LING/C SC/PSYC 438/538

Lecture 7

Sandiway Fong

Today's Topics

- Getting code to run (*emailed out earlier*)
- Homework 5 Review
- Worked file I/O example:
  - `falconheavylaunch.txt`
  - I'll be assuming Bash (*from macOS or Ubuntu*)
- Sorting

Windows 10/11 Problems?

- Having trouble adapting the examples on the slides to the quoting rules in CMD or PowerShell?
  - Shell quoting rule details for Windows were given in Lecture 5
- Use Ubuntu under WSL2 (*free install from Microsoft*)
  - https://learn.microsoft.com/en-us/windows/wsl/install
- Bring your laptop to class (*follow the examples on the slides*)
  - *get the syntax sorted before the homework*!

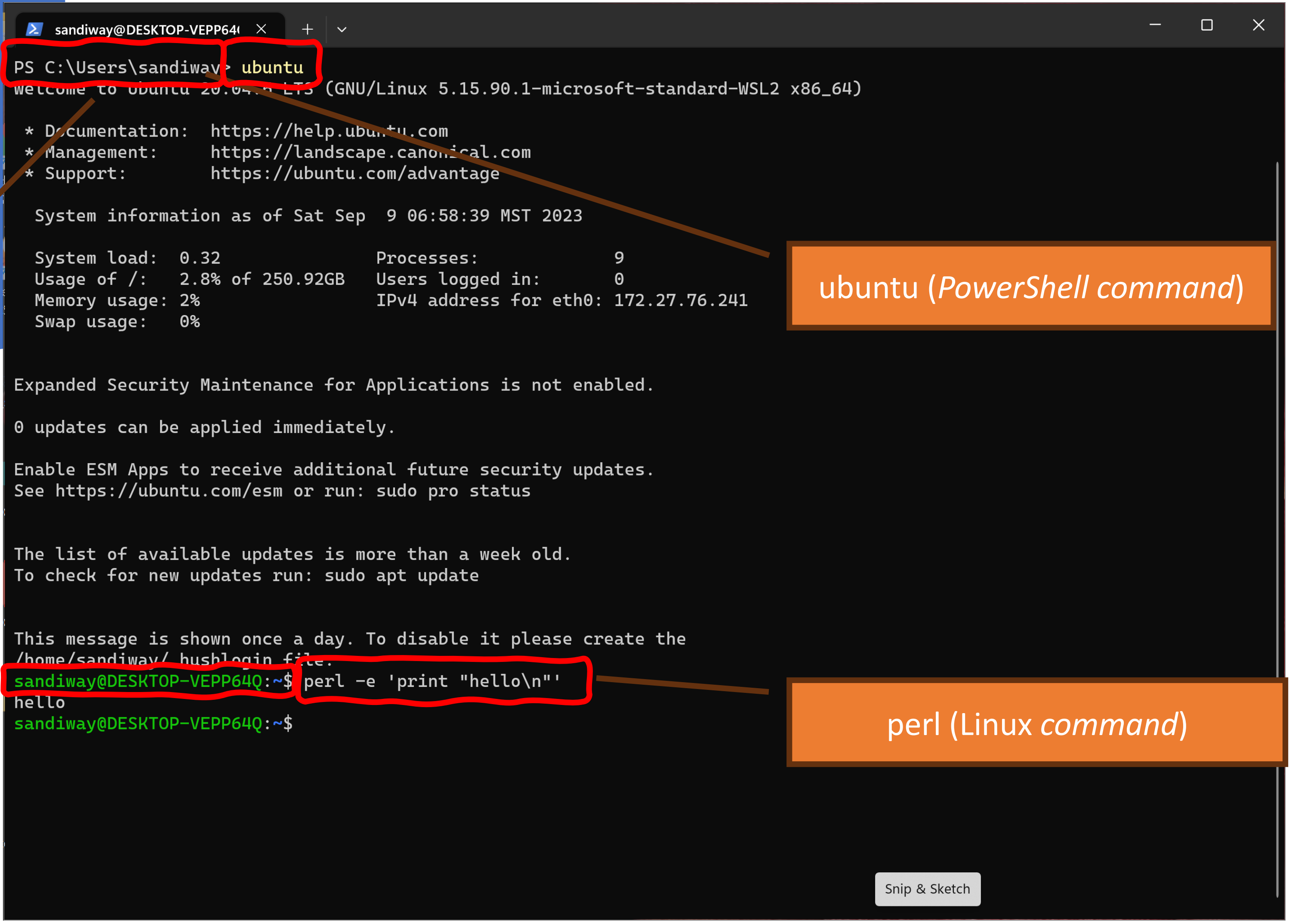
Problems in Windows 10/11

PowerShell

Ubuntu

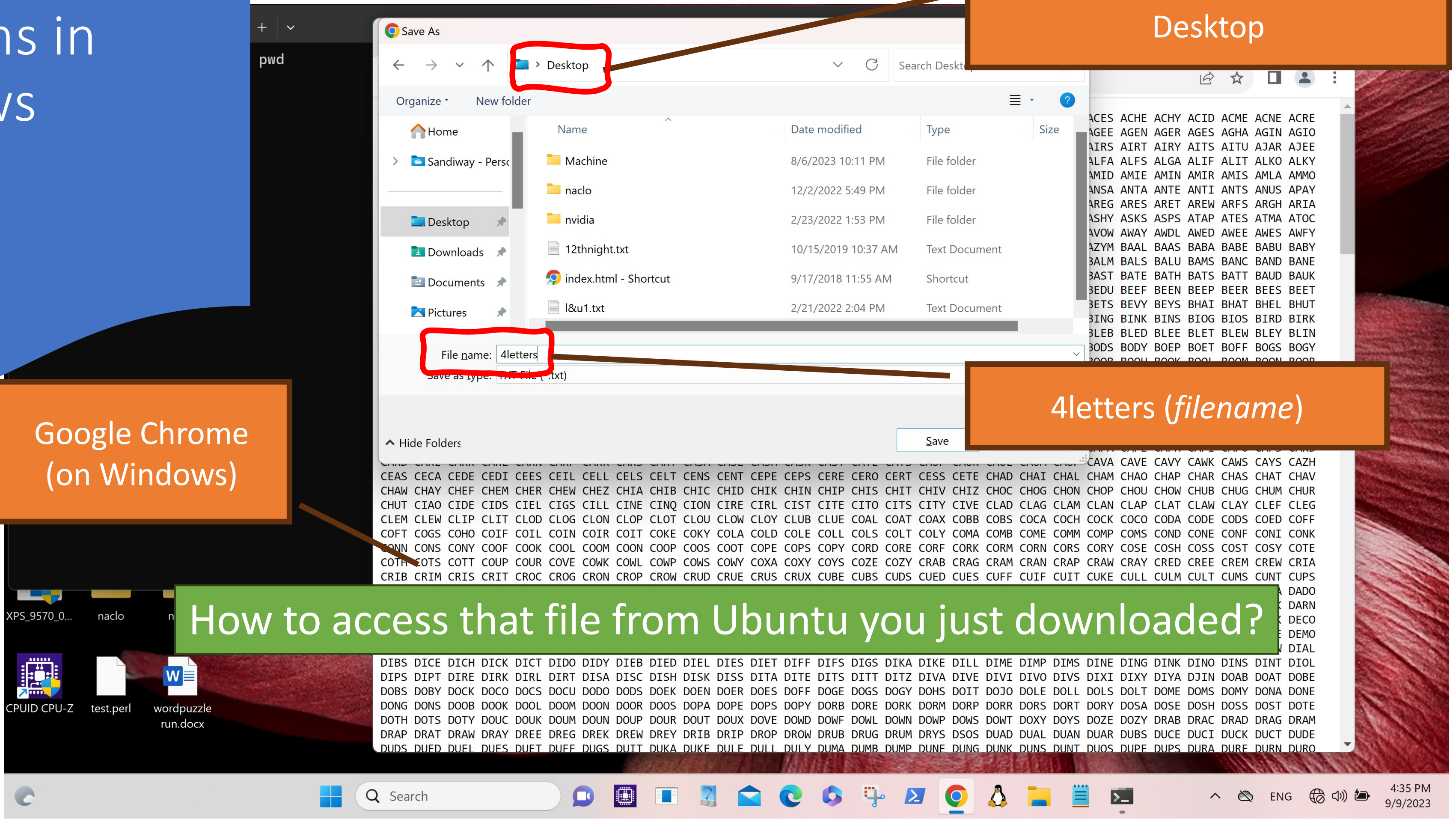
ubuntu (*PowerShell command*)

perl (Linux *command*)



```
PS C:\Users\sandiway> ubuntu
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Sep  9 06:58:39 MST 2023

  System load:  0.32               Processes:             9
  Usage of /:   2.8% of 250.92GB   Users logged in:       0
  Memory usage: 2%                 IPv4 address for eth0: 172.27.76.241
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


This message is shown once a day. To disable it please create the
/home/sandiway/.hushlogin file.
sandiway@DESKTOP-VEPP64Q:~$ perl -e 'print "hello\n"'
hello
sandiway@DESKTOP-VEPP64Q:~$
```

