditional Formatting
Format as Table
Cell Styles
Insert
Delete
Format
Sort & Filter
Find & Select
Sensitivity
Add-ins
Analyze Data
A8
Computer Name
Last Reported IP Address
Model
Serial Number
Last Check-in
Operating System
William's MacBook Air
192.168.5.115
MacBook Air (11-inch Early 2015)
3/22/23 18:32
macOS 12.6.3
MacBook Air
192.168.5.98
MacBook Air (11-inch Early 2015)
5/22/21 17:32
macOS 11.2.3
MacBook Air (M1, 2020)
C02DV32FQ6LT
3/16/23 8:53
macOS 13.2.1
Sam's MacBook Pro
ZMG0D1XHM9
6/12/24 23:33
macOS 14.5.0
192.168.5.82
MacBook Pro (13-inch, 2018)
C02X82E1JHD3
7/3/24 16:48
macOS 14.5.0
All Computers (5)
Ready
Accessibility: Unavailable
180%

Notes

Table structure and text structures

Computer Name	Last Reported IP Address	Model	Serial Number	Last Check-in	Operating System
William's MacBook Air	192.168.5.115	MacBook Air (11-inch Early 2015)	C02QR0D0GFWM	3/22/23 18:32	macOS 12.6.3
MacBook Air	192.168.5.98	MacBook Air (11-inch Early 2015)		5/22/21 17:32	macOS 11.2.3
admin2's MacBook Air	192.168.108.119	MacBook Air (M1, 2020)	C02DV32EQ6LT	3/16/23 8:53	macOS 13.2.1
Sam's MacBook Pro	192.168.64.2	VirtualMac2,1	ZMG0D1XHM9	6/12/24 23:33	macOS 14.5.0
William's MacBook Pro	192.168.5.82	MacBook Pro (13-inch, 2018)	C02X82E1JHD3	7/3/24 16:48	macOS 14.5.0

Notes

Comma-separated values (CSV) structure

Computer Name,Last Reported IP Address,Model,Serial Number,Last Check-in,Operating System

William’s MacBook Air,192.168.5.115,MacBook Air (11-inch Early 2015),C02QR0D0GFWM,2023-03-22 18:32:48,macOS 12.6.3

MacBook Air,192.168.5.98,MacBook Air (11-inch Early 2015),,2021-0

admin2’s MacBook Air,192.168.108.119,"MacBook Air (M1, 2020)",C02DV32EQ6LT,2023-03-16 08:53:44,macOS 13.2.1

Sam's MacBook Pro,192.168.64.2,"VirtualMac2,1",ZMG0D1XHM9,2024-06-12 23:33:45,macOS 14.5.0

William’s MacBook Pro,192.168.5.82,"MacBook Pro (13-inch, 2018)",C02X82E1JHD3,2024-07-03 16:48:23,macOS 14.5.0

Notes

Extensible markup language (XML) structure

```
<Computers>
 <Computer>
  <Computer_Name>William's MacBook Air</Computer_Name>
  <Last_Reported_IP_Address>192.168.5.115</Last_Reported_IP_Address>
  <Model>MacBook Air (11-inch Early 2015)</Model>
  <Serial_Number>C02QR0D0GFWM</Serial_Number>
  <Last_Check_in>2023-03-22 18:32:48</Last_Check_in>
  <Operating_System>macOS 12.6.3</Operating_System>
 </Computer>
 <Computer>
  <Computer_Name>MacBook Air</Computer_Name>
  <Last_Reported_IP_Address>192.168.5.98</Last_Reported_IP_Address>
  <Model>MacBook Air (11-inch Early 2015)</Model>
  <Serial_Number/>
  <Last_Check_in>2021-05-22 17:32:52</Last_Check_in>
  <Operating_System>macOS 11.2.3</Operating_System>
 </Computer>
 <Computer>
  <Computer_Name>admin2's MacBook Air</Computer_Name>
  <Last_Reported_IP_Address>192.168.108.119</Last_Reported_IP_Address>
  <Model>MacBook Air (M1, 2020)</Model>
  <Serial_Number>C02DV32EQ6LT</Serial_Number>
  <Last_Check_in>2023-03-16 08:53:44</Last_Check_in>
  <Operating_System>macOS 13.2.1</Operating_System>
 </Computer>
 <Computer>
  <Computer_Name>Sam's MacBook Pro</Computer_Name>
  <Last_Reported_IP_Address>192.168.64.2</Last_Reported_IP_Address>
  <Model>VirtualMac2,1</Model>
  <Serial_Number>ZMG0D1XHM9</Serial_Number>
  <Last_Check_in>2024-06-12 23:33:45</Last_Check_in>
  <Operating_System>macOS 14.5.0</Operating_System>
 </Computer>
 <Computer>
   <Computer_Name>William's MacBook Pro</Computer_Name>
  <Last_Reported_IP_Address>192.168.5.82</Last_Reported_IP_Address>
  <Model>MacBook Pro (13-inch, 2018)</Model>
  <Serial_Number>C02X82E1JHD3</Serial_Number>
  <Last_Check_in>2024-07-03 16:48:23</Last_Check_in>
  <Operating_System>macOS 14.5.0</Operating_System>
 </Computer>
</Computers>
```

