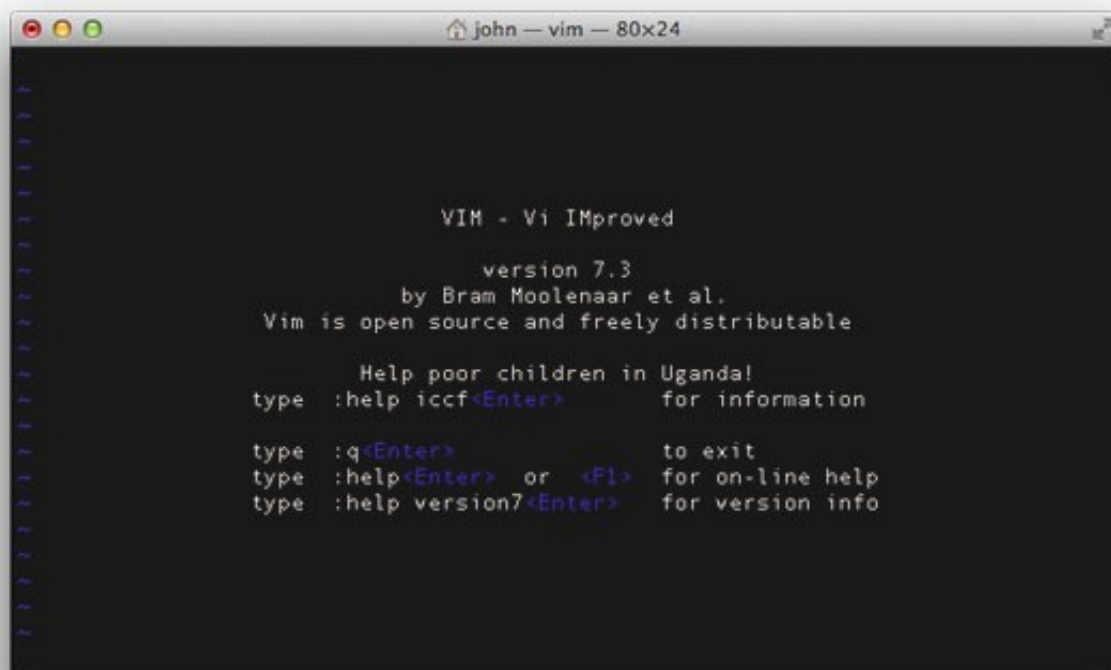


MENU

VIM 101: a quick-and-dirty guide to our favorite free file editor

By John Browning posted Jul 10th, 2012 at 3:00 PM

0



In the world of text editors, there's a plethora of options out there. If you've ever Googled "how to edit HTML sites" or some such, you know what we mean. Allow us, then, to introduce you to VIM, a free website editor that offers many of the same features as Adobe Dreamweaver, and runs on just about every desktop platform. Specifically, it comes by default on the vast majority of Linux distributions, OS X and commercial Unix systems. (It's available to install on Windows, too.) And did we mention it's free? That command line UI isn't necessarily self-explanatory, though, so join us after the break for a quick crash course to help you get started.

GETTING STARTED

If you're running OS X or Linux, start out by opening a terminal. Now type "vim" et voilà: you're using VIM and you didn't even install it. Using Windows? Head over [here](#) and grab the binaries for Windows. Double-click the installer and you'll have VIM on Windows in no time.

What's VIM?

VIM is a file editor that supports multiple formats. It's been around for two decades and is used by millions of programmers, hackers and system administrators. For starters, let's get the obvious question answered: "Why would I want to use VIM instead of ... ?" Put it this way: if you're a web programmer and love [Dreamweaver](#) and it does everything you need, great. However it'd behoove you to give VIM a shot, because it takes many features offered by Dreamweaver, a \$400 product, and offers them for free. As it happens, it's also easy to use once you get the hang of it. Yes, that command line interface might seem complex at first blush, but as with any complex piece of software, you can learn to use it and use it well.

What am I looking at?

VIM is sometimes referred to as a dual-mode text editor. To put it plainly, VIM has two modes in which you can manipulate files: Edit and GUI mode. Edit mode is the state in which the keys you bang on are actually inserted into your document, whereas the GUI layout allows you to navigate through the document, search and replace text, copy and paste, etc. By default when you open a file, you're placed in GUI mode.