Problems in Windows 10/11



```
pwd
```

Desktop › Search Deskt

Name	Date modified	Type	Size
Machine	8/6/2023 10:11 PM	File folder	
naclo	12/2/2022 5:49 PM	File folder	
nvidia	2/23/2022 1:53 PM	File folder	
12thnight.txt	10/15/2019 10:37 AM	Text Document	
index.html - Shortcut	9/17/2018 11:55 AM	Shortcut	
l&u1.txt	2/21/2022 2:04 PM	Text Document	

File name: 4letters

Save as type: TXT File (*.txt)

```
ACES ACHE ACHY ACID ACME ACNE ACRE
AGEE AGEN AGER AGES AGHA AGIN AGIO
AIRS AIRT AIRY AITS AITU AJAR AJEE
ALFA ALFS ALGA ALIF ALIT ALKO ALKY
AMID AMIE AMIN AMIR AMIS AMLA AMMO
ANSA ANTA ANTE ANTI ANTS ANUS APAY
AREG ARES ARET AREW ARFS ARGH ARIA
ASHY ASKS ASPS ATAP ATES ATMA ATOC
AVOW AWAY AWDL AWED AWEE AWES AWFY
AZYM BAAL BAAS BABA BABE BABU BABY
BALM BALS BALU BAMS BANC BAND BANE
BAST BATE BATH BATS BATT BAUD BAUK
BEDU BEEF BEEN BEEP BEER BEES BEET
BETS BEVY BEYS BHAI BHAT BHEL BHUT
BING BINK BINS BIOG BIOS BIRD BIRK
BLEB BLED BLEE BLET BLEW BLEY BLIN
BODS BODY BOEP BOET BOFF BOGS BOGY

CAVA CAVE CAVY CAWK CAWS CAYS CAZH
CEAS CECA CEDE CEDI CEES CEIL CELL CELS CELT CENS CENT CEPE CEPS CERE CERO CERT CESS CETE CHAD CHAI CHAL CHAM CHAO CHAP CHAR CHAS CHAT CHAV
CHAW CHAY CHEF CHEM CHER CHEW CHEZ CHIA CHIB CHIC CHID CHIK CHIN CHIP CHIS CHIT CHIV CHIZ CHOC CHOG CHON CHOP CHOU CHOW CHUB CHUG CHUM CHUR
CHUT CIAO CIDE CIDS CIEL CIGS CILL CINE CINQ CION CIRE CIRL CIST CITE CITO CITS CITY CIVE CLAD CLAG CLAM CLAN CLAP CLAT CLAW CLAY CLEF CLEG
CLEM CLEW CLIP CLIT CLOD CLOG CLON CLOP CLOT CLOU CLOW CLOY CLUB CLUE COAL COAT COAX COBB COBS COCA COCH COCK COCO CODA CODE CODS COED COFF
COFT COGS COHO COIF COIL COIN COIR COIT COKE COKY COLA COLD COLE COLL COLS COLT COLY COMA COMB COME COMM COMP COMS COND CONE CONF CONI CONK
CONN CONS CONY COOF COOK COOL COOM COON COOP COOS COOT COPE COPS COPY CORD CORE CORF CORK CORM CORN CORS CORY COSE COSH COSS COST COSY COTE
COTH COTS COTT COUP COUR COVE COWK COWL COWP COWS COWY COXA COXY COYS COZE COZY CRAB CRAG CRAM CRAN CRAP CRAW CRAY CRED CREE CREM CREW CRIA
CRIB CRIM CRIS CRIT CROC CROG CRON CROP CROW CRUD CRUE CRUS CRUX CUBE CUBS CUDS CUED CUES CUFF CUIF CUIT CUKE CULL CULM CULT CUMS CUNT CUPS
DADO
DARN
DECO
DEMO
DIAL
DIBS DICE DICH DICK DICT DIDO DIDY DIEB DIED DIEL DIES DIET DIFF DIFS DIGS DIKA DIKE DILL DIME DIMP DIMS DINE DING DINK DINO DINS DINT DIOL
DIPS DIPT DIRE DIRK DIRL DIRT DISA DISC DISH DISK DISS DITA DITE DITS DITT DITZ DIVA DIVE DIVI DIVO DIVS DIXI DIXY DIYA DJIN DOAB DOAT DOBE
DOBS DOBY DOCK DOCO DOCS DOCU DODO DODS DOEK DOEN DOER DOES DOFF DOGE DOGS DOGY DOHS DOIT DOJO DOLE DOLL DOLS DOLT DOME DOMS DOMY DONA DONE
DONG DONS DOOB DOOK DOOL DOOM DOON DOOR DOOS DOPA DOPE DOPS DOPY DORB DORE DORK DORM DORP DORR DORS DORT DORY DOSA DOSE DOSH DOSS DOST DOTE
DOTH DOTS DOTY DOUC DOUK DOUM DOUN DOUP DOUR DOUT DOUX DOVE DOWD DOWF DOWL DOWN DOWP DOWS DOWT DOXY DOYS DOZE DOZY DRAB DRAC DRAD DRAG DRAM
DRAP DRAT DRAW DRAY DREE DREG DREK DREW DREY DRIB DRIP DROP DROW DRUB DRUG DRUM DRYS DSOS DUAD DUAL DUAN DUAR DUBS DUCE DUCI DUCK DUCT DUDE
```

4:35 PM 9/9/2023

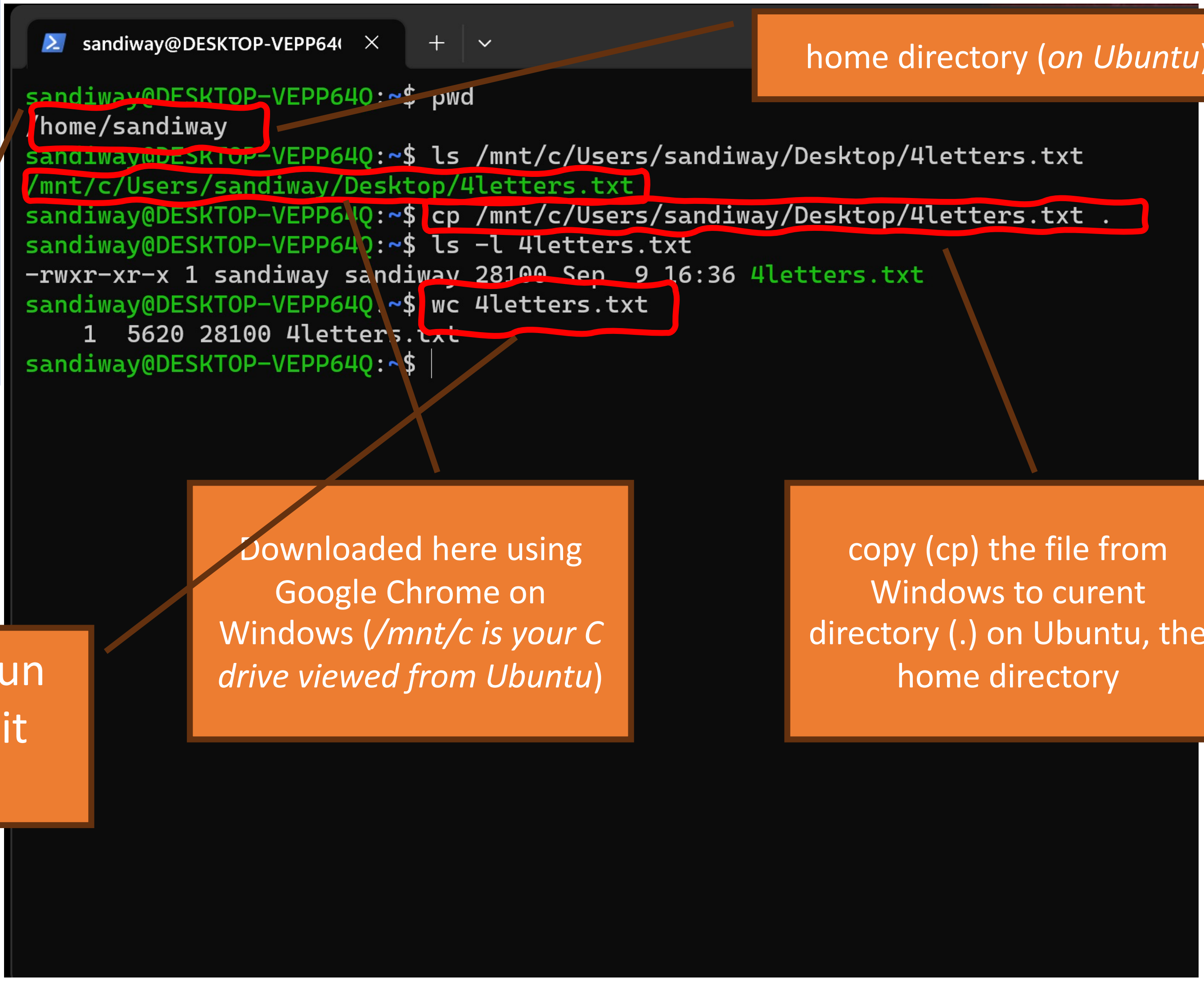
Desktop

4letters (*filename*)

Google Chrome (on Windows)

How to access that file from Ubuntu you just downloaded?