Notes

Extensible markup language (XML) structure

```
<Computers>
 <Computer>
  <Computer_Name>William's MacBook Air</Computer_Name>
  <Last_Reported_IP_Address>192.168.5.115</Last_Reported_IP_Address>
  <Model>MacBook Air (11-inch Early 2015)</Model>
  <Serial_Number>C02QR0D0GFWM</Serial_Number>
  <Last_Check_in>2023-03-22 18:32:48</Last_Check_in>
  <Operating_System>macOS 12.6.3</Operating_System>
 </Computer>
 <Computer>
  <Computer_Name>MacBook Air</Computer_Name>
  <Last_Reported_IP_Address>192.168.5.98</Last_Reported_IP_Address>
  <Model>MacBook Air (11-inch Early 2015)</Model>
  <Serial_Number/>
  <Last_Check_in>2021-05-22 17:32:52</Last_Check_in>
  <Operating_System>macOS 11.2.3</Operating_System>
 </Computer>
 <Computer>
  <Computer_Name>admin2's MacBook Air</Computer_Name>
  <Last_Reported_IP_Address>192.168.108.119</Last_Reported_IP_Address>
  <Model>MacBook Air (M1, 2020)</Model>
```

Notes

Extensible markup language (XML) structure

<Computers><Computer><Computer_Name>William’s MacBook Air</Computer_Name><Last_Reported_IP_Address>192.168.5.115</Last_Reported_IP_Address><Model>MacBook Air (11-inch Early 2015)</Model><Serial_Number>C02QR0D0GFWM</Serial_Number><Last_Check_in>2023-03-22 18:32:48</Last_Check_in><Operating_System>macOS 12.6.3</Operating_System></Computer><Computer><Computer_Name>MacBook Air</Computer_Name><Last_Reported_IP_Address>192.168.5.98</Last_Reported_IP_Address><Model>MacBook Air (11-inch Early 2015)</Model><Serial_Number/><Last_Check_in>2021-05-22 17:32:52</Last_Check_in><Operating_System>macOS 11.2.3</Operating_System></Computer><Computer><Computer_Name>admin2’s MacBook Air</Computer_Name><Last_Reported_IP_Address>192.168.108.119</Last_Reported_IP_Address><Model>MacBook Air (M1, 2020)</Model><Serial_Number>C02DV32EQ6LT</Serial_Number><Last_Check_in>2023-03-16 08:53:44</Last_Check_in><Operating_System>macOS 13.2.1</Operating_System></Computer><Computer><Computer_Name>Sam's MacBook Pro</Computer_Name><Last_Reported_IP_Address>192.168.64.2</Last_Reported_IP_Address><Model>VirtualMac2,1</Model><Serial_Number>ZMG0D1XHM9</Serial_Number><Last_Check_in>2024-06-12 23:33:45</Last_Check_in><Operating_System>macOS 14.5.0</Operating_System></Computer><Computer><Computer_Name>William’s MacBook Pro</Computer_Name><Last_Reported_IP_Address>192.168.5.82</Last_Reported_IP_Address><Model>MacBook Pro (13-inch, 2018)</Model><Serial_Number>C02X82E1JHD3</Serial_Number><Last_Check_in>2024-07-03 16:48:23</Last_Check_in><Operating_System>macOS 14.5.0</Operating_System></Computer></Computers>

Notes

Extensible markup language (XML) structure

<Computers><Computer><Computer_Name>William’s Mac

Notes

Other data structures

- ★ Table
- ★ CSV/Tab
- ★ XML
- ★ Time stamp
- ★ Date
- ★ Time
- ★ JSON
- ★ Log file
- ★ HTML
- ★ Markdown
- ★ Property list
- ★ Camel case

Notes

Choose the right tool

‘What data do we have and what do we want from it?’

Notes

Choose the right tool

	grep	sed	awk
	Search	Search and replace	Process text
I don't see a pattern.			
I'm only looking for the existence of something.			
I'm trying to change something.			
I'm trying to extract specific data points.			
I'm trying to reformat my data.			
My data has no line breaks.			