BASIC COMMANDS

To navigate VIM you use a cursor, but instead of dragging and clicking your mouse, you'll have to learn to use the arrow buttons as a kind of crude D-pad. Or, depending on where your hands are resting on the keyboard, you can use the "H, K, J and L" keys to move around. Here's how it breaks down: "H" moves left; "K" moves up; "L" moves right; "J" moves down. You can, of course, use the arrow keys as well. Got it? Good.

EDITING DOCUMENTS

As we said, VIM launches into the GUI mode by default. To switch to edit mode (also called "insert mode") you need to give VIM a command to tell it to switch modes. There are two ways to do this:

- Press "i" to begin inserting text at the current cursor position
- Press "a" to begin inserting after the current cursor position.

Getting out of editing mode is a bit more intuitive: just hit "escape."

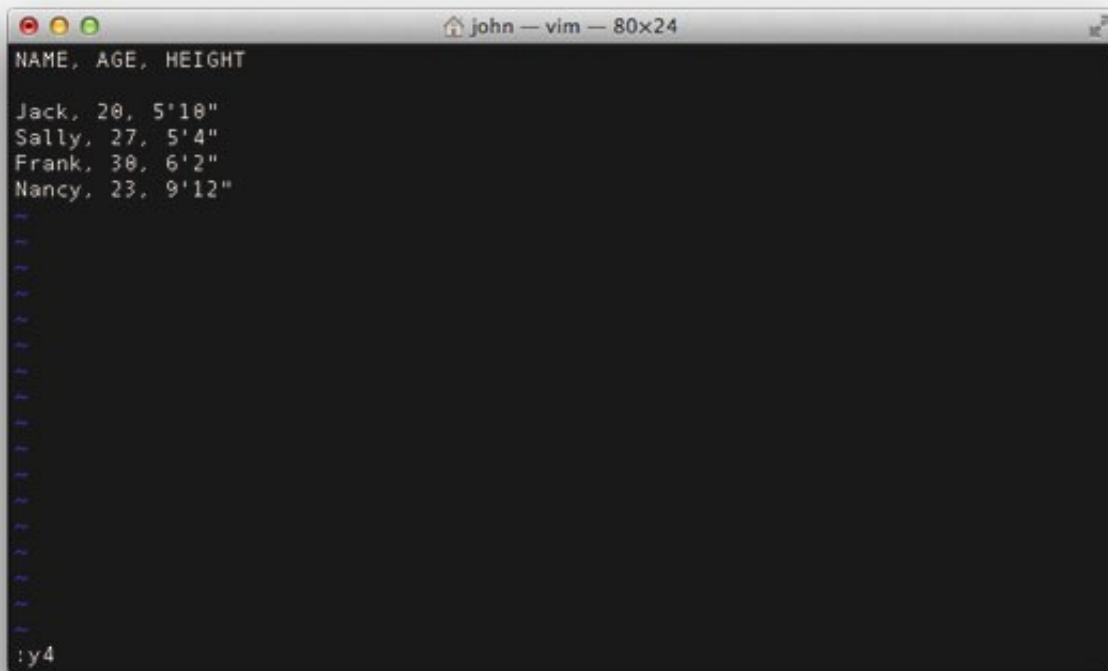
Got a handle on those three? Here are some other commands in GUI mode. And remember, these letters are case-sensitive, so take note of those stray capital letters:

- "Y" copies a line of text to the buffer.
- "P" pastes it to the cursor's current position.
- "dd" will delete the whole line of text. This will also effectively "cut" a line of text as well. When you delete a line, it's placed in the buffer.
- "yy" copies a whole line of text.

Typing commands in GUI mode

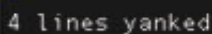
So you've mastered basic shortcuts in VIM. But what if you want to give the program a more specific command -- like, say, copying four lines of text? Here's what you need to know.

For starters, move the cursor to the beginning of where you'd like to copy. (Remember that you can use the arrow keys or the "H, K, J and L" buttons.) Press the ":" button to bring up the command line in the bottom of your VIM window. Type "y4" and press "enter."