Problems in Windows 10/11



```
sandiway@DESKTOP-VEPP64Q:~$ pwd
/home/sandiway
sandiway@DESKTOP-VEPP64Q:~$ ls /mnt/c/Users/sandiway/Desktop/4letters.txt
/mnt/c/Users/sandiway/Desktop/4letters.txt
sandiway@DESKTOP-VEPP64Q:~$ cp /mnt/c/Users/sandiway/Desktop/4letters.txt .
sandiway@DESKTOP-VEPP64Q:~$ ls -l 4letters.txt
-rwxr-xr-x 1 sandiway sandiway 28100 Sep  9 16:36 4letters.txt
sandiway@DESKTOP-VEPP64Q:~$ wc 4letters.txt
    1  5620 28100 4letters.txt
sandiway@DESKTOP-VEPP64Q:~$
```

home directory (*on Ubuntu*)

Ubuntu (*notice prompt*)

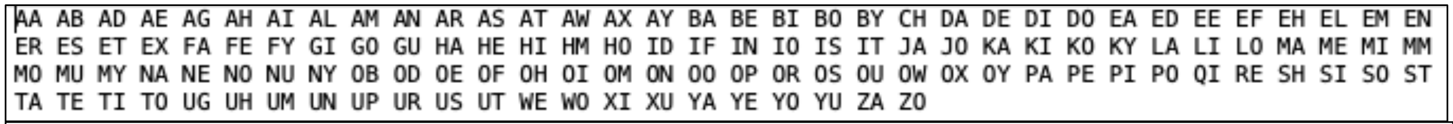
Downloaded here using Google Chrome on Windows (*/mnt/c is your C drive viewed from Ubuntu*)

copy (cp) the file from Windows to curent directory (.) on Ubuntu, the home directory

Now you can run wc or perl on it locally

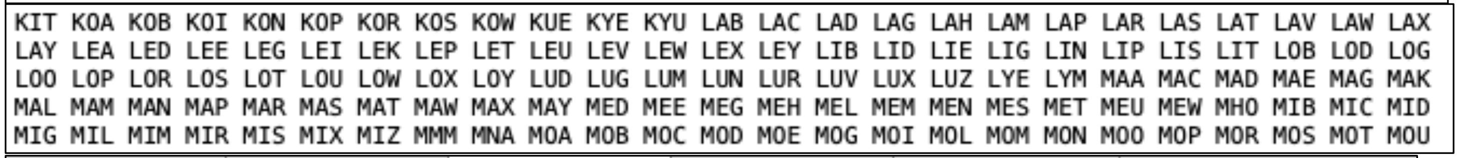
Homework 5 Review

- There are 4 files on the course website:
  - *n*`letters.txt` *n* = 2, 3, 4 & 5

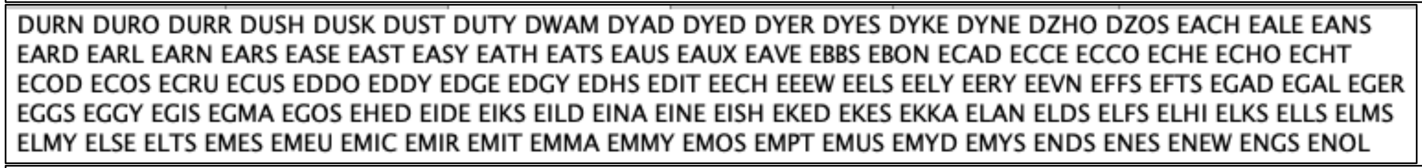


```
AA AB AD AE AG AH AI AL AM AN AR AS AT AW AX AY BA BE BI BO BY CH DA DE DI DO EA ED EE EF EH EL EM EN
ER ES ET EX FA FE FY GI GO GU HA HE HI HM HO ID IF IN IO IS IT JA JO KA KI KO KY LA LI LO MA ME MI MM
MO MU MY NA NE NO NU NY OB OD OE OF OH OI OM ON OO OP OR OS OU OW OX OY PA PE PI PO QI RE SH SI SO ST
TA TE TI TO UG UH UM UN UP UR US UT WE WO XI XU YA YE YO YU ZA ZO
```

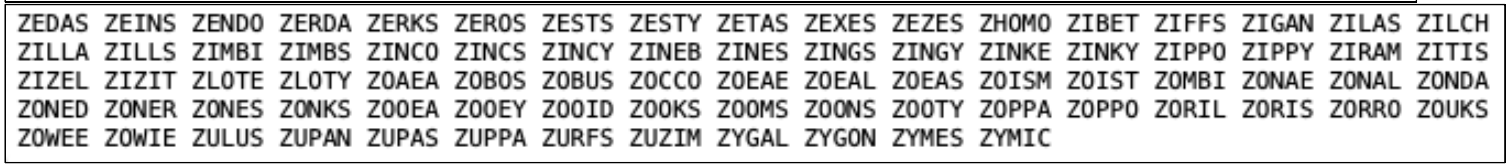
`2letters.txt`



```
KIT KOA KOB KOI KON KOP KOR KOS KOW KUE KYE KYU LAB LAC LAD LAG LAH LAM LAP LAR LAS LAT LAV LAW LAX
LAY LEA LED LEE LEG LEI LEK LEP LET LEU LEV LEW LEX LEY LIB LID LIE LIG LIN LIP LIS LIT LOB LOD LOG
LOO LOP LOR LOS LOT LOU LOW LOX LOY LUD LUG LUM LUN LUR LUV LUX LUZ LYE LYM MAA MAC MAD MAE MAG MAK
MAL MAM MAN MAP MAR MAS MAT MAW MAX MAY MED MEE MEG MEH MEL MEM MEN MES MET MEU MEW MHO MIB MIC MID
MIG MIL MIM MIR MIS MIX MIZ MMM MNA MOA MOB MOC MOD MOE MOG MOI MOL MOM MON MOO MOP MOR MOS MOT MOU
```



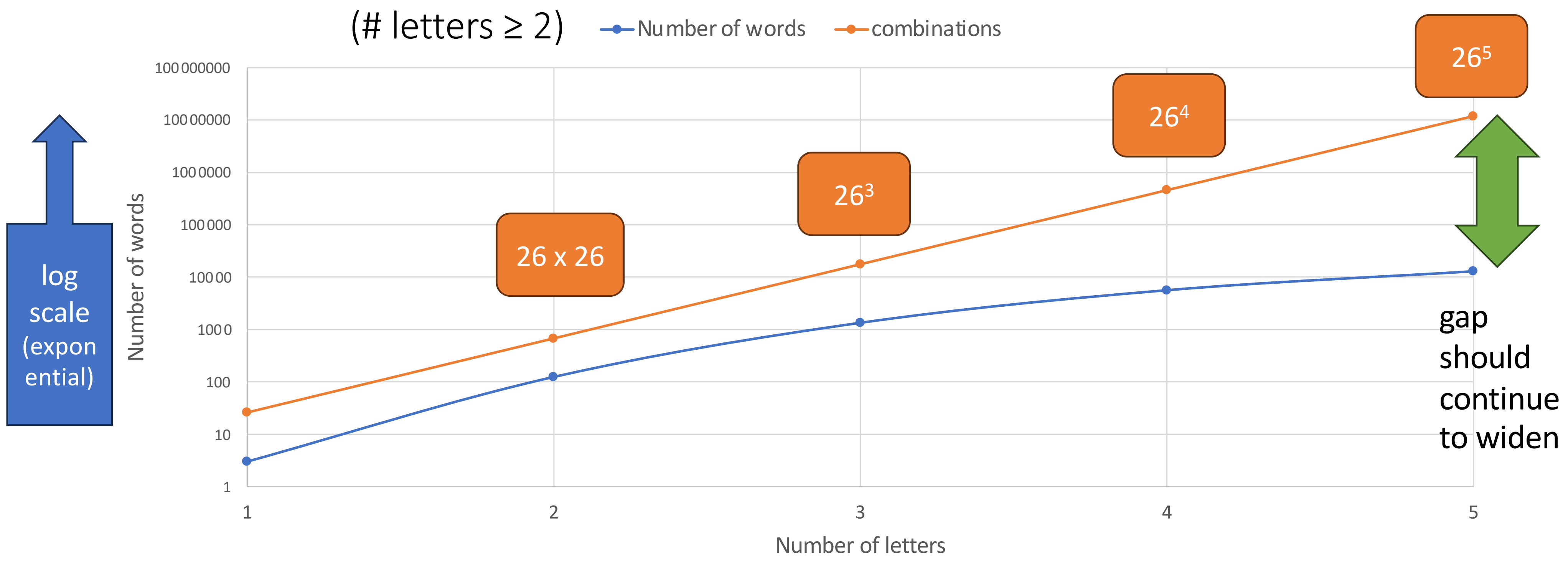
```
DURN DURO DURR DUSH DUSK DUST DUTY DWAM DYAD DYED DYER DYES DYKE DYNE DZHO DZOS EACH EALE EANS
EARD EARL EARN EARS EASE EAST EASY EATH EATS EAUS EAUX EAVE EBBS EBON ECAD ECCE ECCO ECHE ECHO ECHT
ECOD ECOS ECRU ECUS EDDO EDDY EDGE EDGY EDHS EDIT EECH EEEW EELS EELY EERY EEVN EFFS EFTS EGAD EGAL EGER
EGGS EGGY EGIS EGMA EGOS EHED EIDE EIKS EILD EINA EINE EISH EKED EKES EKKA ELAN ELDS ELFS ELHI ELKS ELLS ELMS
ELMY ELSE ELTS EMES EMEU EMIC EMIR EMIT EMMA EMMY EMOS EMPT EMUS EMYD EMYS ENDS ENES ENEW ENGS ENOL
```



