VIM

Estou aprendendo VIM e fiz algumas anotações do que achei mais importante.

Modos:

1. Insert Mode: **No modo Inserir (insert), você digitará a letra w, como em qualquer outro editor.**
2. Normal Mode: **No modo Normal, você moverá o cursor para o início da próxima palavra.**
3. Visual Mode: **No modo Visual, você selecionará o texto até e incluindo o início da próxima palavra.**

Comandos básicos:

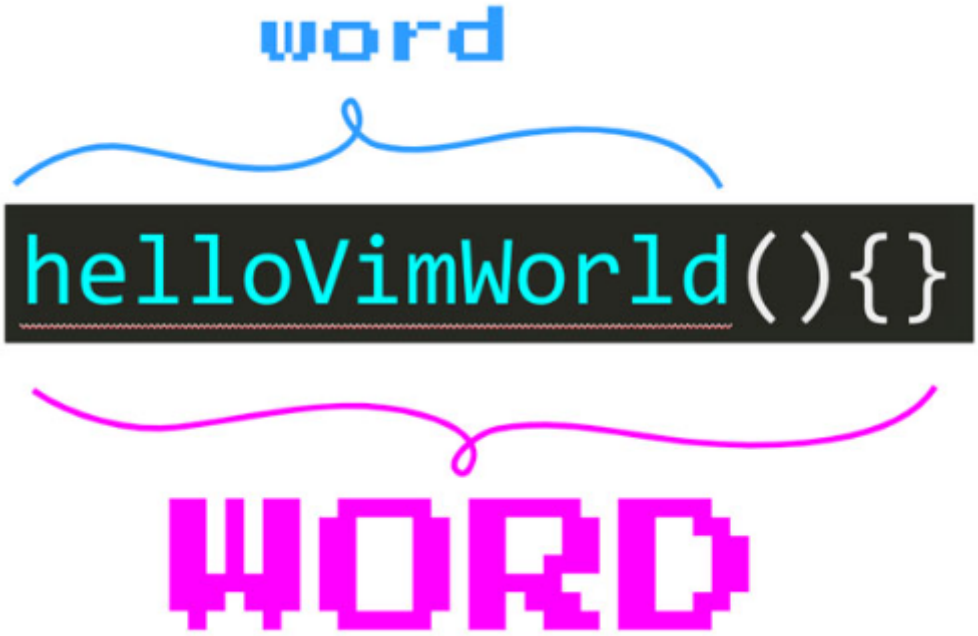
- `h`: Move uma casa para a esquerda.
- `j`: Desce para a linha de baixo.
- `k`: Sobe uma linha.
- `l`: Move uma casa para a direita.
- `i`: Entra no modo insert.
- `w`: Vai para a primeira letra da próxima palavra.
- `e`: Vai para a última letra da próxima palavra.
- `b`: Volta para a primeira letra da palavra anterior.
- `ge`: Volta para a última letra da palavra anterior.
- `Ctrl + c`: **Retorna para o modo NORMAL.**
- `Ctrl + v`: **Não cola mais e, em vez disso, envia você para o modo de bloqueio visual.**
- `Ctrl + f`: **Não permite mais pesquisar e, em vez disso, permite rolar a tela uma página para frente.**
- `/alguma-coisa:` Faz a função do `Ctrl+f`, procurando algo.
- `y`: Copia.
- `p`: Cola.

Portanto, w, b, e e ge permitem que você mova palavra por palavra no Vim. Mas o que é exatamente uma palavra? Uma palavra no Vim é:

1. A sequence of letters, digits and numbers
2. A sequence of other non-blank characters

```
these are 4 words
and these below too
,,, ..... ((((( ,.(
```

Mas o Vim também tem o conceito de tipos especiais de palavras (com letras, dígitos e números) que também incluem caracteres especiais como ., (, {, etc. Eles são chamados de PALAVRAS no jargão do Vim:



As PALAVRAS (WORD) são particularmente úteis para nós, programadores, porque o código costuma ter muitas delas:

```
this is a WORD: Iam_A_WORD(WORD)
this function call sum(2,3) is also a WORD
this array [1,2,3,4,5] is a WORD as well
```

Se você quiser mover PALAVRA (WORD) por PALAVRA (WORD), você pode usar os equivalentes em letras maiúsculas dos movimentos descritos anteriormente (`W, B, E, gE`).

Em geral, os movimentos de palavras (word) permitem mudanças mais precisas, enquanto os movimentos de PALAVRAS (WORD) permitem movimentos mais rápidos:

```
wwww ==> v   v v   v   v
         word. are two words
         word. is one WORD
WWW  ==> ^     ^  ^   ^
```

Comando caracteres:

- `f{caractere}`: Com o `f` podemos procurar um caractere específico na linha em que estamos, então pularemos para a próxima ocorrência deste caractere.
- `F{caractere}`: Com o `F` podemos procurar um caractere específico na linha em que estamos, então pularemos para a ocorrência anterior deste caractere.

Podemos ver claramente como f é mais rápido e preciso do que usar movimentos de palavras, lançando um contra o outro em um exemplo:

```
f(   ==> v                       v
         const fireball = function(target){
wwww ==> ^     ^        ^ ^       ^
```

Depois de usar `f{character}` você pode digitar `;` para ir para a próxima ocorrência do caractere ou `,` para ir para a anterior. Você pode ver o `;` e `,` como comandos para repetir a busca do último caractere. Isso é bom porque evita que você digite a mesma pesquisa repetidamente:

```
fdfdfd ==> v   v              v        v
           let damage = weapon.damage * d20();
           let damage = weapon.damage * d20();
fd;;   ==> v   v              v        v
```

Além do f, o Vim também oferece o comando t (until):

- `t{character}`: Use `t{character}` para mover o cursor imediatamente antes da próxima ocorrência de um caractere (pense em t{character} para mover seu cursor até aquele caractere).
- `T{character}`: Novamente, você pode usar `T{character}` para fazer o mesmo que t{character}, mas ao contrário

Se a diferença entre os comandos f e t ainda não está muito clara, aqui está um exemplo que compara os dois:

```
t(   ==> v                       v
         const fireball = function(target){
f(   ==> ^
```

`t` é realmente útil quando você combina movimentos com operadores para realizar alterações de texto, como você descobrirá em breve (por exemplo, você pode excluir tudo até `(` e alterá-lo para outra coisa).