TERMINAL

Published : 2011-03-14
License : None

Terminal Activity

1. BASIC COMMANDS

The **Terminal Activity** is the most powerful method of interacting with **Sugar**. However if you are not used to it then knowing some basics can help. The best strategy is to start using some simple commands. Don't attempt to do all your work from the command line straight away. Learn a few commands, use them and add to your understanding of what they can do over time.

Below are some basic commands that you could try starting with. Don't try and learn all of these at once. Just choose a few and practice them.

- ls
- cd
- mkdir
- pwd
- mv
- rm
- cp
- ping
- less
- date
- cat
- top

So, let's have a look at each. Feel free to experiment with these commands. Be a little careful as it is possible to do some damage to your computer if you are too casual. If there is a possibility one of the commands can accidentally create havoc then a note warns you about that.

LS

The **ls** c
ommand is the **List** command. You can use this to list the contents of any directory you are in. Try typing this command in a terminal window and see what you get. Now, one feature of commands is that you can add various parameters to them. This is quite a simple thing to do, and refines the way you use the command. Usually these parameters are added to the command by typing a " - " directly after the command and then the parameter names or abbreviations. For example if you type the following:

```
ls -l
```

Then you are *passing* the **l** parameter to the **ls** command. The **l** parameter is short for "long list". This format gives more information than just typing the **ls** command by itself... Try the two out and compare the difference.

You might well ask, "How do I know what the parameters are for each command?" This information can be found by using the "help" parameter for **ls**:

```
ls --help
```

For the **ls** command you should get familiar with the formats using **ls** by itself, as well as **ls -al**, and **ls -l**

CD

cd is the most common command used to navigate the file system on your computer.
cd stands for **Change Directory**. Try it out by typing **ls** to get a list of all the files and folders in

the directory you are currently in. Now try typing **cd** followed by the name of one of the files in the list, for example if there was a file called "me.txt" you could type:

```
cd me.txt
```

This gives an error! Why? Because you can't change to a directory if it is a file. It's good to try this so that you understand that you can't do any damage by making a mistake with **cd**. To change to a directory you type **cd** followed by the name of a directory you want to navigate to. If there was a directory called "src" you would type:

```
cd src
```

If that was successful then the terminal won't throw up an error. Try it with a real directory on your computer. If you fail it is because either you don't have permissions to enter the directory, you misspelled the directory name, or the directory simply doesn't exist.

MKDIR

This is the command you used to create a directory. It is short for **Make Directory**. To use this, simply type the name of the directory you want to create after the **mkdir** command as so:

```
mkdir bleep
```

The above command creates a directory called "bleep" in the current directory. If a directory with this name already exists, you get an error but fortunately the computer doesn't overwrite the existing directory.

PWD

If you get lost and don't know where you are in the file system you can always type **pwd** a nd it tells you where you are. **pwd** means **Present Working Directory** - this command gives you the location or *absolute path* of where you are. For example, if you are in your olpc home directory, the output of the **pwd** command is:

```
/home/olpc
```

Experiment with changing directories with **cd** then typing **pwd** to see where you are.

MV

This command is short for **Move**. It is as it sounds in that **mv** allows you to move files around on the computer. To use **mv** you must first type the command, followed by the file you want to move and then the place where you want to move the file to. For example, if you wanted to move a file "me.txt" from your current directory to the "/usr/bin" directory you would type the following:

```
mv me.txt /usr/bin
```

Note: You don't have to type the filename in the path name where you want to move the file unless you also wish to change the name of the file. If for example while you were moving "me.txt" you wanted to change the filename to "you.txt" you would type:

```
mv me.txt /usr/bin/you.txt
```

If you just wanted to rename the file and not move it you could use **mv** by typing this:

```
mv me.txt you.txt
```

Note that when you use **mv** you are *moving* the file not copying it. Be a bit careful because you can overwrite files accidentally. If for example there is already a file named "you.txt" in the example above, is it overwritten with the data from "me.txt".

RM

rm is a command you should be
very careful about using. **rm** is short for **Remove**, and is the command you use to delete a file or directory. To use this command type **rm** followed by the name of the file you wish to destroy for good. To remove a directory you can use the same command with the parameter -**R** like so:

```
rm -R directoryname
```

In this example "directoryname" is the name of the directory you wish to delete. You can also use **rmdir** for this. Be EXTREMELY careful when using these commands. If used unwisely they could be the end of your operating system.

CP

This is short for **Copy**. Use it like
mv, the only difference is that it leaves the original file where it was while also creating a copy.

PING

Not usually included in the top 10 commands you need to know but it's handy if you need to know if you are online. **ping** sends a request to any computer on the network. If that computer gets the request it responds. Type
ping followed by a URL that you know, for example it might be a good idea to try the following:

```
ping www.cnn.com
```

If that computer gets the request you get some information coming back through the terminal... this keeps scrolling so to stop it type **ctrl + c**.

If you get no response from **ping** then you are probably offline. However, some machines online don't answer ping requests for security and other reasons, so make sure you really know that the machine you are pinging does reply to these requests. Some Internet connections won't allow **ping**.