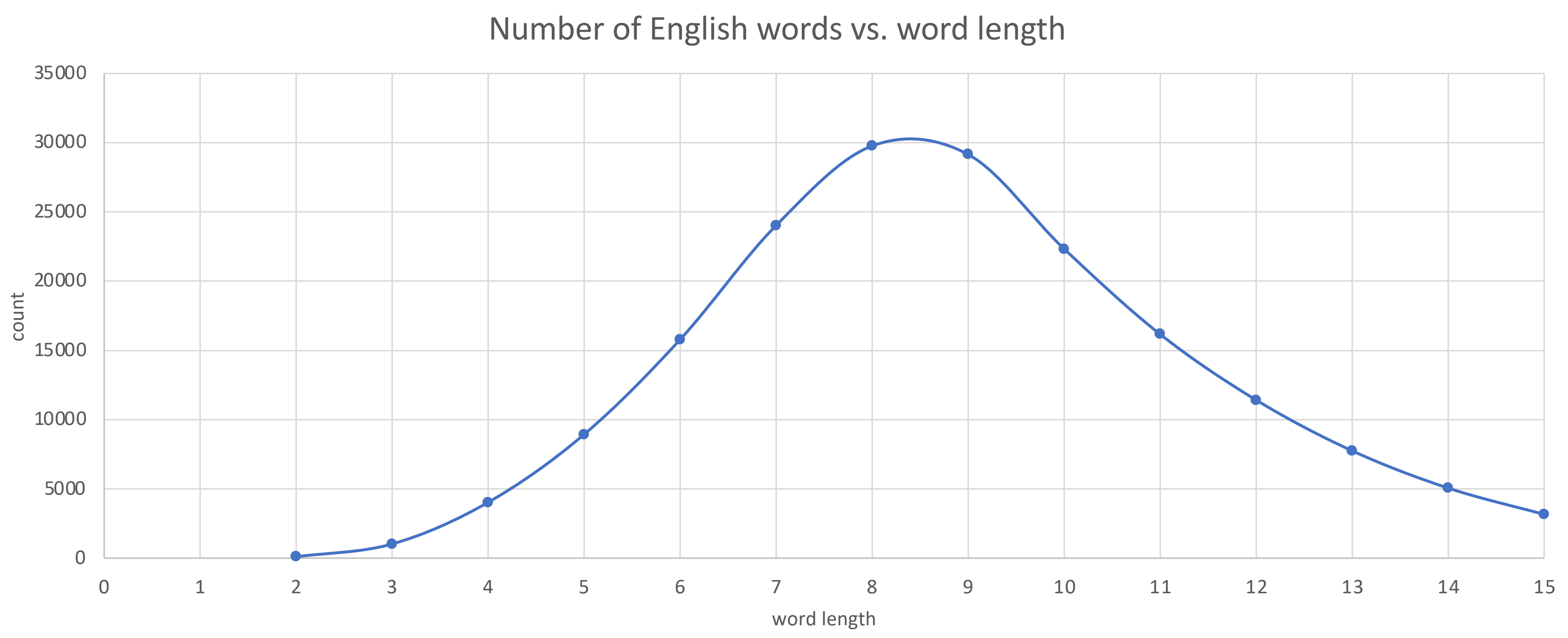
```
ZEDAS ZEINS ZENDO ZERDA ZERKS ZEROS ZESTS ZESTY ZETAS ZEXES ZEZES ZHOMO ZIBET ZIFFS ZIGAN ZILAS ZILCH
ZILLA ZILLS ZIMBI ZIMBS ZINCO ZINCS ZINCY ZINEB ZINES ZINGS ZINGY ZINKE ZINKY ZIPPO ZIPPY ZIRAM ZITIS
ZIZEL ZIZIT ZLOTE ZLOTY ZOAEA ZOBOS ZOBUS ZOCCO ZOEAE ZOEAL ZOEAS ZOISM ZOIST ZOMBI ZONAE ZONAL ZONDA
ZONED ZONER ZONES ZONKS ZOOEA ZOOEY ZOOID ZOOKS ZOOMS ZOONS ZOOTY ZOPPA ZOPPO ZORIL ZORIS ZORRO ZOUKS
ZOWEE ZOWIE ZULUS ZUPAN ZUPAS ZUPPA ZURFS ZUZIM ZYGAL ZYGON ZYMES ZYMIC
```

`5letters.txt`

A particular set of Scrabble Words



Scrabble Words Statistics



Homework 5 Review

- Question 1: print out the number of words.
- Example:

```
$ perl -e 'open($fh, $ARGV[0]); $line = <$fh>; @a = split("
",$line); print $#a+1,"\n"' 3letters.txt
124
```

- Bash:

```
$ for n in {2..5}; do
>  perl -e 'open $fh, $ARGV[0]; @a = split " ", <$fh>; print
"$ARGV[0]: ",$#a+1,"\n"' ${n}letters.txt
> done
2letters.txt: 124
3letters.txt: 1340
4letters.txt: 5620
5letters.txt: 12915
```

Homework 5 Review

- Question 2: which words spell the same forwards as backwards?

```
$ perl -e 'open $fh, $ARGV[0]; for (split " ",<$fh>)
{print "$_\n" if ($_ eq reverse $_)}' 4letters.txt

$ for n in {2..5}; do

> perl -e 'open $fh, $ARGV[0]; for (split " ",<$fh>)
{print "$_\n" if ($_ eq reverse $_)}' ${n}letters.txt |
wc -l

> done
       4
      69
      18
      40
```

Homework 5 Review

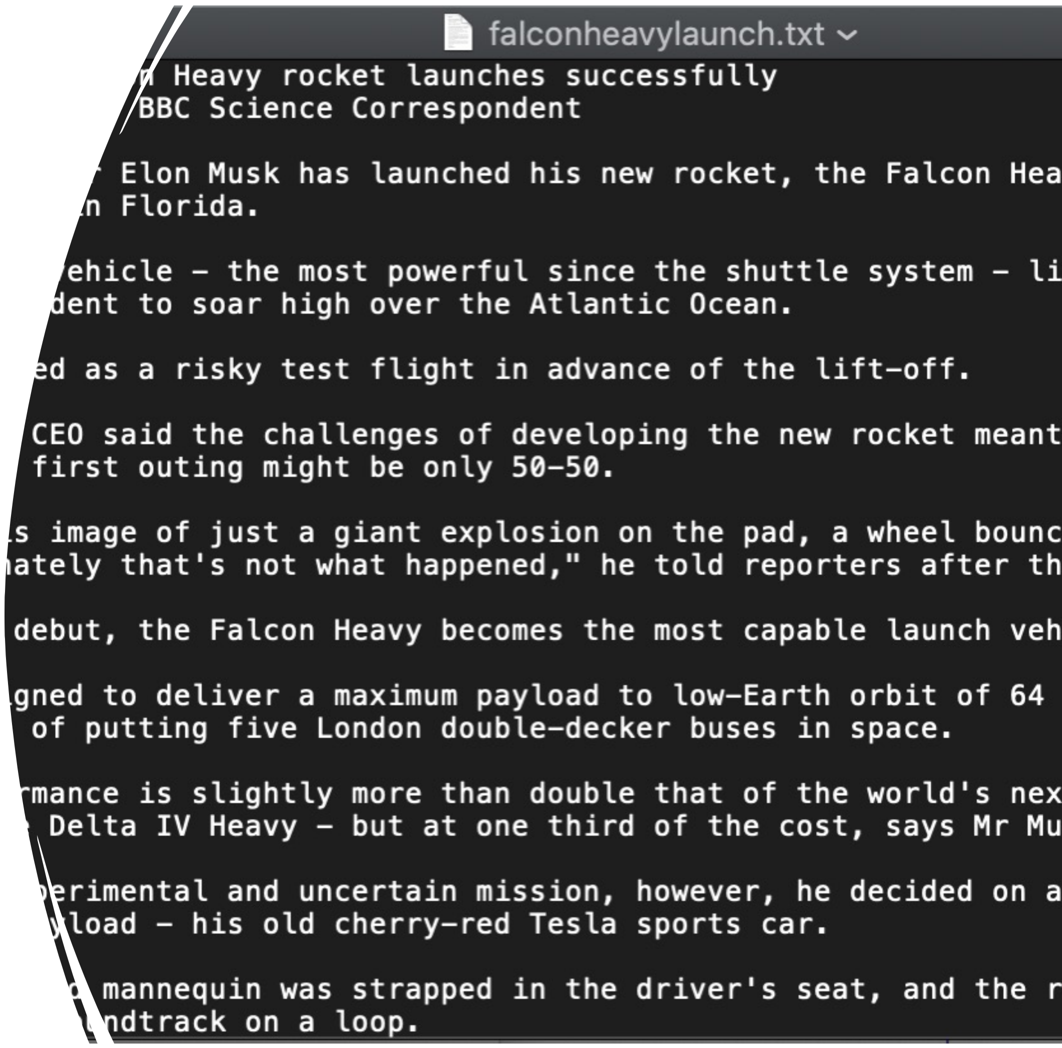
Is it true for n =6,7,8,…?

How long before the streak is broken?

