VIM

1 Instalação

Clique aqui e faça o downloaddo script deinstalação do Vim.

1.1 A instalação do vim é fácil pois ele faz parte do repositório do Debian/Ubuntu e da maioria das distros existentes.

```
sudo apt install vim -y
```

2 Manual do VIM

2.1 Navegação

- h → Use o comando h para navegar para o lado esquerdo do console do Vim.Isso é análogo à seta esquerda do seu teclado.
- l → Os comandos l o ajudam a navegar para a direita e substituem a tecla de seta para a direita.
- k → Use o comando k em minúsculas para mover para cima.Semelhante à tecla de seta para cima.
- j → O comando j leva você para baixo e é um substituto para a tecla de seta para baixo.
- H → Este comando leva o cursor na região superior da tela.
- M → Colocará o cursor no meio da tela do Vim.
- L → A letra maiúscula coloca o cursor no final da tela.
- 0 ou [HOME] → Digitar 0 ou o botão [HOME] levará você ao início da linha.
- ^ → Este é um dos comandos do Vim mais usados para obter o primeiro caractere não em branco de uma linha.
- $ → Este comando leva o cursor no final da linha atual.
- b → O comando b em minúscula permite retornar por tokens.
- w → Este comando ajuda você a avançar por tokens.
- B → A variante maiúscula de b permite retornar por palavras.
- W → Este comando permite avançar por palavras.
- ctrl+u → Este comando Vim é basicamente um comando Page Up.No entanto, ele move a tela pela metade e mantém a posição atual do cursor.
- ctrl+d → A variante Page Down do comando acima.Funciona da mesma maneira.
- G → Prefixe o comando G com um número para pular diretamente para um número de linha específico. Suponha que, se você tiver a numeração de linha ativada e estiver escrevendo um script de shell, poderá ir para a 10ª linha simplesmente digitando 10G.
- → Dado, # é o número de uma linha específica, digitar este comando o levará diretamente a essa linha.É um dos comandos mais usados do Vim para pular de uma linha para outra.

- “ → Este comando (dois ticks de volta) leva você aonde você estava.
-) → Você pode acessar no início da frase a seguir com este comando.
- (→ Ir para no início da frase anterior.
- } → Usado para ir para o início do próximo bloco de texto.
- { → Pega o cursor no início do bloco de texto anterior.Comandos do Vim para editar textos

2.2 Modo Comando

- i → Pressionar i no console leva você ao modo de inserção.Agora você pode começar a digitar seus textos no Vim.O texto digitado aparecerá na frente da tecla do cursor.
- a → Está entre outros comandos do Vim comumente usados que você pode usar para inserir texto.Os textos serão anexados logo após o cursor.
- I → Deseja colocar seus textos no início da linha atual?O comando l permite inserir texto diretamente no local desejado.
- A → Este é um dos meus comandos favoritos do Vim para acrescentar texto no final da minha linha atual.Ele também se enquadra nos comandos do Vim usados para inserir texto.
- o → Pressionar a letra minúscula o cria uma nova linha logo após a linha atual em que você está.
- O → A variante maiúscula cria a nova linha antes da linha atual.
- gf → Você pode usar este pequeno comando para abrir seus arquivos sob o cursor.
- gi → Um dos meus comandos favoritos do Vim, gi reinicia o modo de inserção no último ponto de inserção que você usou.
- wq → Digitar longas seqüências de texto não importa se você não pode salvá-las para uso futuro.Digite o comando wq no modo de comando para salvar seu arquivo e sair do Vim.
- q! → Muitas vezes, você se encontra em uma posição em que não está satisfeito com os textos digitados e gostaria de fechar o editor sem salvar seu conteúdo.Digitando q!no modo de comando, você pode fazer exatamente isso.Comandos do Vim para mover-se pelo console
- yy → O comando yy permite copiar uma linha inteira.É algo que você costuma empregar durante seu tempo com o Vim.
- yw → Este é um dos comandos do Vim mais amplamente usados para copiar uma palavra no editor.
- y$ → Um dos meus comandos favoritos do Vim de todos os tempos, oferece aos usuários a capacidade de copiar textos da posição atual do cursor para o final de uma linha.
- v → O comando v pode ser usado para destacar um único caractere de cada vez em combinação com as teclas de movimento (setas / hjkl).
- V → Quase o mesmo que o anterior, mas em vez de uma palavra, este comando destacará uma linha inteira.
- p → Você pode colar o conteúdo da sua área de transferência no registro sem nome com este pequeno e prático comando.
- d → Este é um dos comandos do Vim mais usados para excluir textos destacados.
- dd → Deseja excluir uma linha inteira com um único comando do Vim?O comando dd é criado especialmente para essa finalidade.
- dw → Você pode excluir uma única palavra rapidamente com o comando dw.