LESS

Use **less** if you want to control the overly verbose output of any command to the terminal. If for example, you are in a directory which contains 1000 files and you type
ls the output of the command won't fit nicely into your little terminal window so it goes scrolling past faster than is useful. To slow it down so you can read the output try this:

```
ls | less
```

If you used this in your 1000 file directory you get one page at a time of output and pressing the **spacebar** shows the next page. Pressing **q** quits less. Ok, so you might be wondering what the funny straight line is in the above command... well, this is known as the **pipe** command.

pipe allows you to combine commands together to control the kind of output you get, usually it's used to refine a command (which is what the command parameters also do). So, when you get really fluent with these commands you can write things that look more like equations but are really efficient ways of using standard commands... **pipe** will be central to enhancing your efficiency.

DATE

This command tells you the time and date as it is set on your computer.

CAT

cat displays the contents of files in your terminal window. You must type the name of the file you wish to display after **cat**. For example if you want to see the contents of the file "README" you would type:

```
cat README
```

If that file is too big to have its contents displayed in the terminal you might use it in combination with the **less** command like this:

```
cat README | less
```

TOP

The **top** command tells you which operations on your computer are using memory and your cpu. It's really only useful if you wish to see if there is an Activity or command slowing down your computer. The output of **top** continues running until you press **q**.

2. COMBINING COMMANDS

You can combine commands using the **Terminal Activity**. There are many situations where this is very useful. For example, if you were to use **ls** the output to the terminal would be a list of all the files and folders in the directory you are currently in. If this is a very long list then the names of the files and folders flash past so quickly you cannot read them. So we can combine the **ls** command with another command so that we see the list of names one "page" at a time. To do this combine the **ls** command with the **less** command like this:

```
ls | less
```

Now the list of file and folder names fills up the display area in the terminal but stops when the terminal window is full. It then waits for you to press the **spacebar** to display the next "page".

You combine commands by using the vertical line that you see in the example command. This must always go between the commands you wish to combine. This line is known as the "pipe" and in the above example we would say that we "pipe ls through less". That is to say, the output of the **ls** command is fed through the **less** command.

You can actually string many commands together in this way. However it's good to remember that this does not always work!

What do you think this example does?

```
ls | sort -r | cut -c1-3
```

You can try it in various combinations to see what happens:

```
ls | sort -r
ls | cut -c1-3
```

3. LICENSE

All chapters copyright of the authors (see below). Unless otherwise stated all chapters in this manual licensed with **GNU General Public License version 2**

This documentation is free documentation; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This documentation is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this documentation; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

AUTHORS

BasicCommands
© adam hyde 2008
Modifications:
Morgan Collett 2008
Tom Boyle 2008

CombiningCommands
© adam hyde 2008
Modifications:
Morgan Collett 2008
Tom Boyle 2008

Credits
© adam hyde 2006, 2007, 2008

DirectoryStructure
© adam hyde 2008
Modifications:
Anne Gentle 2008
Tom Boyle 2008

EnteringCommands
© adam hyde 2008
Modifications:
Tom Boyle 2008

HelpAboutCommands
© adam hyde 2008
Modifications:
Tom Boyle 2008

Introduction
© adam hyde 2006, 2007, 2008
Modifications:
Greg DeKoenigsberg 2008

KeyboardCommands
© adam hyde 2008
Modifications:
Tom Boyle 2008

StartingTheTerminal
© adam hyde 2008
Modifications:
Greg DeKoenigsberg 2008
Tom Boyle 2008

UsingVI
© adam hyde 2008
Modifications:
Morgan Collett 2008
Tom Boyle 2008

Free manuals for free software

GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

> **a)** You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
>
> **b)** You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
>
> **c)** If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that