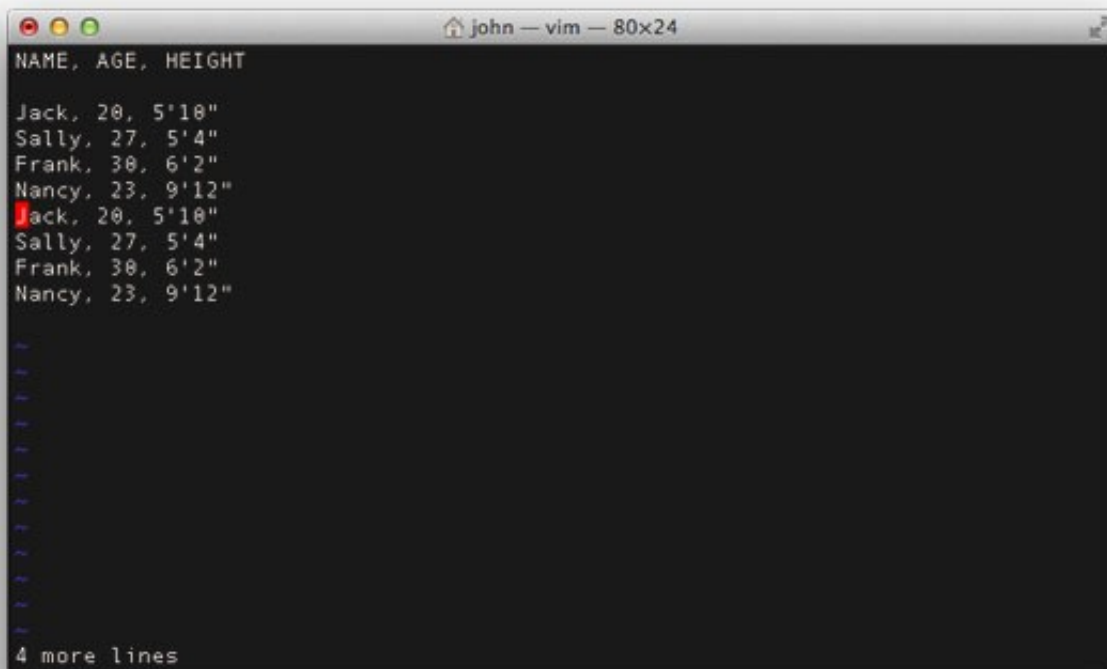
```
john — vim — 80x24
NAME, AGE, HEIGHT
Jack, 20, 5'10"
Sally, 27, 5'4"
Frank, 30, 6'2"
Nancy, 23, 9'12"
:
:
:
:
:
:
:
:
:
:
:
:y4
```

VIM will tell you "4 lines yanked."



```
4 lines yanked
```

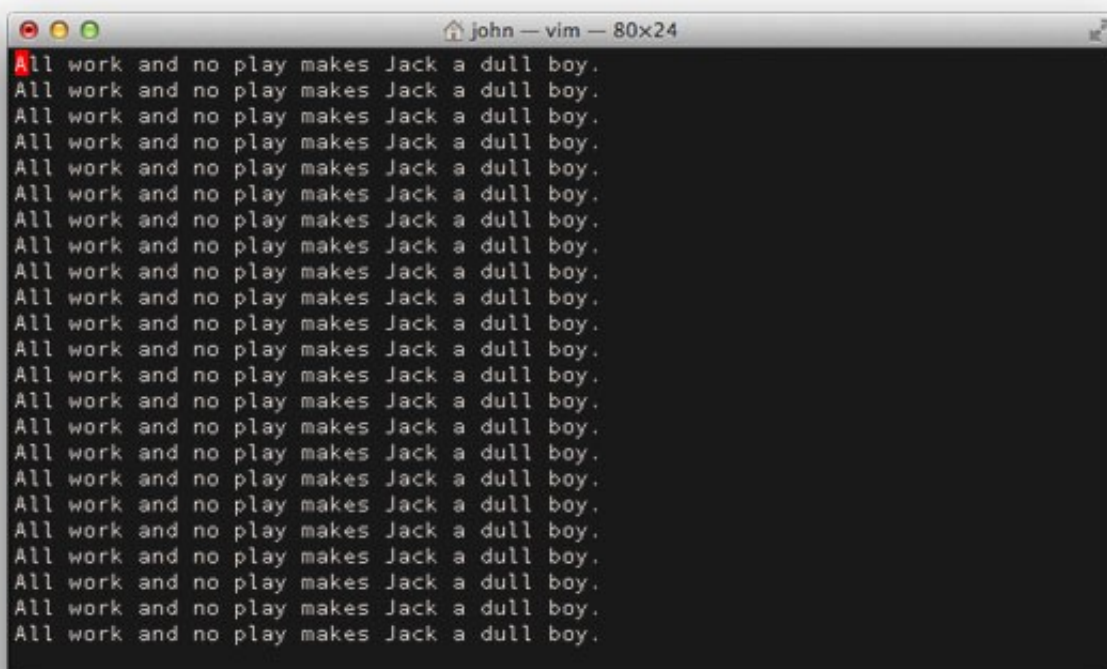
Move the cursor to where you want to begin your paste, then press "P". Basically, that breaks down as "yN" where "N" equals the number of lines you wish to copy. You can do the same thing with the delete command, "d".