- D → Um dos comandos Vim mais poderosos de todos os tempos, o comando D exclui tudo da localização atual do cursor até o final da linha.
- d0 → Este comando é usado para excluir tudo da posição atual do cursor até o início da linha.
- dgg → Você pode usar este comando para excluir tudo da posição atual do seu cursor até o início do arquivo.
- dG → Este comando apaga tudo, desde a posição atual do seu cursor até o final do arquivo.
- x → Use o comando x sempre que precisar excluir um único caractere.
- u → O comando u está entre os comandos Vim mais amplamente utilizados por muitos para desfazer a última operação.Combinando com um postfix, os usuários podem desfazer várias ações.Então, você desfaz o último número de ações.
- ctrl+r → Use o comando acima para refazer a última operação de desfazer.
- . → O comando ponto (.) É um daqueles comandos úteis do Vim que diminuem significativamente sua carga de trabalho, repetindo a última ação sempre que você precisar dessa funcionalidade.
- cc → Você pode usar o comando cc para alterar as linhas, limpando e entrando no modo de inserção ao mesmo tempo.O cursor é colocado no nível atual de recuo.Comandos úteis do Vim para substituir textos

2.3 Modo Visual

- r → O comando r é uma ferramenta bastante útil para alterar um único caractere.Siga-o com[caractere] e ele mudará o caractere atual sob o cursor com[caractere].
- R → O R maiúsculo abre o modo de inserção, mas em vez de inserir textos, você pode substituí-los por este comando.
- ~ → O comando tilda (~) é bastante útil quando você precisa alterar a caixa de um caractere no seu documento.Siga-o com um número para inverter tantos caracteres.
- t[caractere] → Digite t[caractere] para selecionar até, mas não incluindo, o próximo [caractere] em uma linha específica.
- f[caractere] → Pressione f[caractere] para selecionar até e incluindo o próximo [caractere] em uma linha.
- i[caractere] → Deseja selecionar tudo entre um parênteses ou outro caractere exclusivo?Digite i[caractere] para selecionar tudo que fica entre dois[caractere] consecutivos.
- a[caractere] → Este comando é idêntico ao anterior, mas inclui o[caractere] nas duas extremidades do texto.Comandos Vim mais usados para pesquisar em um documento

2.4 Pesquisa e substituição

- / → O comando barra é o comando mais usado para pesquisar em grandes arquivos de texto no Vim.Basta digitar / e seguir com os textos que você deseja que o Vim procure por você e olhar para o canto inferior do console.
- /\c → A opção, quando direcionada para o comando search (/), permite que os usuários pesquisem textos com distinção entre maiúsculas e minúsculas.A utilização sábia desse comando pode economizar horas de trabalho duro.
- ?[pattern] → Este é um dos comandos mais úteis do Vim para pesquisar textos anteriores para um determinado [pattern].

- n → O comando n pesquisa na direção da sua última pesquisa.Use este comando se souber em qual direção está o item de pesquisa.
- N → Quase idêntico ao comando acima, mas pesquisa na direção oposta à sua última pesquisa.
- :%s/[pattern]/[replacement]/g → O comando acima utiliza expressão regular para pesquisar todas as ocorrências de [padrão] e substitui-o por [substituição] sem solicitar confirmação.
- :%s/[pattern]/[replacement]/gc → Igual ao comando anterior, mas solicita confirmação antes de substituir cada instância de [padrão] por [substituição].
- 😒/[pattern]/[replacement]/g → Em vez de substituir todas as instâncias de [padrão] em seu arquivo, este comando Vim substituirá apenas aqueles [padrão] que estão na linha atual por [substituição].
- :bufdo /[pattern] → Este é um dos poderosos comandos do Vim que permitem aos usuários procurar por [pattern] em todos os buffers abertos no momento.Isso aumentará sua produtividade e diminuirá significativamente o tempo de pesquisa.
- :g/string/d → Este é um dos comandos úteis do Vim que serão úteis sempre que você desejar excluir todas as linhas que contenham string do seu documento.Folha de dicas de comandos do Linux para trabalhar com vários arquivos no Vim