there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

4. FILE STRUCTURE

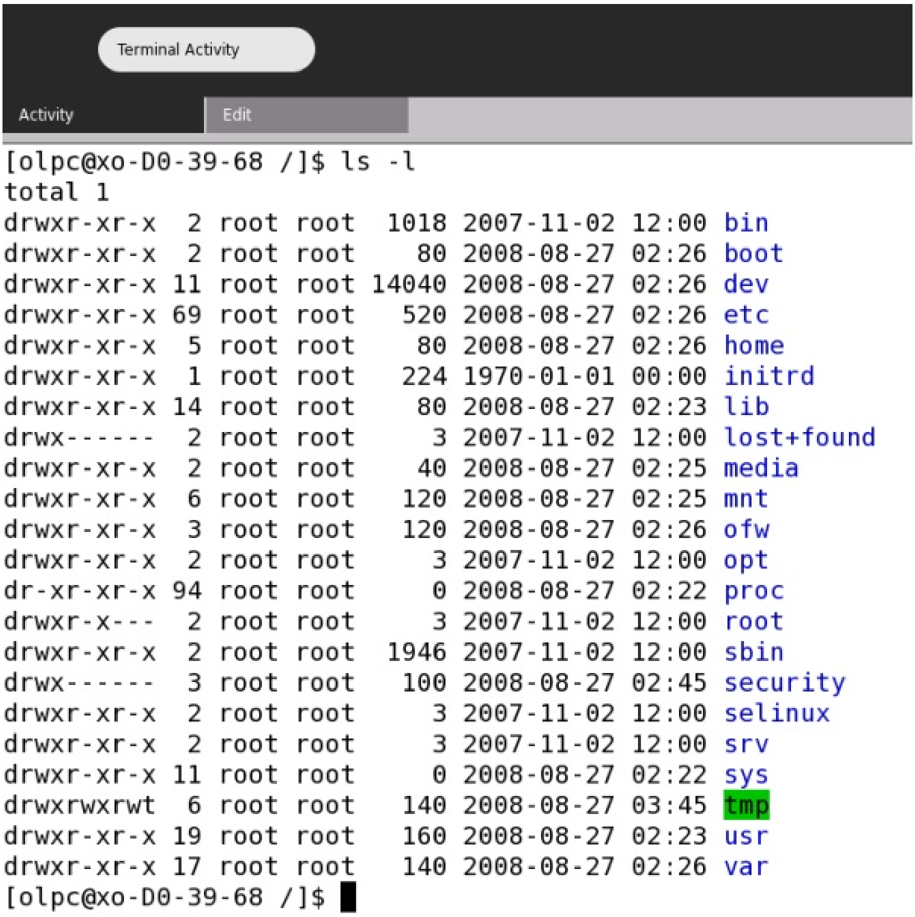
If you open your Terminal Activity and type the following (followed by pressing the **enter** key):

```
cd /
```

You are placed in the top directory of your computer. If you then type:

```
ls -l
```

You see something similar to this:



```
[olpc@xo-D0-39-68 /]$ ls -l
total 1
drwxr-xr-x  2 root root  1018 2007-11-02 12:00 bin
drwxr-xr-x  2 root root    80 2008-08-27 02:26 boot
drwxr-xr-x 11 root root 14040 2008-08-27 02:26 dev
drwxr-xr-x 69 root root   520 2008-08-27 02:26 etc
drwxr-xr-x  5 root root    80 2008-08-27 02:26 home
drwxr-xr-x  1 root root   224 1970-01-01 00:00 initrd
drwxr-xr-x 14 root root    80 2008-08-27 02:23 lib
drwx------  2 root root     3 2007-11-02 12:00 lost+found
drwxr-xr-x  2 root root    40 2008-08-27 02:25 media
drwxr-xr-x  6 root root   120 2008-08-27 02:25 mnt
drwxr-xr-x  3 root root   120 2008-08-27 02:26 ofw
drwxr-xr-x  2 root root     3 2007-11-02 12:00 opt
dr-xr-xr-x 94 root root     0 2008-08-27 02:22 proc
drwxr-x---  2 root root     3 2007-11-02 12:00 root
drwxr-xr-x  2 root root  1946 2007-11-02 12:00 sbin
drwx------  3 root root   100 2008-08-27 02:45 security
drwxr-xr-x  2 root root     3 2007-11-02 12:00 selinux
drwxr-xr-x  2 root root     3 2007-11-02 12:00 srv
drwxr-xr-x 11 root root     0 2008-08-27 02:22 sys
drwxrwxrwt  6 root root   140 2008-08-27 03:45 tmp
drwxr-xr-x 19 root root   160 2008-08-27 02:23 usr
drwxr-xr-x 17 root root   140 2008-08-27 02:26 var
[olpc@xo-D0-39-68 /]$ █
```

The above listing is what is known as the Sugar File Structure. Each name on the far right represents a directory, and each directory has a specific purpose. The **lib** directory, for example contains code libraries that the software on your system uses. For now you only need to be concerned with one directory: the **home** directory. This directory contains folders that have names corresponding to each user of the machine. If you log in as "adam" for example then you are logged into a directory in the "home" directory with the same name as your username ("adam" in this example).

The other important thing to know is that Sugar is mostly comprised of text files, so you can change almost every part of Sugar - how it looks and works - by just editing the appropriate text file.

TO DETERMINE THE IP ADDRESS FOR YOUR XO

Make sure you are connected to the Internet by going to the Neighborhood View and clicking the icon for an open network. Wait until the network icon stops blinking and the outline for that icon turns to a white circle.

1. Launch the Terminal Activity.
2. Type the following and press **enter**:

```
iwconfig
```

In the line next to inetaddr: , look for the IP address for your laptop, such as 192.168.0.2.