Notes

Choose the right tool

	grep	sed	awk
	Search	Search and replace	Process text
I don't see a pattern.	✔	✔	✘
I'm only looking for the existence of something.			
I'm trying to change something.			
I'm trying to extract specific data points.			
I'm trying to reformat my data.			
My data has no line breaks.			

Notes

Choose the right tool

	grep	sed	awk
	Search	Search and replace	Process text
I don't see a pattern.	✔	✔	✘
I'm only looking for the existence of something.	✔	✘	✘
I'm trying to change something.			
I'm trying to extract specific data points.			
I'm trying to reformat my data.			
My data has no line breaks.			

Notes

Choose the right tool

	grep	sed	awk
	Search	Search and replace	Process text
I don't see a pattern.	✔	✔	✘
I'm only looking for the existence of something.	✔	✘	✘
I'm trying to change something.	✘	✔	✘
I'm trying to extract specific data points.			
I'm trying to reformat my data.			
My data has no line breaks.			

Notes

Choose the right tool

	grep	sed	awk
	Search	Search and replace	Process text
I don't see a pattern.	✔	✔	✘
I'm only looking for the existence of something.	✔	✘	✘
I'm trying to change something.	✘	✔	✘
I'm trying to extract specific data points.	✘	✔	✔
I'm trying to reformat my data.			
My data has no line breaks.			

Notes

Choose the right tool

	grep	sed	awk
	Search	Search and replace	Process text
I don't see a pattern.	✔	✔	✘
I'm only looking for the existence of something.	✔	✘	✘
I'm trying to change something.	✘	✔	✘
I'm trying to extract specific data points.	✘	✔	✔
I'm trying to reformat my data.	✘	✘	✔
My data has no line breaks.			

Notes

Choose the right tool

	grep	sed	awk
	Search	Search and replace	Process text
I don't see a pattern.	✔	✔	✘
I'm only looking for the existence of something.	✔	✘	✘
I'm trying to change something.	✘	✔	✘
I'm trying to extract specific data points.	✘	✔	✔
I'm trying to reformat my data.	✘	✘	✔
My data has no line breaks.	✘	✔	✘

Notes

Structured data

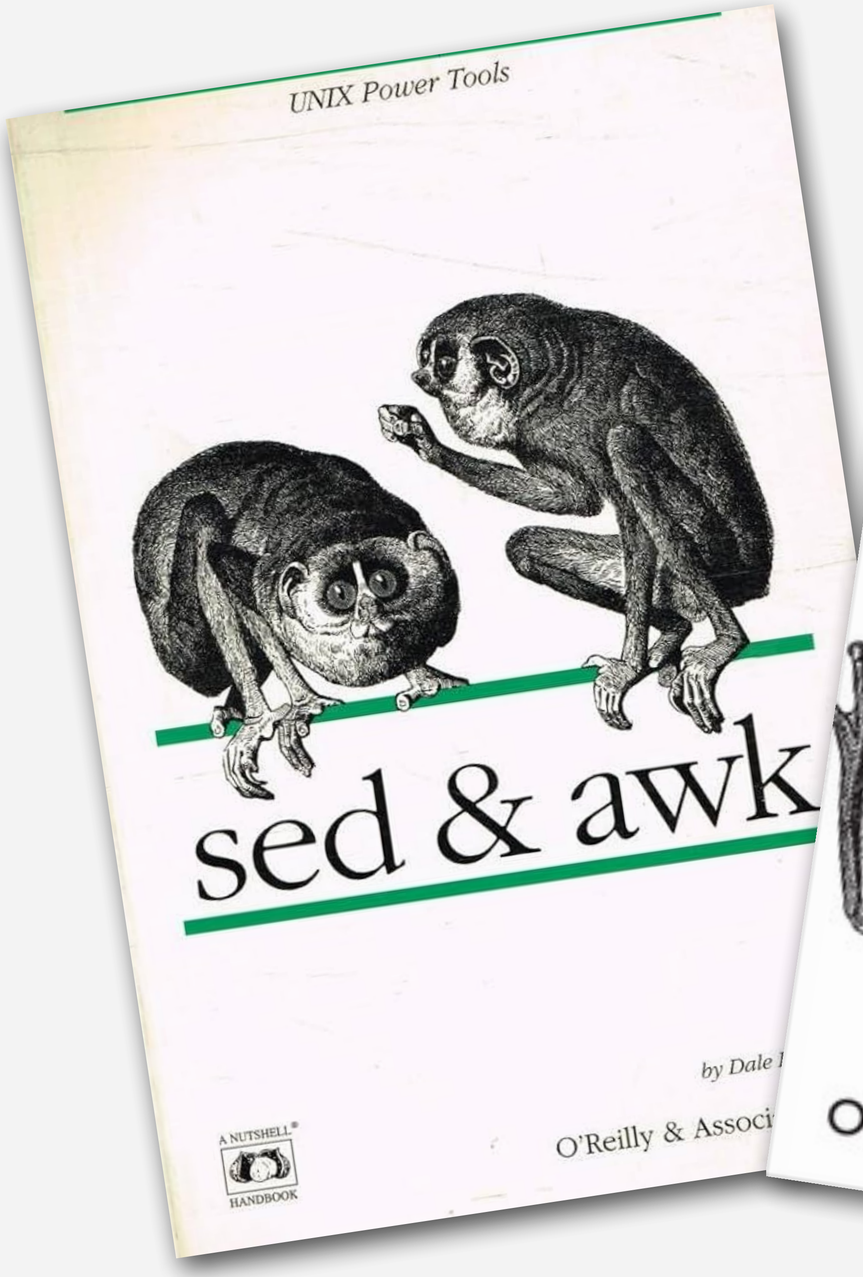
‘We really want data in a standardized format.’