- Results:
  - AA EE MM OO
  - ABA AGA AHA AIA AKA ALA AMA ANA AUA AVA AWA BIB BOB BUB DAD DID DOD DUD EKE EME ENE ERE EVE EWE EYE GAG GIG HAH HEH HOH HUH IWI KAK MAM MEM MIM MMM MOM MUM NAN NON NUN OBO OHO ONO OXO PAP PEP PIP POP PUP SIS SOS SUS TAT TET TIT TOT TUT ULU UMU UTU VAV WAW WOW YAY ZIZ ZUZ ZZZ
  - ABBA ACCA ANNA BOOB DEED ECCE ESSE GOOG KEEK KOOK NAAN NOON OPPO OTTO PEEP POOP SEES TOOT
  - ALULA ANANA AYAYA CIVIC DEKED DELED DERED DEWED KAIAK KAYAK LAHAL LAVAL LEMEL LEVEL MADAM MALAM MINIM QAJAQ RADAR REFER ROTOR SAGAS SAMAS SEDES SELES SEMES SENES SERES SEXES SHAHS SIMIS SIRIS SOLOS STATS STETS STOTS SULUS SUSUS TENET TOROT

Perl: file I/O

- **What does this code do?**
- `perl -e 'open $f, "falconheavylaunch.txt"; while (<$f>) {print((split " ")[0],"\n")}'`
- `perl -e 'open $f, "falconheavylaunch.txt"; while (<$f>) {print((split " ")[0],"\n")}'  | wc -l`
- *reports 49 lines*
- `perl -e 'open $f, "falconheavylaunch.txt"; while (<$f>) {@words = split " "; $sum+=@words}; print $sum'`
- *reports 669 words*



falconheavylaunch.txt

```
n Heavy rocket launches successfully
BBC Science Correspondent

r Elon Musk has launched his new rocket, the Falcon Hea
in Florida.

ehicle - the most powerful since the shuttle system - li
dent to soar high over the Atlantic Ocean.

ed as a risky test flight in advance of the lift-off.

CEO said the challenges of developing the new rocket meant
first outing might be only 50-50.

s image of just a giant explosion on the pad, a wheel bounc
ately that's not what happened," he told reporters after th

debut, the Falcon Heavy becomes the most capable launch veh

gned to deliver a maximum payload to low-Earth orbit of 64
of putting five London double-decker buses in space.

rmance is slightly more than double that of the world's nex
 Delta IV Heavy - but at one third of the cost, says Mr Mu

perimental and uncertain mission, however, he decided on a
yload - his old cherry-red Tesla sports car.

 mannequin was strapped in the driver's seat, and the r
ndtrack on a loop.
```

Perl: file I/O

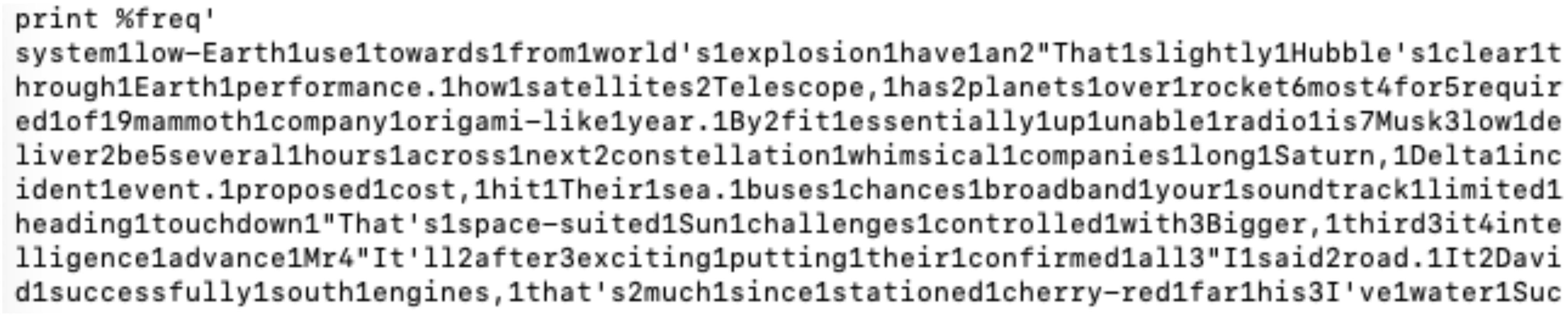
And another bit more. Looking ahead:

- `%freq` (is a hash table, or key/value `dict` – Python)
- `$freq{`*`word`*`}` will hold the number of times *word* occurs
- `keys %freq` will be the list of keys, i.e. *words*, recorded

-
```
perl -e 'open $f, "falconheavylaunch.txt"; while (<$f>) {for
(split " ") {$freq{$_}++}};'
```

Perl: file I/O

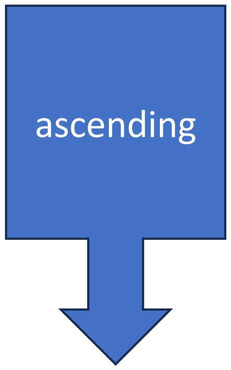
- Unfortunately:
  - `perl -e 'open $f, "falconheavylaunch.txt"; while (<$f>) {for (split " ") {$freq{$_}++}}; print %freq'`
- pretty-printing for hash tables is non-existent:

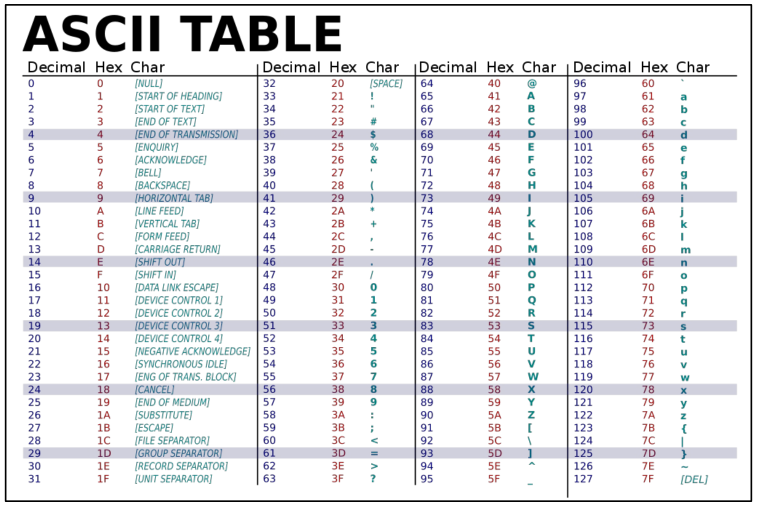


```
print %freq'
system1low-Earth1use1towards1from1world's1explosion1have1an2"That1slightly1Hubble's1clear1t
hrough1Earth1performance.1how1satellites2Telescope,1has2planets1over1rocket6most4for5requir
ed1of19mammoth1company1origami-like1year.1By2fit1essentially1up1unable1radio1is7Musk3low1de
liver2be5several1hours1across1next2constellation1whimsical1companies1long1Saturn,1Delta1inc
ident1event.1proposed1cost,1hit1Their1sea.1buses1chances1broadband1your1soundtrack1limited1
heading1touchdown1"That's1space-suited1Sun1challenges1controlled1with3Bigger,1third3it4inte
lligence1advance1Mr4"It'll2after3exciting1putting1their1confirmed1all3"I1said2road.1It2Davi
d1successfully1south1engines,1that's2much1since1stationed1cherry-red1far1his3I've1water1Suc
```

Perl Arrays: sorting

- Example:
  - `perl -e '@a=sort @ARGV; print "@a\n"' 3 2 99 7 13 100`
  - `100 13 2 3 7 99`
  - `perl -e '@a=sort @ARGV; print "@a\n"' mary 77 john vicky Pete`
  - `77 Pete john mary vicky`
- Standard order:





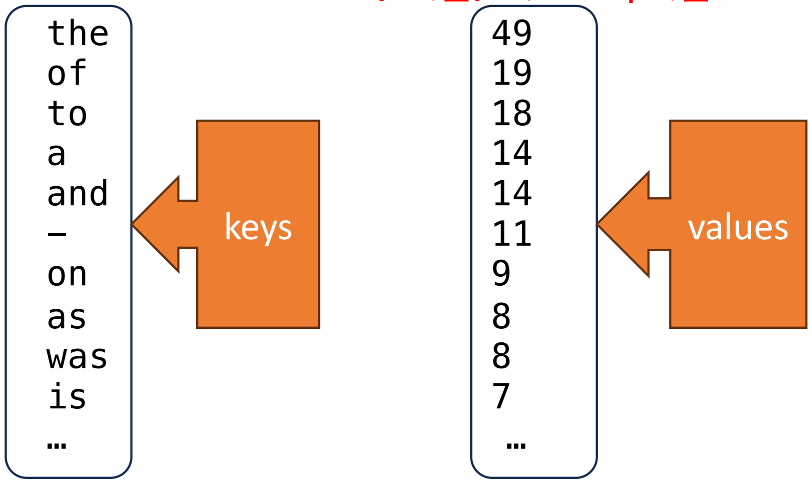
ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Perl: file I/O