TO TRANSFER FILES TO AND FROM YOUR XO LAPTOP WIRELESSLY

To upload the file test.py from another computer to the computer running Sugar, such as the XO laptop so that it goes into /home/olpc, type this in a computer that has SCP installed:

```
scp FILE_NAME USER@IP:TO_DIRECTORY
```

For example:

```
scp test.py olpc@192.168.0.2:/home/olpc
```

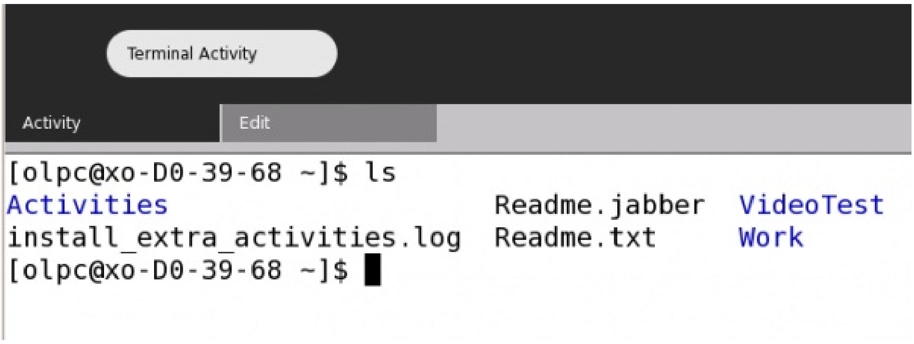
To download the file /home/olpc/xo_test.py from the XO to a local computer, simply reverse the file names and locations like this example:

```
scp olpc@192.168.0.2:/home/olpc/xo_test.py ./
```

5. ENTERING COMMANDS

Using the **Terminal Activity** is quite simple - you just need to type **commands** and press the **enter** key. The trick is knowing what to type and the basic structure of a command.

Let's look at entering a simple command into the terminal and then we will look at the structure of commands. Open the Terminal Activity and simply type the "ls" command and press **enter**:



```
[olpc@xo-D0-39-68 ~]$ ls
Activities                    Readme.jabber  VideoTest
install_extra_activities.log  Readme.txt     Work
[olpc@xo-D0-39-68 ~]$ █
```

You see something similar to the image above. "ls" is the command that lists files and directories. So the output of "ls" is a list of all the files and folders in the directory you are currently in.

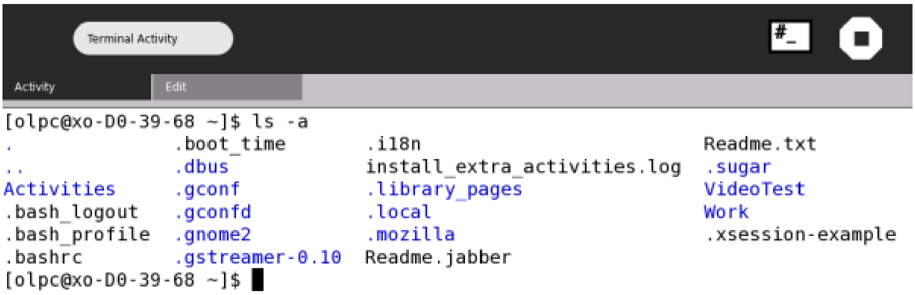
PARAMETERS

Next we move on to controlling commands a little bit more by asking them to do more specific actions. We do this by sending more specific requests to the commands - these are known as parameters and they are simply extra information that refine the command's actions.

The "ls" command has several of these parameters you can use. The "a" parameter, for example, means list *all* files and folders. To use this parameter we would type this:

```
ls -a
```

In the terminal you would then see something like this:



```
[olpc@xo-D0-39-68 ~]$ ls -a
.              .boot_time      .i18n                         Readme.txt
..             .dbus           install_extra_activities.log  .sugar
Activities     .gconf          .library_pages                VideoTest
.bash_logout   .gconfd         .local                        Work
.bash_profile  .gnome2         .mozilla                      .xsession-example
.bashrc        .gstreamer-0.10 Readme.jabber
[olpc@xo-D0-39-68 ~]$ █
```

```
ls --help
```

and the output would be information about the available parameters. Unfortunately there is often too much information to display and so it scrolls though the terminal window too quickly to read. If this is the case you need to combine the command with a "more" command like this:

```
ls --help | more
```

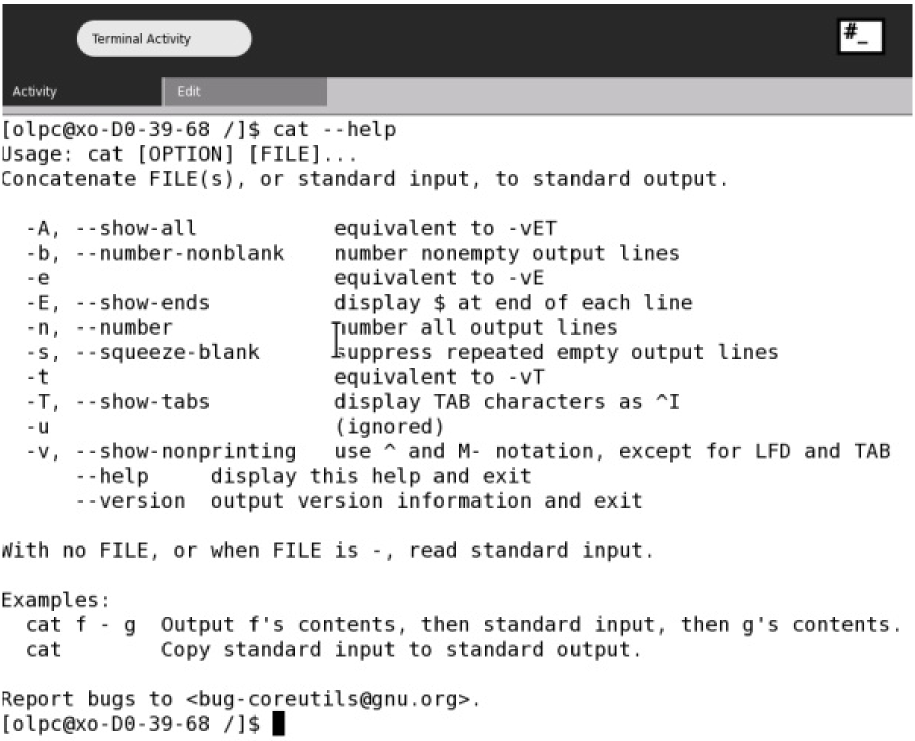
In the above example you would have as much information as can be displayed in the terminal window at one time. Then you press the **spacebar** and you see the next "page" of information etc.