Notes

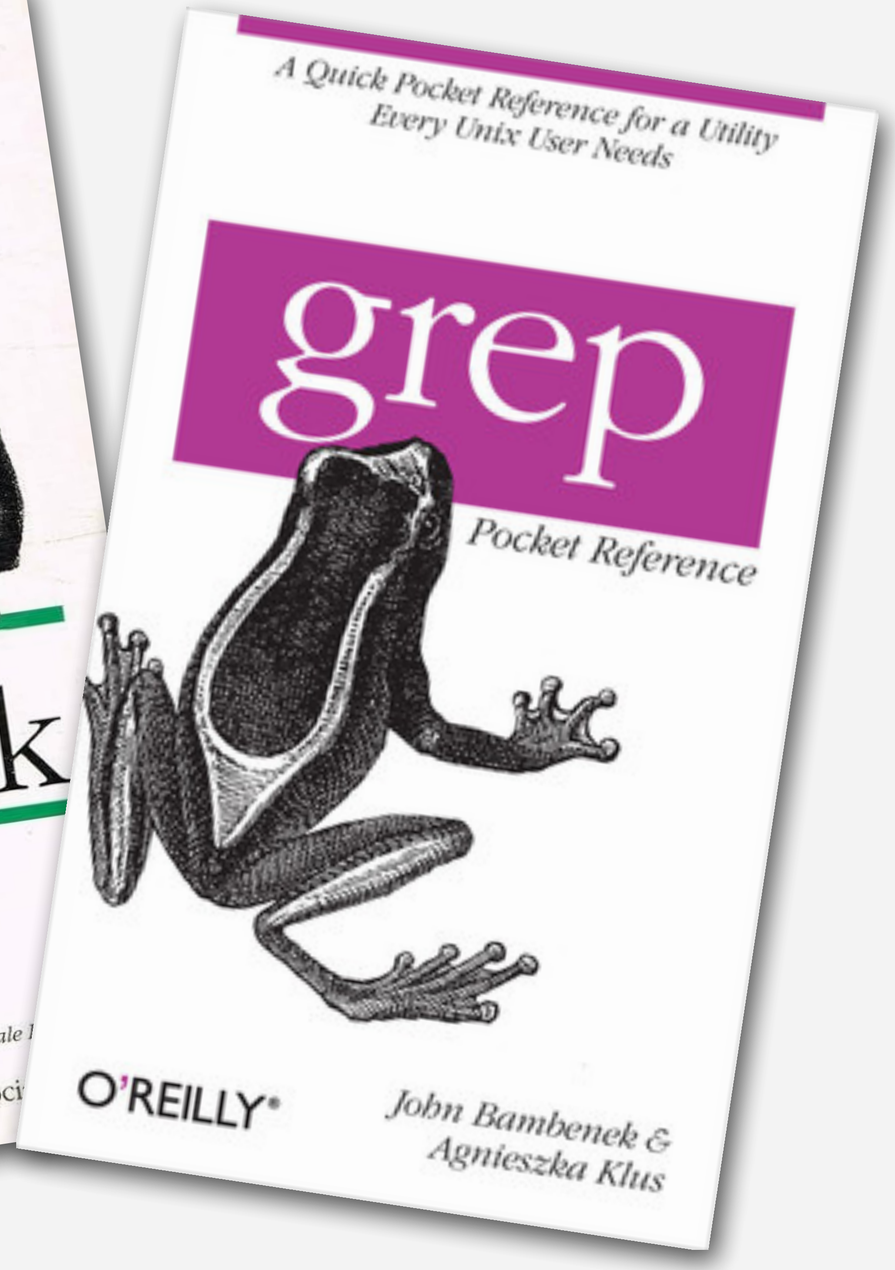
- Origins
- What they have in common
- When to use each
- Syntax