Search and replace

Moving on, if you want to search and replace from GUI mode, start off by hitting the ":" button once again to bring up that familiar prompt at the bottom of the editor. Then type "%s/***text to search for/text to replace***/g".

Let's replace all instances of "Jack" with "John" in the example below.



```
Type ":%s/Jack/John/g".
```

```
All work and no play makes Jack a dull boy.  
:%s/Jack/John/g
```

And that's it: all instances of "Jack" are now replaced with "John".

Undo

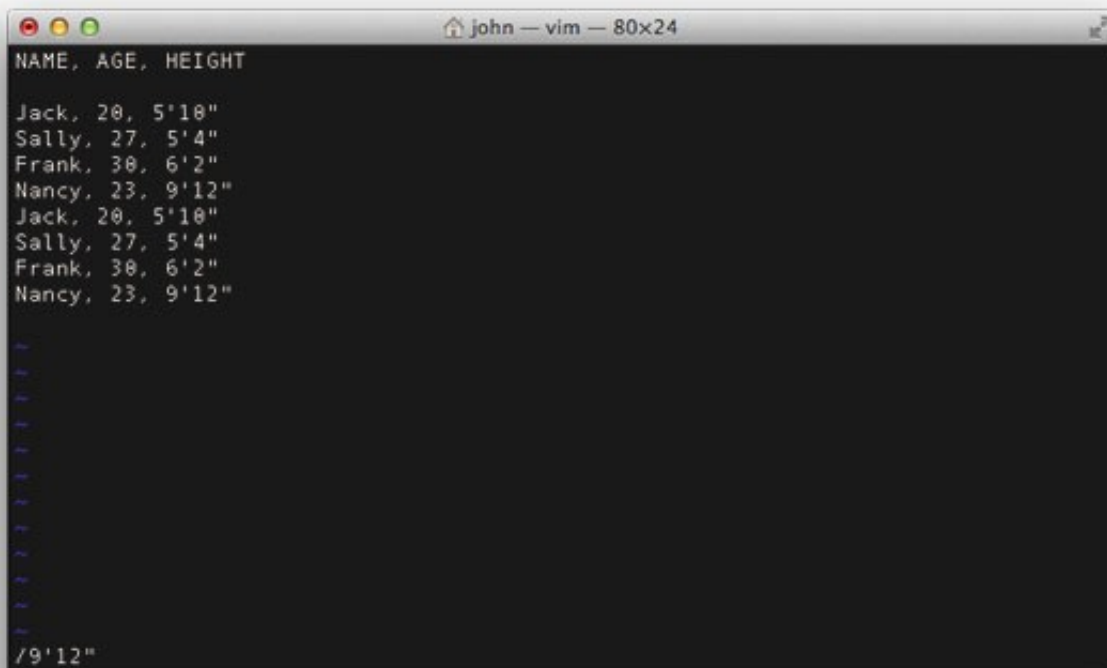
Let's scrap what we just did and search and replace on one line. You naturally need an "undo" function in a modern-day editor. Don't worry, VIM has one. To reverse some edit you made, make sure you're in GUI mode and hit "u".