6. GETTING HELP ON COMMANDS

To learn more about a command you can often type the command followed by either "-h" or "--help". If you wish to learn more about the **cat** command type this:

```
cat --help
```

This shows you the following:



```
[olpc@xo-D0-39-68 /]$ cat --help
Usage: cat [OPTION] [FILE]...
Concatenate FILE(s), or standard input, to standard output.

  -A, --show-all           equivalent to -vET
  -b, --number-nonblank    number nonempty output lines
  -e                       equivalent to -vE
  -E, --show-ends          display $ at end of each line
  -n, --number             number all output lines
  -s, --squeeze-blank      suppress repeated empty output lines
  -t                       equivalent to -vT
  -T, --show-tabs          display TAB characters as ^I
  -u                       (ignored)
  -v, --show-nonprinting   use ^ and M- notation, except for LFD and TAB
      --help     display this help and exit
      --version  output version information and exit

With no FILE, or when FILE is -, read standard input.

Examples:
  cat f - g  Output f's contents, then standard input, then g's contents.
  cat        Copy standard input to standard output.

Report bugs to <bug-coreutils@gnu.org>.
[olpc@xo-D0-39-68 /]$ █
```

The information displayed tells you how to use the **cat** command. At the top you can see that the help displays the basic pattern for using the command. This is noted by the "Usage: cat [OPTION] [FILE] ..." line.

This means that you should use the command by typing "cat" and then the parameters you wish to use (OPTION) and then the name of the file you wish **cat** to perform its operations on.

Then there is a short description of what the command does, in this case it says "Concatenate FILES(s), or standard input to standard output". Sounds a bit mysterious and I am afraid this kind of information is not always very easy to understand. In this case it means that the command can be used to either join ("concatenate") files or output a file referenced in the parameter ("standard input") to the terminal display ("standard output").

Then you have a list of parameters that you can use with the command and a short description of what they do. At the end are two examples of a "typical" use of **cat**.

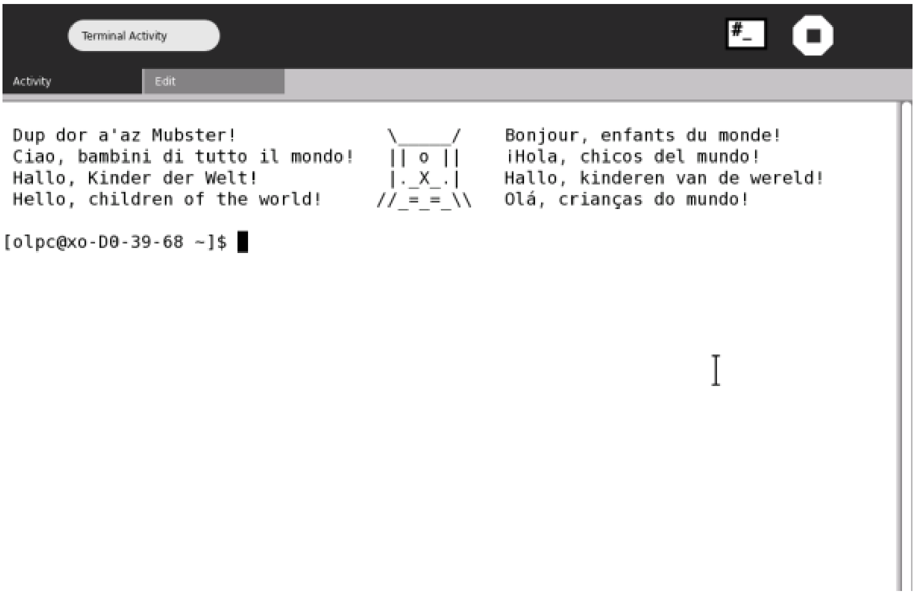
7. THE TERMINAL

Most modern computers have two very different ways that you can interact with them: the **Graphical User Interface** (**GUI**) and the **Command Line Interface** (**CLI**).

Most users are familiar with GUIs. When you use your mouse to point and click on things, you are using a GUI. It is very likely that you have never used the other type of interface: the CLI. With a CLI, you use text to control functions and software on your computer.