- Print the word frequency table sorted in descending order:
  - `sort {$freq{$b} <=> $freq{a}} keys %freq`

  -
    ```
    perl -e 'open $f, "falconheavylaunch.txt"; while (<$f>) {for (split " ") {$freq{$_}++}}; for (sort {$freq{$b} <=> $freq{$a}} keys %freq) {printf "%-20s %s\n", $_, $freq{$_}}' | head -10
    ```



```
the
of
to
a
and
–
on
as
was
is
…
```

```
49
19
18
14
14
11
9
8
8
7
…
```

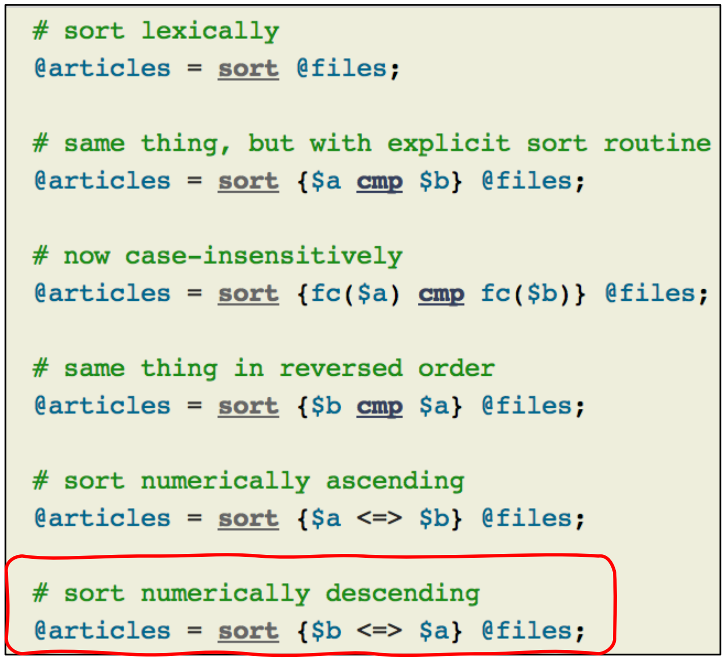
Perl Arrays: sorting

https://perldoc.perl.org/functions/sort

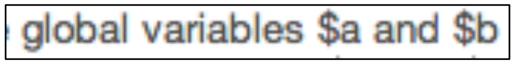
Numeric sort?

- **Idea**: to pass an inline comparison function as parameter…
- **Note**: function fc (fold case)

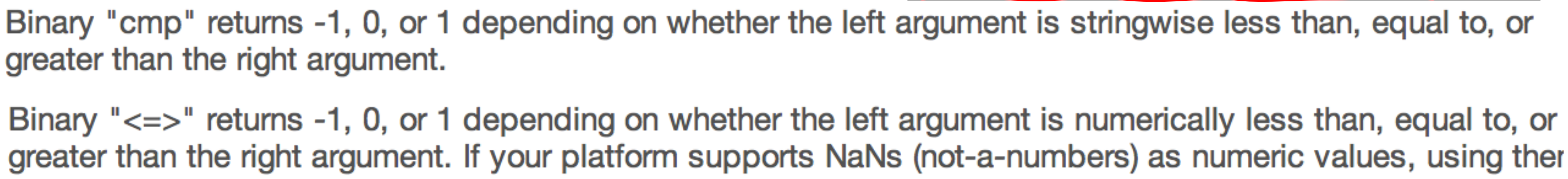
```
use feature 'fc';
```



```
# sort lexically
@articles = sort @files;

# same thing, but with explicit sort routine
@articles = sort {$a cmp $b} @files;

# now case-insensitively
@articles = sort {fc($a) cmp fc($b)} @files;

# same thing in reversed order
@articles = sort {$b cmp $a} @files;

# sort numerically ascending
@articles = sort {$a <=> $b} @files;

# sort numerically descending
@articles = sort {$b <=> $a} @files;
```



global variables $a and $b

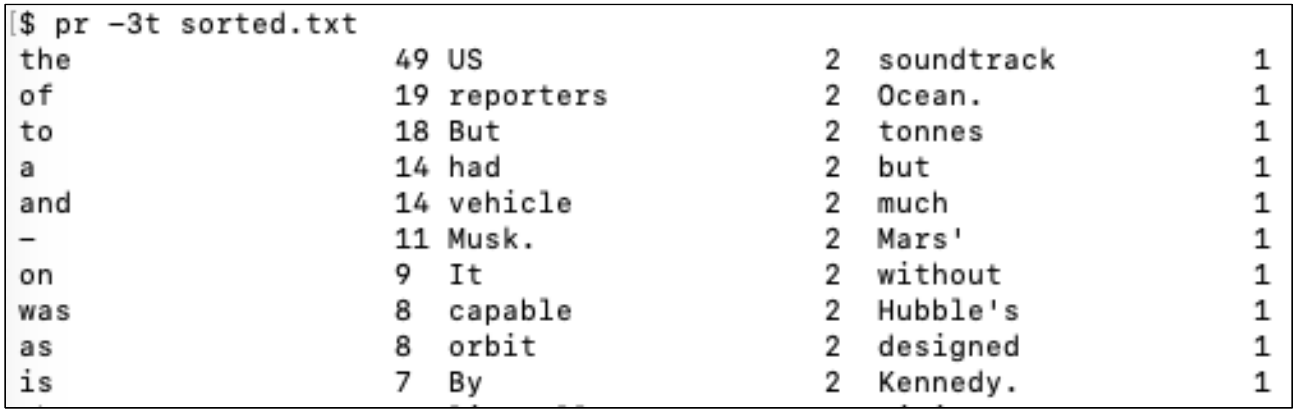


Binary "cmp" returns -1, 0, or 1 depending on whether the left argument is stringwise less than, equal to, or greater than the right argument.

Binary "<=>" returns -1, 0, or 1 depending on whether the left argument is numerically less than, equal to, or greater than the right argument. If your platform supports NaNs (not-a-numbers) as numeric values, using ther

Perl: file I/O

- Save the sorted output into a file `sorted.txt`
  - `perl -e 'open $f, "falconheavylaunch.txt"; while (<$f>) {for (split " ") {$freq{$_}++}}; for (sort {$freq{$b} <=> $freq{$a}} keys %freq) {printf "%-20s %s\n", $_, $freq{$_}}' > sorted.text`
- Then show it in 3 column format
  - Terminal command `pr -3t sorted.txt`



```
$ pr -3t sorted.txt
the                  49 US                   2  soundtrack          1
of                   19 reporters            2  Ocean.              1
to                   18 But                  2  tonnes              1
a                    14 had                  2  but                 1
and                  14 vehicle              2  much                1
-                    11 Musk.                2  Mars'               1
on                   9  It                   2  without             1
was                  8  capable              2  Hubble's            1
as                   8  orbit                2  designed            1
is                   7  By                   2  Kennedy.            1
```

Perl: file I/O

Strictly speaking, we'd like to remove prefix/suffix punctuation,

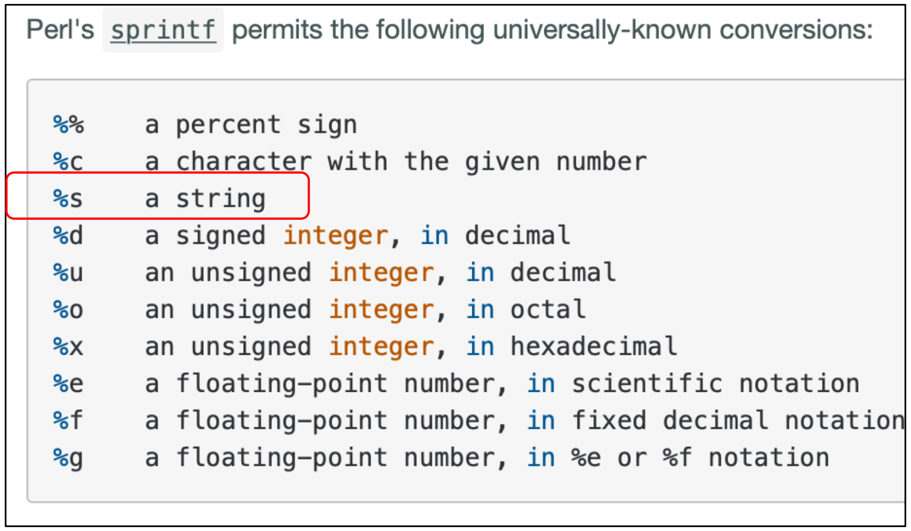
- i.e. avoid accumulating "words" like:

```
"It'll                2
…
sea.                  1
literally."           1
```

- We haven't covered regex yet:
  - `for $w (split " ") {$w=~s/(^\W+)|(\W+$)//g; $freq{$w}++}`

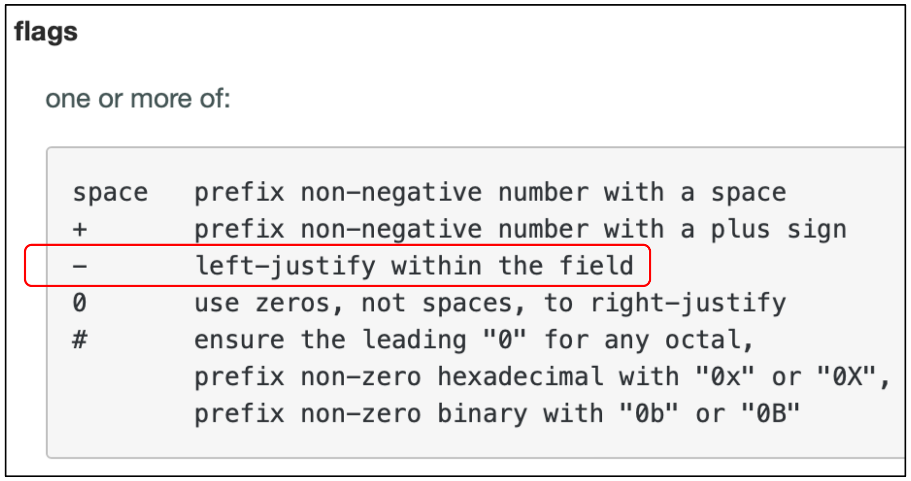
Formatted printing