Now, say you only want to replace something on a specific line. You can do that quickly without having to hold down an arrow key to move your cursor. Just bring up the command (using the ":" key, of course) and type the line number you wish to go to. For example: ":15".

Now just search and replace without the percent sign. That "%" symbol essentially means you want to search the whole document; when you don't use it, you are only searching the line your cursor is currently on.

```
All work and no play makes Jack a dull boy.  
:s/Jack/John/g
```

What if you don't know the exact line number what you want to search for? You can also just search for a specific term. To trawl the document, make sure you're in GUI mode and type a forward slash. For example, let's search for someone who is 9 feet 6 inches tall (9'6") in our data file. To do this, just type `"/9'6"` and press Enter.

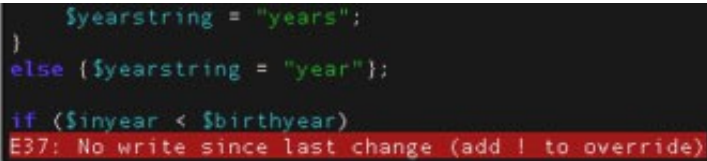


This will automatically move your cursor to the first instance of "9'6"" that it finds. Press Enter again and it'll move to the next instance.

SAVE YOUR WORK

To save your work, the most important part of editing files, you'll need to give VIM the command to do it. From good ol' GUI mode type `":w"` and press enter. File saved. Wanna save and quit VIM at the same time? Typing `":wq"` will do the trick.

What if you want to quit without saving? Type `":q!"` And remember to include that exclamation point -- it's required because VIM will let you know you've made changes to the file, and doesn't want you to potentially lose any work. By using the exclamation point you're insisting to VIM that you know what you're doing.

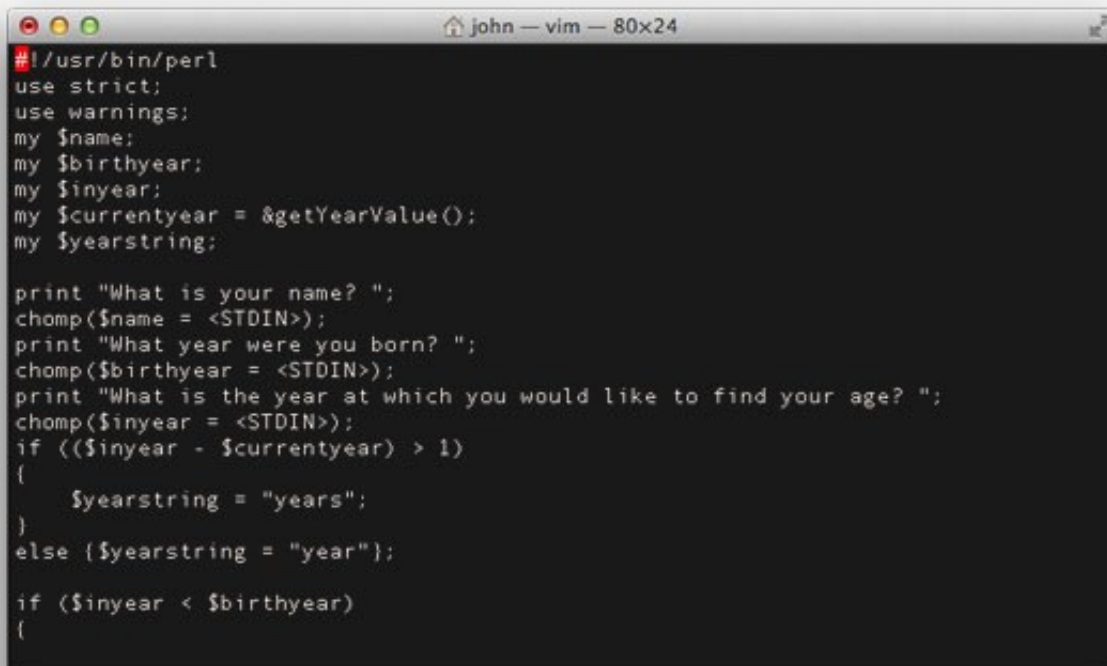


```
$yearstring = "years";
}
else {$yearstring = "year"};
if ($inyear < $birthyear)
E37: No write since last change (add ! to override)
```

SYNTAX HIGHLIGHTING

You aren't always going to be editing simple text files -- maybe, just maybe you'll whip up that family website everyone's been asking for. Naturally, you're going to want a look that's a bit easier on the eyes. VIM fully supports colored syntax highlighting for multiple programming languages. There are even different color themes.