If you really want a fast way to work on your computer to change configurations, install software, or work remotely on another computer, then the command line is often the most efficient way to do it.

A type of CLI is the **Terminal**. In the Sugar world we call this the **Terminal Activity**.



```
 Dup dor a'az Mubster!              \_____/    Bonjour, enfants du monde!
 Ciao, bambini di tutto il mondo!   || o ||    ¡Hola, chicos del mundo!
 Hallo, Kinder der Welt!            |._X_.|    Hallo, kinderen van de wereld!
 Hello, children of the world!     //_=_=_\\   Olá, crianças do mundo!

[olpc@xo-D0-39-68 ~]$ █
```

8. KEYBOARD COMMANDS

There are several keyboard tricks that are good to know when using the Terminal Activity. These save time and work regardless of the commands you are using.

CTRL L

If you been doing a lot in your terminal and the terminal window is full of text you can press **ctrl** on the keyboard and **l** (lower case **L**) simultaneously and the terminal window is cleared of text leaving you with a nice clear terminal.

CTRL C

If for any reason you replace that a command you are using seems to be stuck or is taking too long you can use **ctrl** and **c** (pressed simultaneously) and this halts the command. This means your command will be stopped before it finishes so make sure you really want to do this before trying it.

TAB COMPLETE

If you want to save some time typing out long commands you can always use the **tab** key. This is called "Tab completion". If you have partly typed a command try pressing the **tab** key. If there are no other commands that start with the same sequence of letters then the complete command is automatically displayed in the terminal. Try for example typing half of the "help" command like this:

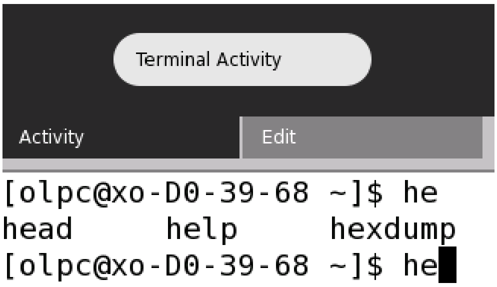
```
hel
```

followed by pressing **tab**. You see the terminal displays "help". This is because the terminal "knows" that there are no other commands starting with "hel" so it assists you by completing the full command name when you press **tab**.

If there is more than one command that starts with the letters you have typed then pressing **tab** once does nothing. Try, for example, typing:

```
he
```

followed by the **tab** key. Nothing happens. Now try the same thing but press **tab** and quickly follow it by pressing **tab** again. In this case you see all the other commands displayed that start with "he" like so:



```
[olpc@xo-D0-39-68 ~]$ he
head      help      hexdump
[olpc@xo-D0-39-68 ~]$ he
```

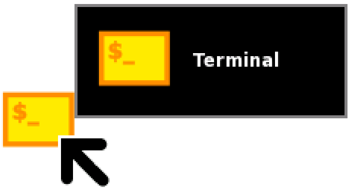
You can now see the other commands starting with these same letters and you can either choose one by typing out the whole command, or you can type some more letters of the command you want and finish it with "Tab complete".

9. STARTING THE TERMINAL

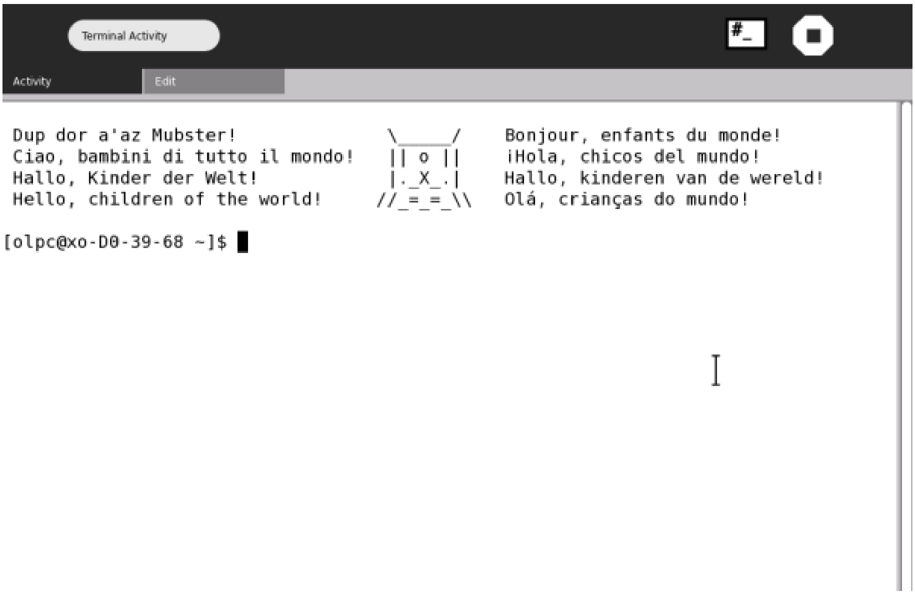
Starting the Terminal Activity is easy and quick. You can do it either from your Home View or from the List View.

STARTING FROM YOUR HOME VIEW

If you have added the **Terminal Activity** to your Home View, then just click it to start.