Notes

1990



2009



Notes

The useless use of cat

```
cat ~/Desktop/list.txt | grep "tacos"
```

```
cat ~/Desktop/list.txt | sed -n "tacos/p"
```

```
cat ~/Desktop/list.txt | awk '/tacos/ { print $0 }'
```

Notes

The useless use of cat

~~`cat ~/Desktop/list.txt | grep "tacos"`~~

✔ `grep "tacos" ~/Desktop/list.txt`

~~`cat ~/Desktop/list.txt | sed -n "tacos/p"`~~

✔ `sed -n "tacos/p" ~/Desktop/list.txt`

~~`cat ~/Desktop/list.txt | awk '/tacos/ { print $0 }'`~~

✔ `awk '/tacos/ { print $0 }' ~/Desktop/list.txt`

Notes

The useless use of echo

```
echo "$variable" | grep "tacos"
```

```
echo "$variable" | sed -n "tacos/p"
```

```
echo "$variable" | awk '/tacos/ { print $0 }'
```

Notes

The useless use of echo

~~`echo "$variable" | grep "tacos"`~~

```
grep "tacos" <<< "$variable"
```

~~`echo "$variable" | sed -n "tacos/p"`~~

```
sed -n "tacos/p" <<< "$variable"
```

~~`echo "$variable" | awk '/tacos/ { print $0 }'`~~

```
awk '/tacos/ { print $0 }' <<< "$variable"
```

Notes

Single quotes, double quotes, and no quotes

```
grep tacos <<< "$variable"
grep 'too many tacos' <<< "$variable"
grep "$variable" ~/Desktop/list.txt
```

Notes

Single quotes, double quotes, and no quotes

```
grep tacos <<< "$variable"
grep 'too many tacos' <<< "$variable"
grep "$variable" ~/Desktop/list.txt
```

```
sed -n 'tacos/p' <<< "$variable"
sed -n "$variable/p" ~/Desktop/list.txt
```

Notes

Single quotes, double quotes, and no quotes

```
grep tacos <<< "$variable"
grep 'too many tacos' <<< "$variable"
grep "$variable" ~/Desktop/list.txt


sed -n 'tacos/p' <<< "$variable"
sed -n "$variable/p" ~/Desktop/list.txt


awk '/tacos/ { print $0 }' <<< "$variable"
```

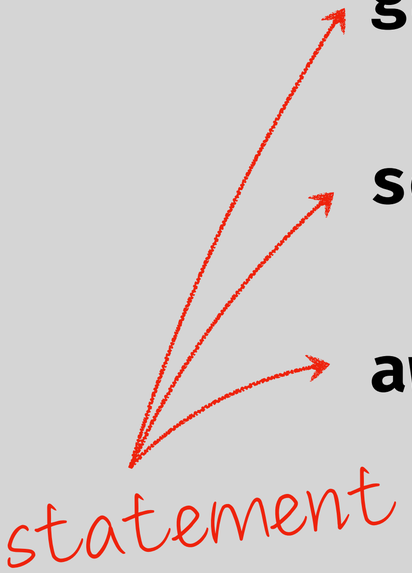
Notes

Terms

```
grep 'Mary' file.txt

sed -n '/Mary/p' <<< "$poem"

awk '/Mary/ { print $0 }' <<< "$poem"
```



Notes

Terms

```
grep 'Mary' file.txt
sed -n '/Mary/p' <<< "$poem"
awk '/Mary/ { print $0 }' <<< "$poem"
```

binary

program

application

command line tool

Notes

Terms

```
grep 'Mary' file.txt

sed -n '/Mary/p' <<< "$poem"

awk '/Mary/ { print $0 }' <<< "$poem"
```

option

Notes

Terms


```
grep -E 'Mary' file.txt

grep --extended-regexp 'Mary' file.txt
```

Notes

Terms

pattern

```
grep 'Mary' file.txt

sed -n '/Mary/p' <<< "$poem"

awk '/Mary/ { print $0 }' <<< "$poem"
```

Notes

Terms

```
grep 'M.*y' file.txt

sed -n '/M.*y/p' <<< "$poem"

awk '/M.*y/ { print $0 }' <<< "$poem"
```

`'M.*y'` = *'Mary', "Marty", "Misty" or "Magnanimously"*

Notes

Terms

```
grep 'Mary' file.txt

sed -n '/Mary/p' <<< "$poem"

awk '/Mary/ { print $0 }' <<< "$poem"
```