```
printf "%-20s %s\n", $_, $freq{$_}
```



Perl's sprintf permits the following universally-known conversions:

```
%%    a percent sign
%c    a character with the given number
%s    a string
%d    a signed integer, in decimal
%u    an unsigned integer, in decimal
%o    an unsigned integer, in octal
%x    an unsigned integer, in hexadecimal
%e    a floating-point number, in scientific notation
%f    a floating-point number, in fixed decimal notation
%g    a floating-point number, in %e or %f notation
```



flags

one or more of:

```
space   prefix non-negative number with a space
+       prefix non-negative number with a plus sign
-       left-justify within the field
0       use zeros, not spaces, to right-justify
#       ensure the leading "0" for any octal,
        prefix non-zero hexadecimal with "0x" or "0X",
        prefix non-zero binary with "0b" or "0B"
```

https://perldoc.perl.org/functions/sprintf

Perl: file I/O

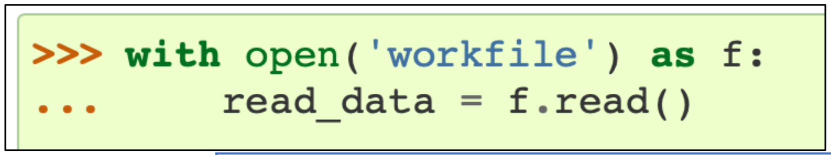
- Alternative: use command sort (instead of Perl sort):
  - `sort -k 2nr` *sort by key field #2 numerically* (n) *in reverse order* (r)
  - https://man7.org/linux/man-pages/man1/sort.1.html

  -
    ```
    perl -e 'open $f, "falconheavylaunch.txt"; while (<$f>) {for (split " ")
    {$freq{$_}++}}; for (keys %freq) {print $_," ", $freq{$_}, "\n"}' | sort -k
    2nr | pr -t5
    ```

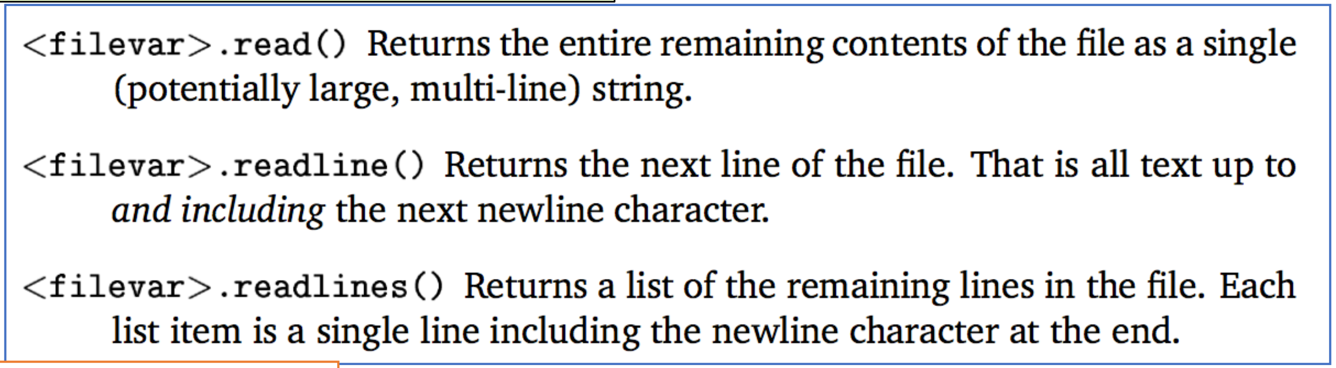
```
the 49           reporters 2      With 1           ever 1           military. 1
of 19            rocket, 2        about 1          exciting 1       missed 1
to 18            said 2           across 1         experimental     mission, 1
a 14             satellites 2     advance 1        explosion 1      moons. 1
and 14           strapped 2       allied 1         far 1            much 1
– 11             that's 2         around 1         fascinating 1    not 1
on 9             told 2           attempt 1        final 1  occasions, 1
as 8             upper-stage 2    available. 1     fire 1           occurring 1
was 8            vehicle 2        back 1           first 1  old 1
is 7             where 2          batches 1        fit 1            open 1
Falcon 6         "I 1             becomes 1        five 1           or 1
The 6            "That 1          been 1           flight 1         orbit. 1
rocket 6         "That's 1        believes 1       folded 1         origami-like
…
```

Python: Files

- Like all other programming languages, uses a file handle, called **file variable**: open()
- `infile = open("file.txt","r")`   `outfile = open("results.txt","w")`



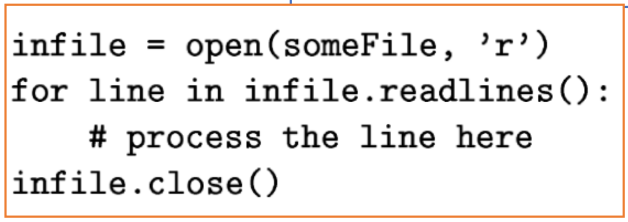
```
>>> with open('workfile') as f:
...     read_data = f.read()
```



`<filevar>.read()` Returns the entire remaining contents of the file as a single (potentially large, multi-line) string.

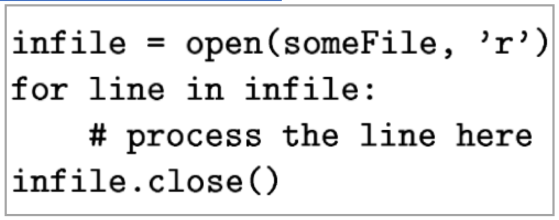
`<filevar>.readline()` Returns the next line of the file. That is all text up to *and including* the next newline character.

`<filevar>.readlines()` Returns a list of the remaining lines in the file. Each list item is a single line including the newline character at the end.



```
infile = open(someFile, 'r')
for line in infile.readlines():
    # process the line here
infile.close()
```

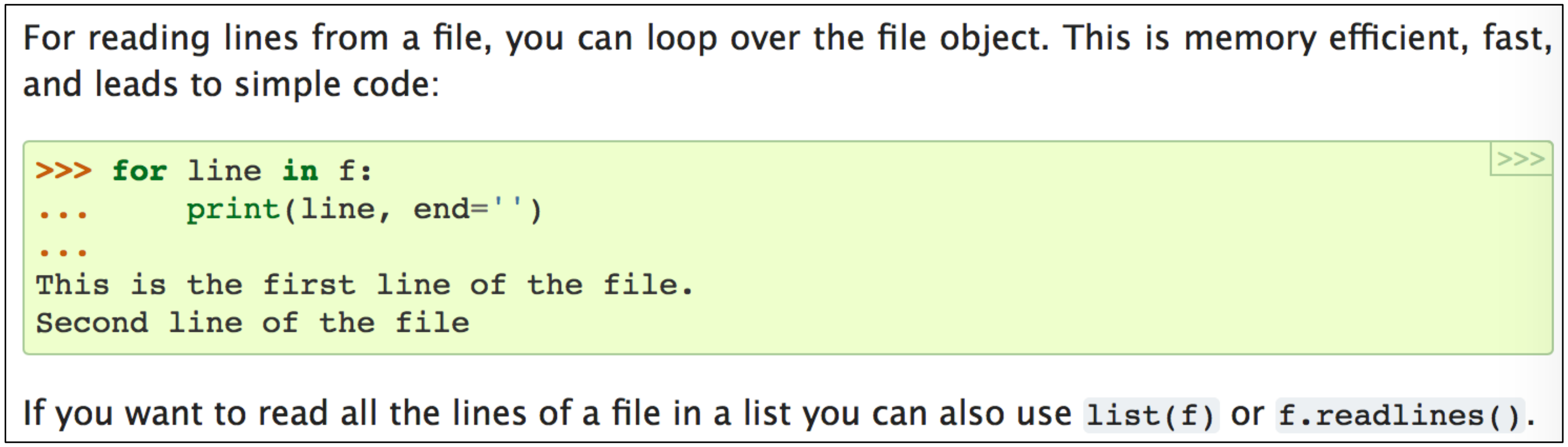
same as



```
infile = open(someFile, 'r')
for line in infile:
    # process the line here
infile.close()
```