The **Terminal** then opens.



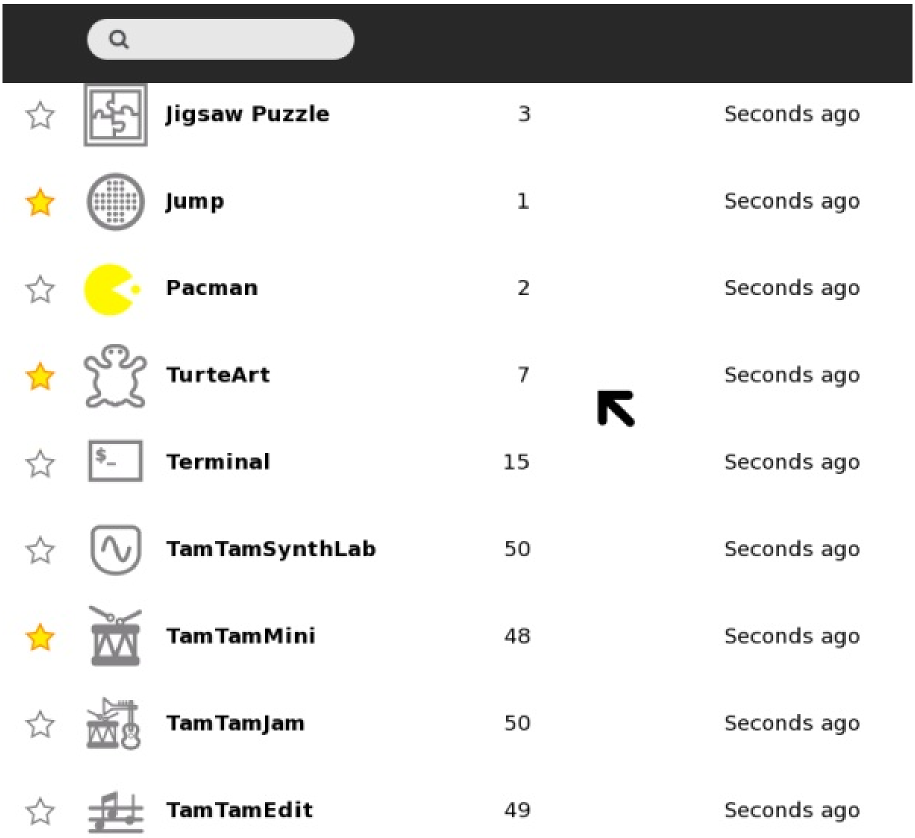
```
 Dup dor a'az Mubster!              \_____/    Bonjour, enfants du monde!
 Ciao, bambini di tutto il mondo!   || o ||    ¡Hola, chicos del mundo!
 Hallo, Kinder der Welt!            |._X_.|    Hallo, kinderen van de wereld!
 Hello, children of the world!     //_=_=_\\   Olá, crianças do mundo!

[olpc@xo-D0-39-68 ~]$ █
```

STARTING FROM LIST VIEW

If the Terminal is not added to your home page, you must start it from the **List View**. You can access the List View from the Home View by clicking its icon:

☰

This shows you a list of Activities:

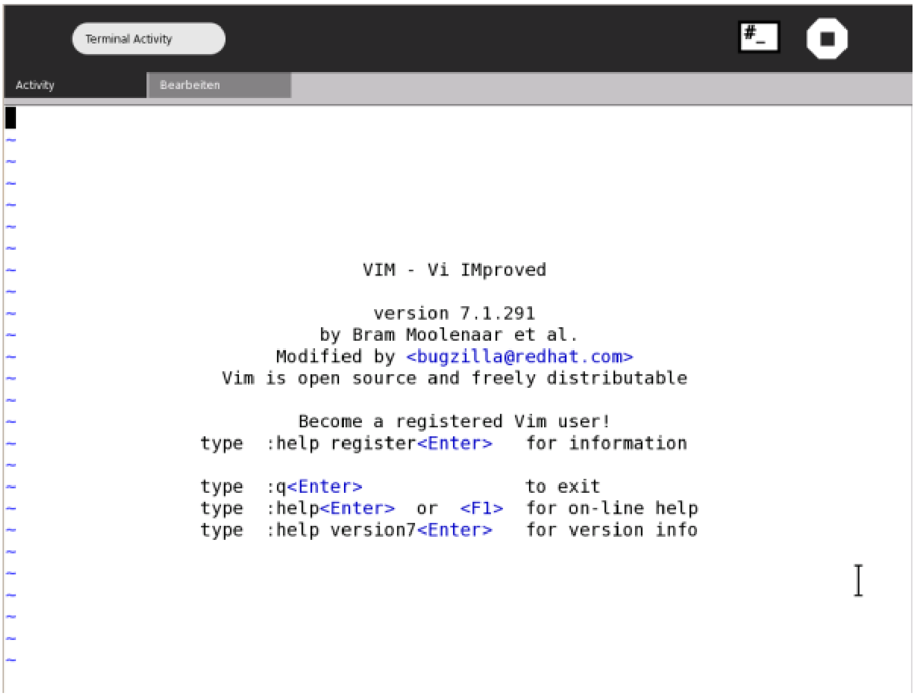


☆	Jigsaw Puzzle	3	Seconds ago
★	Jump	1	Seconds ago
☆	Pacman	2	Seconds ago
★	TurteArt	7	Seconds ago
☆	Terminal	15	Seconds ago
☆	TamTamSynthLab	50	Seconds ago
★	TamTamMini	48	Seconds ago
☆	TamTamJam	50	Seconds ago
☆	TamTamEdit	49	Seconds ago