Notes

Terms

gr

se

awk

p = Print lines

d = Delete lines

w = Write pattern space to file

a = Append line after

i = Insert line before

command

Notes

Terms

```
gr
se
awk
```

```
'{ print $0 }'
'{ print $1, $2 }'
'{ print 10 + 20 }'
'{ a = 10; b = 20 } { print a + b }'
```

command

Notes

Terms

```
grep 'Mary' file.txt

sed -n '/Mary/p' <<< "$poem"

awk '/Mary/ { print $0 }' <<< "$poem"
```

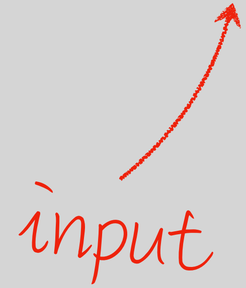
Notes

Terms

```
grep 'Mary' file.txt

sed -n '/Mary/p' <<< "$poem"

awk '/Mary/ { print $0 }' <<< "$poem"
```



Notes

Terms

```
program options address/pattern command input
```

Notes

Grep examples

```
xml="<mobile_device_model>
    <model_name>Watch7,3</model_name>
    <display_name>Apple Watch Series 8</display_name>
</mobile_device_model>
<mobile_device_model>
    <model_name>Watch7,4</model_name>
    <display_name>Apple Watch Series 9</display_name>
</mobile_device_model>"

grep "model_name" <<< "$xml"

  <model_name>Watch7,3</model_name>
  <model_name>Watch7,4</model_name>
```

Notes

Grep examples

```
xml="<mobile_device_model>
    <model_name>Watch7,3</model_name>
    <display_name>Apple Watch Series 8</display_name>
</mobile_device_model>
<mobile_device_model>
    <model_name>Watch7,4</model_name>
    <display_name>Apple Watch Series 9</display_name>
</mobile_device_model>"

grep --after-context 1 "model_name" <<< "$xml"

  <model_name>Watch7,3</model_name>
  <display_name>Apple Watch Series 8</display_name>
--
  <model_name>Watch7,4</model_name>
  <display_name>Apple Watch Series 9</display_name>
```

Notes

Grep examples

```
xml="<mobile_device_model>
    <model_name>Watch7,3</model_name>
    <display_name>Apple Watch Series 8</display_name>
</mobile_device_model>
<mobile_device_model>
    <model_name>Watch7,4</model_name>
    <display_name>Apple Watch Series 9</display_name>
</mobile_device_model>"

grep --after-context 1 --line-number "model_name" <<< "$xml"

2: <model_name>Watch7,3</model_name>
3- <display_name>Apple Watch Series 8</display_name>
--
6: <model_name>Watch7,4</model_name>
7- <display_name>Apple Watch Series 9</display_name>
```

Notes

Sed examples

```
modelNames="Watch7,3
Watch7,4
Watch7,5"

sed 's/Watch7,3/Apple Watch Series 9/' <<< "$modelNames"
```

```
Apple Watch Series 9
Watch7,4
Watch7,5
```

's/pattern/replacement/'

Notes

Sed examples

```
modelNames="Watch7,3
Watch7,4
Watch7,5"

sed 's/Watch7,3/Apple Watch Series 9/ ; s/Watch7,4/Apple Watch Series 9/ ;
s/Watch7,5/Apple Watch Series 9/' <<< "$modelNames"

Apple Watch Series 9
Apple Watch Series 9
Apple Watch Series 9
```

's/pattern/replacement/'

Notes

Sed examples

```
modelNames="Watch7,3
Watch7,4
Watch7,5"

sed 's/Watch7,\d/Apple Watch Series 9/' <<< "$modelNames"
```

```
Apple Watch Series 9
Apple Watch Series 9
Apple Watch Series 9
```

's/pattern/replacement/'