Python: Files

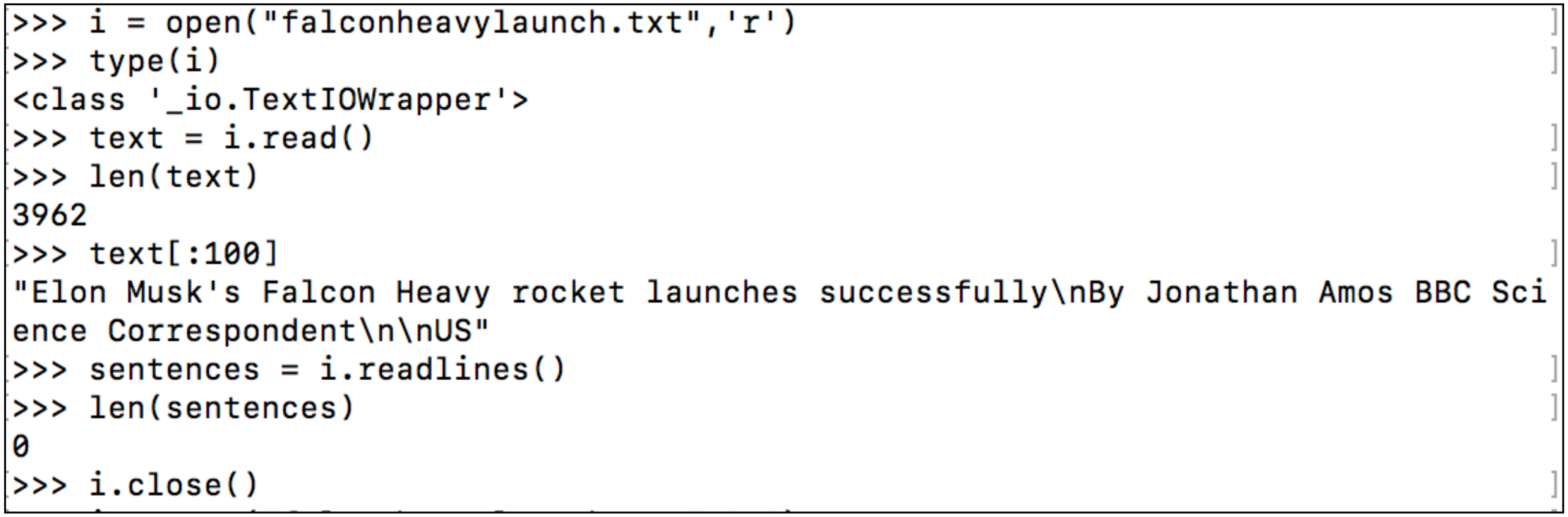
- https://docs.python.org/3/tutorial/inputoutput.html#reading-and-writing-files


For reading lines from a file, you can loop over the file object. This is memory efficient, fast, and leads to simple code:

```
>>> for line in f:
...     print(line, end='')
...
This is the first line of the file.
Second line of the file
```

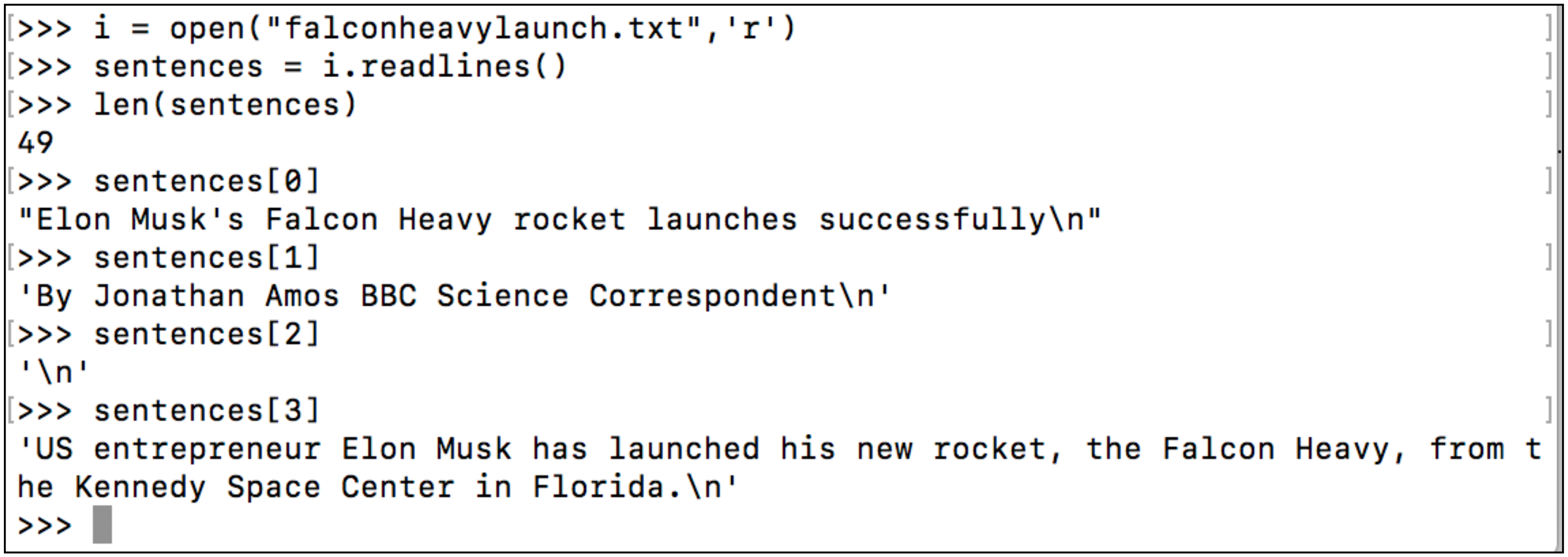
If you want to read all the lines of a file in a list you can also use `list(f)` or `f.readlines()`.

Python: Files



```
>>> i = open("falconheavylaunch.txt",'r')
>>> type(i)
<class '_io.TextIOWrapper'>
>>> text = i.read()
>>> len(text)
3962
>>> text[:100]
"Elon Musk's Falcon Heavy rocket launches successfully\nBy Jonathan Amos BBC Sci
ence Correspondent\n\nUS"
>>> sentences = i.readlines()
>>> len(sentences)
0
>>> i.close()
```

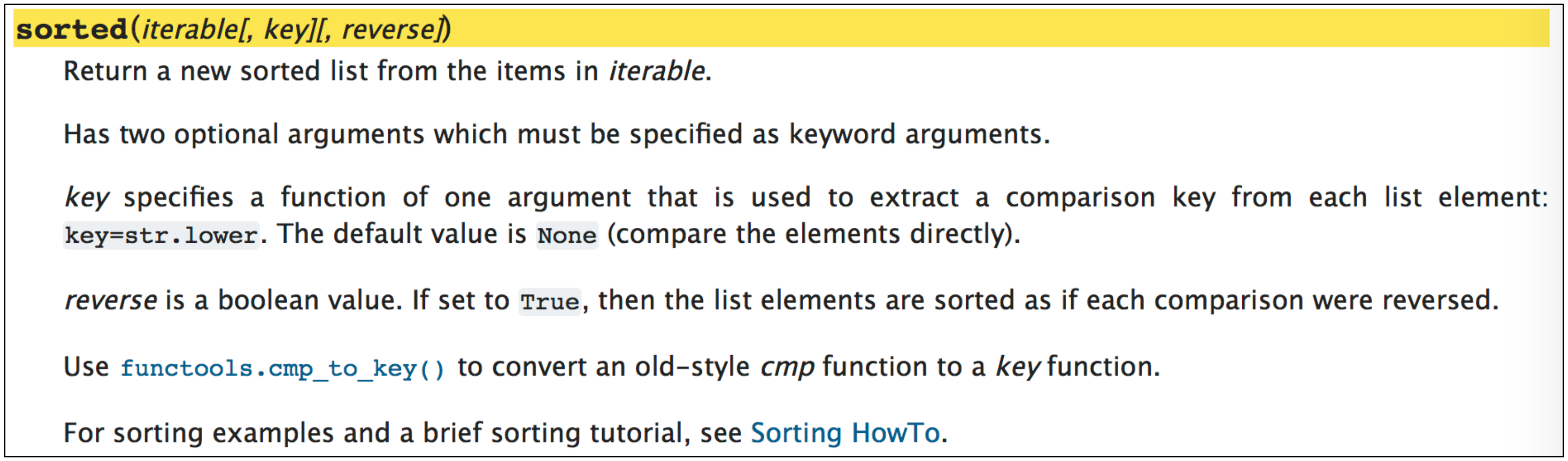
Python: Files



```
>>> i = open("falconheavylaunch.txt",'r')
>>> sentences = i.readlines()
>>> len(sentences)
49
>>> sentences[0]
"Elon Musk's Falcon Heavy rocket launches successfully\n"
>>> sentences[1]
'By Jonathan Amos BBC Science Correspondent\n'
>>> sentences[2]
'\n'
>>> sentences[3]
'US entrepreneur Elon Musk has launched his new rocket, the Falcon Heavy, from t
he Kennedy Space Center in Florida.\n'
>>> 
```

Python: sorting

https://docs.python.org/3/howto/sorting.html



sorted(*iterable[, key][, reverse]*)

Return a new sorted list from the items in *iterable*.

Has two optional arguments which must be specified as keyword arguments.

key specifies a function of one argument that is used to extract a comparison key from each list element: `key=str.lower`. The default value is `None` (compare the elements directly).

reverse is a boolean value. If set to `True`, then the list elements are sorted as if each comparison were reversed.

Use `functools.cmp_to_key()` to convert an old-style *cmp* function to a *key* function.

For sorting examples and a brief sorting tutorial, see Sorting HowTo.