In the below example we're going to be looking at a Perl script.

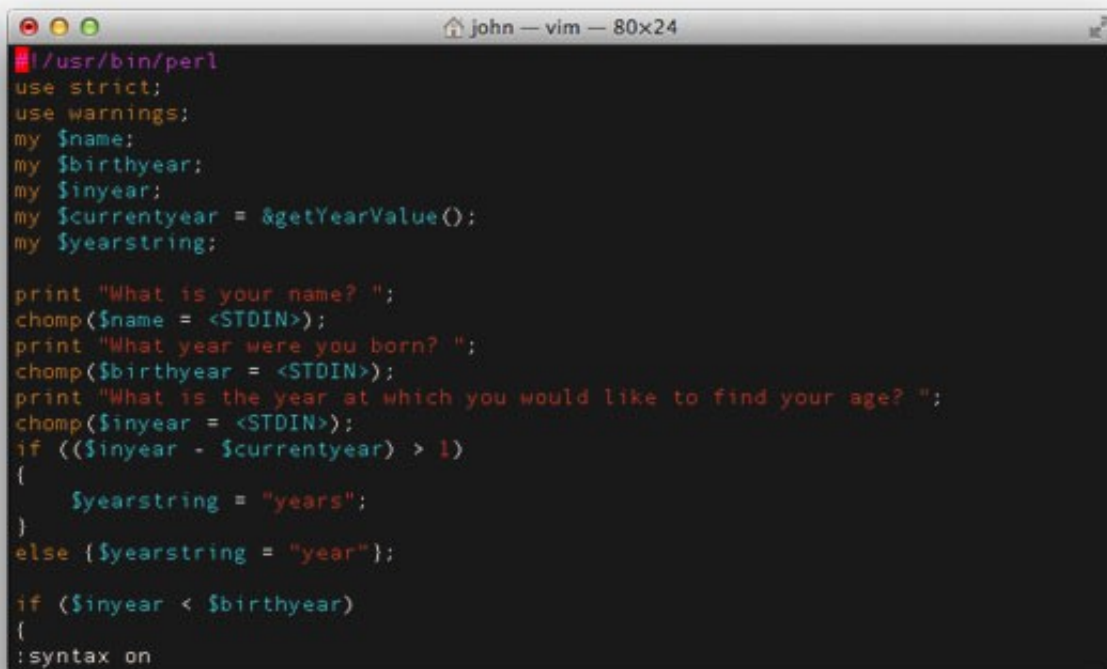


```
#!/usr/bin/perl
use strict;
use warnings;
my $name;
my $birthyear;
my $inyear;
my $currentyear = &getYearValue();
my $yearstring;

print "What is your name? ";
chomp($name = <STDIN>);
print "What year were you born? ";
chomp($birthyear = <STDIN>);
print "What is the year at which you would like to find your age? ";
chomp($inyear = <STDIN>);
if (($inyear - $currentyear) > 1)
{
    $yearstring = "years";
}
else {$yearstring = "year"};

if ($inyear < $birthyear)
{
```

That plain text is hard to see, especially if you've been staring at it for a few hours. From GUI mode, type `":syntax on"`.

A screenshot of a Vim editor window titled "john — vim — 80x24". The window displays a Perl script with syntax highlighting. The code includes variable declarations, input prompts, and conditional logic for calculating age. The syntax is highlighted with various colors: keywords in blue, strings in red, and comments in green.