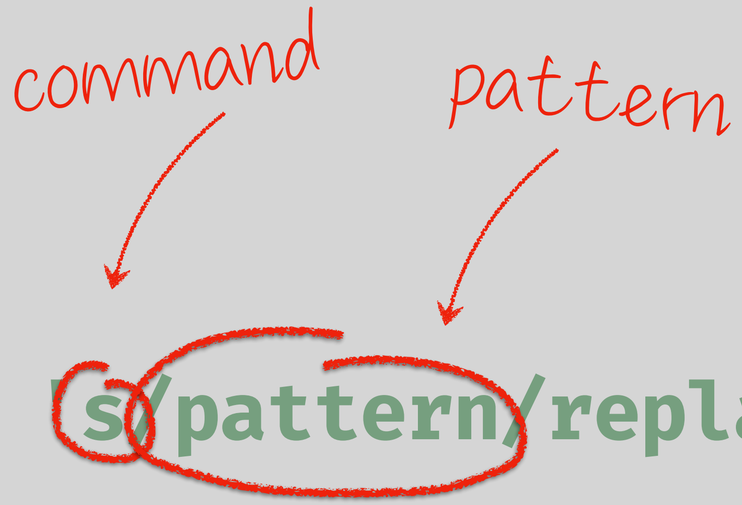
Notes

Sed examples

```
's/pattern/replacement/'
```

Notes

Sed examples




```
program options address/pattern command input
```

Notes

Sed examples

```
'/pattern/one-letter-command'
```

Notes

Sed examples

```
list="Line 1
Line 2
Line 3
Line 4
Line 5"

sed '2,4 d' <<< "$list"

Line 1
Line 5
```

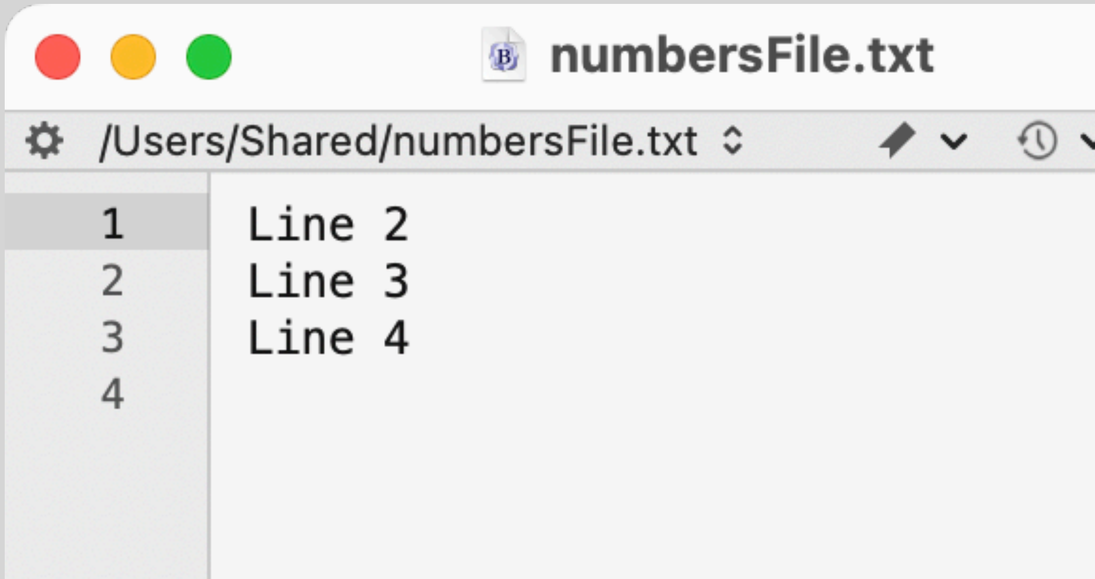
Notes

Sed examples

```
list="Line 1
Line 2
Line 3
Line 4
Line 5"

sed '2,4 w /Users/Shared/numbersFile.txt' <<< "$list"
```

```
Line 1
Line 2
Line 3
Line 4
Line 5
```



Notes

Awk examples

```
mailingList="Abigail Adams, 100 A Street, Albany, CA 94706
Bob Bright, 200 B Street, Bakersfield, CA 93301
Charlie Cartwright, 300 C Street, Cambridge, NY 12816
Denise Darling, 400 D Street, Dale, NY 14039
Edith Ebbing, 500 E Street, Eagleville, CA 96110"
```

Notes

Awk examples

```
mailingList="Abigail Adams, 100 A Street, Albany, CA 94706
Bob Bright, 200 B Street, Bakersfield, CA 93301
Charlie Cartwright, 300 C Street, Cambridge, NY 12816
Denise Darling, 400 D Street, Dale, NY 14039
Edith Ebbing, 500 E Street, Eagleville, CA 96110"

awk -F "," '/CA/ {
```

Notes

Awk examples

```
mailingList="Abigail Adams, 100 A Street, Albany, CA 94706
Bob Bright, 200 B Street, Bakersfield, CA 93301
Charlie Cartwright, 300 C Street, Cambridge, NY 12816
Denise Darling, 400 D Street, Dale, NY 14039
Edith Ebbing, 500 E Street, Eagleville, CA 96110"

awk -F "," '/CA/ {
print $1
print $2
print $3 $4 $5
}' <<< "$mailingList"
```