2.5 Manipulação da Tela e Edição

- :sp [filename] → Use este comando para criar um novo arquivo e divide a tela do console horizontalmente para mostrar os dois buffers diferentes.
- :vsp [filename] → A funcionalidade deste comando Vim é, em essência, idêntica ao comando acima, mas em vez de dividir o console horizontalmente, ele divide a tela verticalmente.
- :bn → Este comando do Vim mudará seu editor para o próximo buffer.Está entre os poucos comandos fundamentais do Vim sem os quais você não poderá trabalhar com vários documentos no Vim.
- :bp → Idêntico ao comando anterior, mas alterna para o buffer anterior em vez de avançar.
- :bd → Use este comando Vim ao fechar um buffer específico.Salve seus dados usando os comandos apropriados do Vim.
- :ls → Este é um dos comandos úteis do Vim que apresentará aos usuários uma lista de todos os buffers abertos.
- ctrl+ws → Se você deseja dividir as janelas do Vim horizontalmente, este é o comando que você está procurando.
- ctrl+wv → Em vez de dividir as janelas horizontalmente, este comando do Vim o dividirá verticalmente.
- ctrl+ww → Utilize este comando para alternar entre várias janelas diretamente do modo de comando.
- ctrl+wq → Você pode usar este comando útil do Vim para sair de uma janela específica.
- ctrl+wh → Este comando move a localização do seu cursor para a janela esquerda.
- ctrl+wl → Igual ao comando anterior, mas em vez de mover o cursor para a esquerda, este comando apontará para a janela direita.
- ctrl+wj → Use este comando sempre que quiser mover uma janela abaixo da janela existente.

- ctrl+wk → O mesmo que acima, mas leva o cursor para a janela acima da atual.Comandos úteis do Vim ao trabalhar com várias guias
- :tabnew → Você pode usar o comando: tabnew para criar uma nova guia e trabalhar com outro documento sem sair do arquivo atual.
- gt → O comando gt mostrará a próxima guia aberta por você.
- :tabfirst → O comando acima mostra a primeira guia que você abriu em uma sessão específica.
- :tablast → Igual ao comando anterior, mas em vez de mostrar a primeira guia, ele exibirá a última guia.
- tabm n(position) → Este poderoso comando do Vim será útil sempre que você sentir a necessidade de reorganizar suas guias existentes.
- tabdo %s/foo/bar/g → Você pode utilizar o comando acima sempre que desejar executar um comando em todas as guias abertas ao mesmo tempo.
- :tab ball → Este comando do Vim é um dos meus comandos favoritos do Vim e coloca cada arquivo aberto em uma :tab ball .
- :new abc.txt → Este é um dos comandos do Vim que permitem abrir um novo arquivo chamado abc.txt em uma nova janela sem sair do documento atual.Comandos diversos do Vim
- :w → Pressionar este comando Vim no modo de comando salva o documento atual, mas não existe a sessão existente.
- :q → Este comando sai da sessão atual sem salvar suas alterações.Observe que você verá o erro E37 se tiver alterações não salvas no seu documento.Em tais cenários, você precisa substituir este comando e usar q!em vez de.
- :help [command] → O comando help realiza uma operação de pesquisa no comando digitado e mostra informações relevantes sobre eles diretamente no console.
- :e [file] → Este comando abrirá um arquivo chamado [arquivo] e criará um novo se ele já não existir no seu sistema de arquivos.
- :w [filename] → Use este comando para salvar o documento existente diretamente em um novo arquivo chamado [nome do arquivo].
- :stop → Escrever este comando no modo de comando suspenderá sua sessão atual do Vim.Você também pode fazer isso pressionando ctrl + z ao mesmo tempo.
- :browse e → Use este comando sempre que desejar chamar o explorador de arquivos gráficos a partir do seu console Vim.
- :%!fmt → Escrever este comando alinhará cada linha do seu arquivo atual.
- !}fmt → Use-o sempre que precisar alinhar todas as linhas na posição atual do seu cursor.
- :set autoindent → Este é um dos comandos mais usados do Vim que você usará durante o seu tempo no Vim.Ele define o autoindent para sua sessão atual.Pensamentos finais