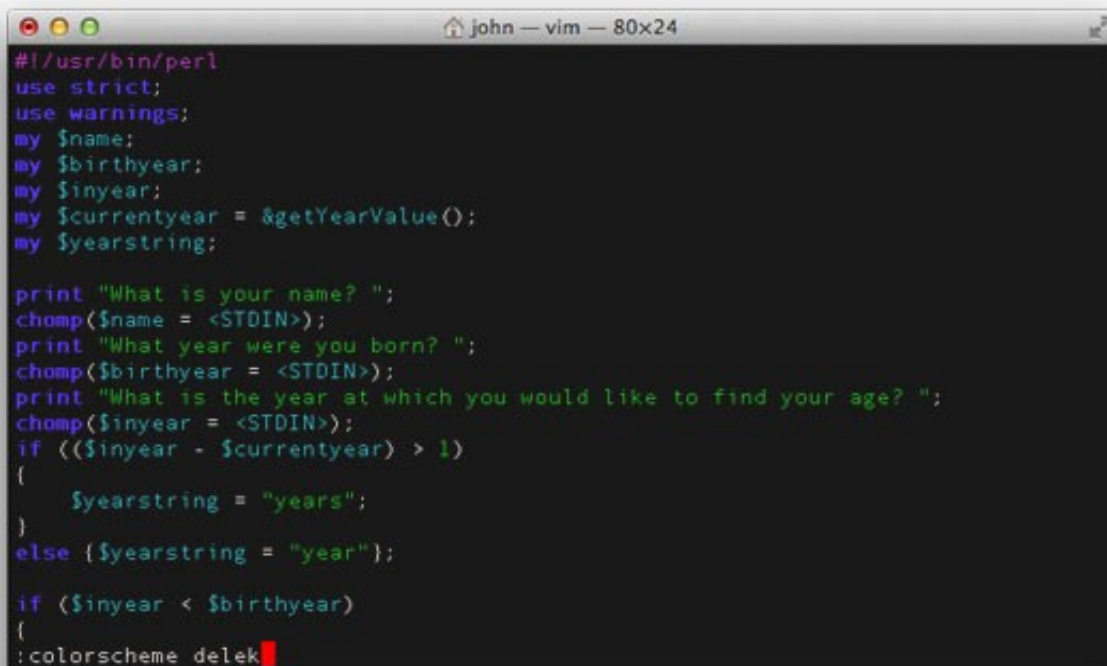
```
#!/usr/bin/perl
use strict;
use warnings;
my $name;
my $birthyear;
my $inyear;
my $currentyear = &getYearValue();
my $yearstring;

print "What is your name? ";
chomp($name = <STDIN>);
print "What year were you born? ";
chomp($birthyear = <STDIN>);
print "What is the year at which you would like to find your age? ";
chomp($inyear = <STDIN>);
if (($inyear - $currentyear) > 1)
{
    $yearstring = "years";
}
else {$yearstring = "year"};

if ($inyear < $birthyear)
{
:syntax on
```

Much better.

Don't like the current color palette? We can change that as well: VIM comes loaded with a handful of different schemes. To pick one from GUI mode, type `:colorscheme` and press the Tab key repeatedly to cycle through the available schemes. Press Enter when you see one that strikes your fancy.

A screenshot of the same Vim editor window, showing the same Perl script. The color scheme has been changed to 'delek', which uses a different set of colors for syntax highlighting. The command `:colorscheme delek` is visible at the bottom of the editor.

```
#!/usr/bin/perl
use strict;
use warnings;
my $name;
my $birthyear;
my $inyear;
my $currentyear = &getYearValue();
my $yearstring;

print "What is your name? ";
chomp($name = <STDIN>);
print "What year were you born? ";
chomp($birthyear = <STDIN>);
print "What is the year at which you would like to find your age? ";
chomp($inyear = <STDIN>);
if (($inyear - $currentyear) > 1)
{
    $yearstring = "years";
}
else {$yearstring = "year"};


if ($inyear < $birthyear)
{
:colorscheme delek
```