Notes

```
mailingList="A
Bob Bright, 20
Charlie Cartwr
Denise Darling
Edith Ebbing,

awk -F "," '/C
print $1
print $2
print $3 $4 $5
}' <<< "$maili
```

Abigail Adams
100 A Street
Albany CA 94706

Bob Bright
200 B Street
Bakersfield CA 93301

Edith Ebbing
500 E Street
Eagleville CA 96110

Notes

Awk examples

```
mailingList="Abigail Adams, 100 A Street, Albany, CA 94706
Bob Bright, 200 B Street, Bakersfield, CA 93301
Charlie Cartwright, 300 C Street, Cambridge, NY 12816
Denise Darling, 400 D Street, Dale, NY 14039
Edith Ebbing, 500 E Street, Eagleville, CA 96110"

fixedMailingList=$( sed 's/, /\t/g' <<< "$mailingList" )

awk -F "," '/CA/ {
print $1
print $2
print $3 $4 $5
}' <<< "$fixedMailingList"
```

Notes

```
mailingList="A
Bob Bright, 20
Charlie Cartwr
Denise Darling
Edith Ebbing,

fixedMailingLi

awk -F "," '/C
print $1
print $2
print $3 $4 $5
}' <<< "$fixed
```

Abigail Adams 100 A Street Albany CA 94706

Bob Bright 200 B Street Bakersfield CA 93301

Edith Ebbing 500 E Street Eagleville CA 96110

Notes

Awk examples

```
mailingList="Abigail Adams, 100 A Street, Albany, CA 94706
Bob Bright, 200 B Street, Bakersfield, CA 93301
Charlie Cartwright, 300 C Street, Cambridge, NY 12816
Denise Darling, 400 D Street, Dale, NY 14039
Edith Ebbing, 500 E Street, Eagleville, CA 96110"

fixedMailingList=$( sed 's/, /\t/g' <<< "$mailingList" )

awk -F "\t" '/CA/ {
print $1
print $2
print $3 $4 $5
}' <<< "$fixedMailingList"
```

```
mailingList="A
Bob Bright, 20
Charlie Cartwr
Denise Darling
Edith Ebbing,

fixedMailingLi

awk -F "\t" '/
print $1
print $2
print $3 $4 $5
}' <<< "$fixed
```

Abigail Adams

100 A Street

AlbanyCA 94706

Bob Bright

200 B Street

BakersfieldCA 93301

Edith Ebbing

500 E Street

EaglevilleCA 96110

Notes

Awk examples

```
mailingList="Abigail Adams, 100 A Street, Albany, CA 94706
Bob Bright, 200 B Street, Bakersfield, CA 93301
Charlie Cartwright, 300 C Street, Cambridge, NY 12816
Denise Darling, 400 D Street, Dale, NY 14039
Edith Ebbing, 500 E Street, Eagleville, CA 96110"

fixedMailingList=$( sed 's/, /\t/g' <<< "$mailingList" )

awk -F "\t" '/CA/ {
print $1
print $2
print $3 ", " $4 $5
}' <<< "$fixedMailingList"
```

```
mailingList="A
Bob Bright, 20
Charlie Cartwr
Denise Darling
Edith Ebbing,

fixedMailingLi

awk -F "\t" '/
print $1
print $2
print $3 ", "
}' <<< "$fixed
```

Abigail Adams

100 A Street

Albany, CA 94706

Bob Bright

200 B Street

Bakersfield, CA 93301

Edith Ebbing

500 E Street

Eagleville, CA 96110

Notes

Awk examples

```
mailingList="Abigail Adams, 100 A Street, Albany, CA 94706
Bob Bright, 200 B Street, Bakersfield, CA 93301
Charlie Cartwright, 300 C Street, Cambridge, NY 12816
Denise Darling, 400 D Street, Dale, NY 14039
Edith Ebbing, 500 E Street, Eagleville, CA 96110"

fixedMailingList=$( sed 's/, /\t/g' <<< "$mailingList" )

awk -F "\t" '/CA/ {
print $1
print $2
print $3 ", " $4 $5
print ""
}' <<< "$fixedMailingList"
```

Notes

```
mailingList="A
Bob Bright, 20
Charlie Cartwr
Denise Darling
Edith Ebbing,

fixedMailingLi

awk -F "\t" '/
print $1
print $2
print $3 ", "
print ""
}' <<< "$fixedM
```

Abigail Adams

100 A Street

Albany, CA 94706

Bob Bright

200 B Street

Bakersfield, CA 93301

Edith Ebbing

500 E Street

Eagleville, CA 96110

Notes

- Origins
- What they have in common
- When to use each
- Syntax

Notes

Feedback

bit.ly/
psumac-2024-53

Code snippets

jamf.it/asg