If you scroll down the list (use the scroll bar on the right) you see the Terminal Activity listed. The colored stars you see mark the Activities that are on your Home View. The colorless stars are Activities that are not on the Home View. You can now either click the star next to the Terminal Activity to add it to your home page, or you can simply click the icon and the Terminal Activity starts.

10. VI

Sugar has a built in text editor that you can use with the **Terminal Activity**. This editor is called **vi** and is used in many other types of **operating systems** such as **Linux**. Let's have a quick look at **vi**: type **vi** in the terminal and you see something like this:



```
                    VIM - Vi IMproved

                     version 7.1.291
                 by Bram Moolenaar et al.
             Modified by <bugzilla@redhat.com>
        Vim is open source and freely distributable

               Become a registered Vim user!
      type  :help register<Enter>   for information

      type  :q<Enter>               to exit
      type  :help<Enter>  or  <F1>  for on-line help
      type  :help version7<Enter>   for version info
```

This is vim running *inside* the terminal. You can use vim as a text editor so that you don't need to open any other Activities to read or write most documents.

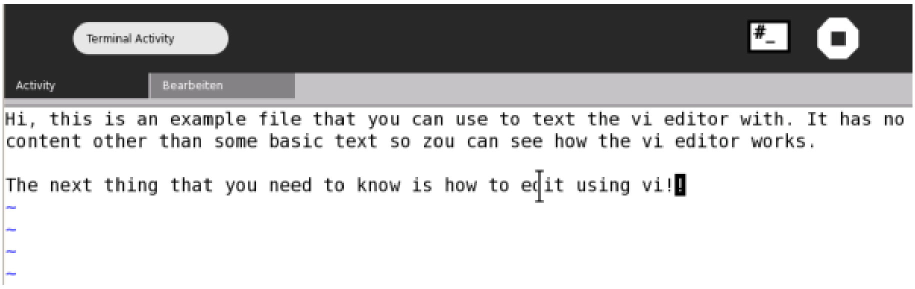
OPEN A TEXT FILE

To open a file with **vi** it is best if you type the name of the file you wish to open after the **vi** command, so that vim opens with the file already loaded. For example if you wanted to read the text file called "MyExample.txt" in the same directory you are currently working in then just type:

```
vi MyExample.txt
```

Note: If you the type the above and the file "MyExample.txt" does not exist then vi opens a new (blank) document.

If we assume there is a file called "MyExample.txt" then the above command opens vi with the file loaded as so:



```
Hi, this is an example file that you can use to text the vi editor with. It has no
content other than some basic text so zou can see how the vi editor works.

The next thing that you need to know is how to edit using vi!
~
~
~
~
```

SIMPLE COMMANDS

Now, vi may be quite a bit different than any text editor you have used before, so perhaps some explanation is needed. First, since vi works on the command line there are no menus to click to make things happen. Instead you must use the keyboard to type commands that vi understands. There are many commands you can use to work on the file and most of them are executed by just typing a single letter, or they are in the format:

```
: command
```

Where "command" is the name of the command you wish to use. The commands are all designated by shortcuts. An "i" , for example, is short for "insert". The following is a table of vi commands you should know:

command	action
i (only used in read-only mode)	insert text
:w (only used in read-only mode)	write changes to file
u (only used in read-only mode)	undo changes
:q (only used in read-only mode)	quit vim

vi always opens a file initially in read-only mode. This means that when you first open the file you cannot change the file. It is in this mode that you type the commands. At anytime you can press the **esc** (escape) key to return to read-only mode.

Let's look at some examples. First open a file as you did in the above example:

```
vi MyExample.txt
```

This opens the "MyExample.txt" file as explained above, or creates a new (blank) file if it did not already exist.

You can scroll up and down the file using the up and down arrows on your keyboard.

To insert new content or change the existing content of the file in vi you need to type:

```
i
```

This puts you in the insertion mode and now anything you type appears in the document itself. When you have finished making the changes you may wish to save the file. You would then press the **esc** key followed by:

```
:w
```

This writes the file with the new changes. You then need to quit from vi so you press the **esc** key followed by:

```
:q
```

Now replace a file and experiment. If you haven't used something like vim before then it might take some getting used to, so spend some time working out for yourself how vi works before you really need to use it.

If you want to get out of vi without saving your changes, use:

```
:q!
```

If the commands you are typing are going into the document instead of having an effect, remember to press **esc** first!

If you make a mistake when editing a document, press **esc** and then **u** for undo.