EDITING FILES REMOTELY

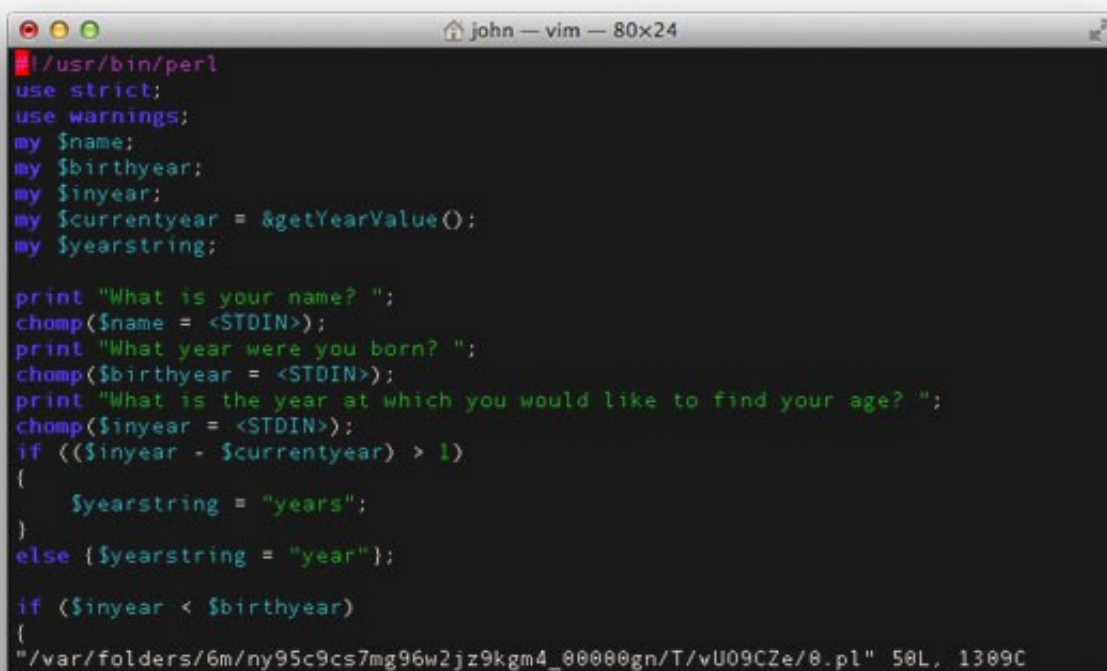
If you're working with a remote web server and need to edit a file over FTP, VIM supports this. You can do this via "vim ftp:///". For example: "vim ftp://172.16.15.73//Users/john/test.pl". (Note the double "/" in the example.)



VIM will then prompt you for the username and password for the remote file server.



Magically, you're editing the remote file.



HOW TO SAVE YOUR SETTINGS

So you've taken the time to pick your color scheme and turn syntax highlighting on. You've saved your file, closed VIM and taken a coffee break. Well, when you come back and load up VIM again, your highlighting is gone and you don't have your favorite color scheme loaded.

You'll need to save your settings in the VIM config file (`_vimrc` on Windows; `.vimrc` on all other platforms). Place this file in your standard home directory. To find out what your home directory is from within VIM, do this:

- From GUI mode type `":echo $home"` and press Enter. VIM will then show the full directory path of where you should place your vimrc file.

To save your settings in your vimrc file, simply add the commands you used to set up your VIM environment in that file. For example, to keep syntax highlighting on by default and always use the

color scheme "delek," add this to your vimrc file:

```
:syntax on  
:colorscheme delek
```

Work complete.

WRAP-UP

This quick-and-dirty guide should get you up and running with VIM. There's much more to this powerful editor, though. We recommend taking the full VIM tutorial by typing ":help" and reading through the VIM guide. If you have even more time on your hands, there are also tons of scripts (read: plugins) available for VIM in case the stock deployment doesn't do that odd thing you need. You can browse all 4,000-plus [here](#). Happy editing!

SOURCE: [VIM](#)

TAGS: [dreamweaver](#), [editor](#), [howto](#), [HTML editor](#), [html editors](#), [HtmlEditor](#), [HtmlEditors](#), [text editor](#), [text editors](#), [TextEditor](#), [TextEditors](#), [VIM](#)

175

432

1

0

0

0

PREVIOUS

[RIM CEO hints at future BBM video chat feature](#)

NEXT

[Apple granted patent for accelerometer-aided theft-detection system](#)

DISQUS 

[Engadget's comments powered by DISQUS](#)

[about us](#)

[subscribe via rss](#)

[like engadget on facebook](#)

[@engadget on twitter](#)

[Español](#) [繁體中文](#) [简体中文](#) [日本版](#) [Deutschland](#)

© 2013 AOL Inc.
All rights reserved.

[Reprints and Permissions](#)

[Privacy Policy](#)

[Terms of Use](#)

[Trademarks](#)

[AOL A-Z Help](#)

[Advertise with Us](#)