

המדריך למתכנת טוב

עובדה

כדי להצליח בתחרות צריך לפתור שאלות מהר

עובדה

כדי להצליח בתחרות צריך לפתור שאלות מהר
כדי לעשות זאת נלמד היום איך משתמשים בעורך Vim

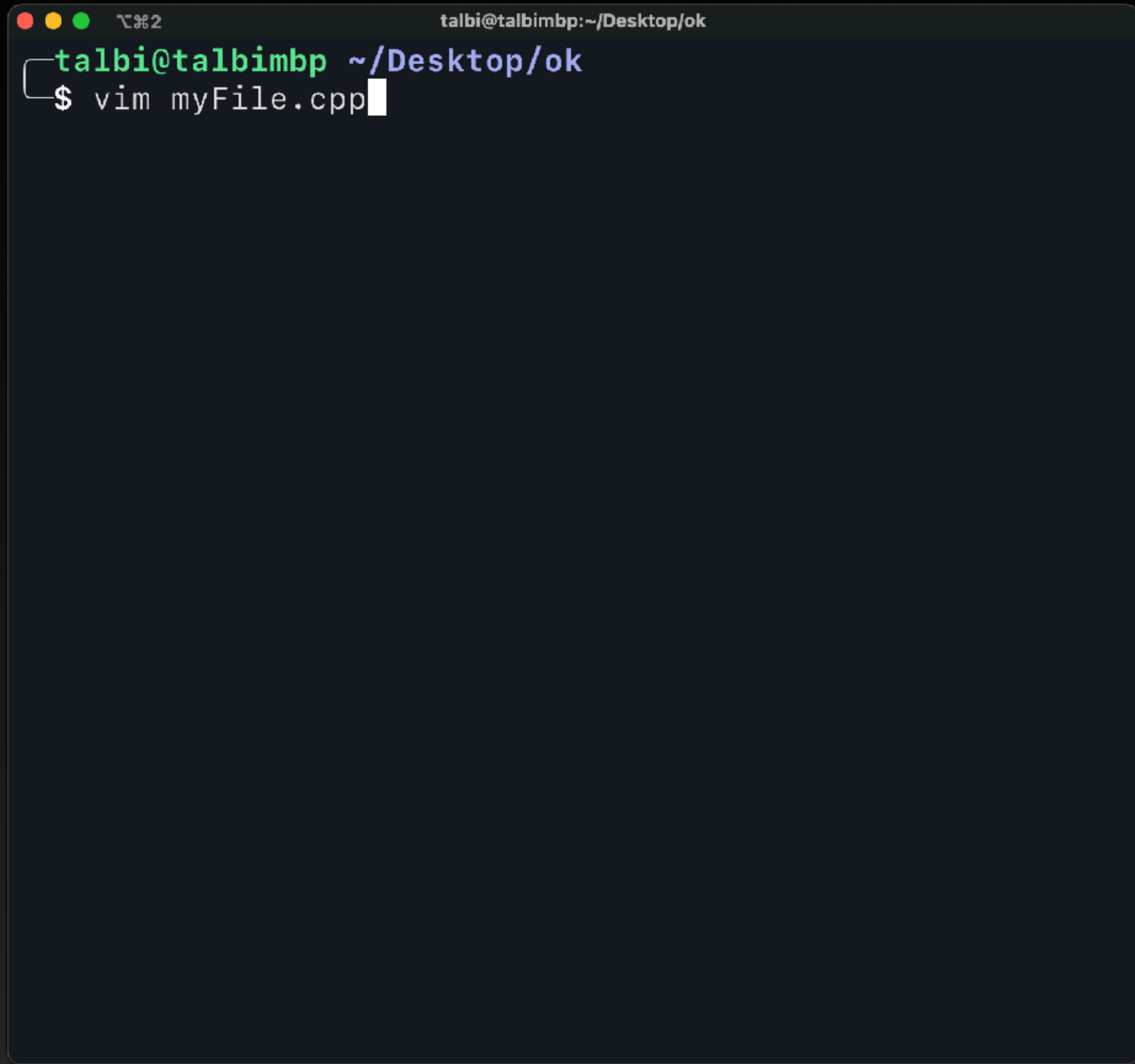
“Vim is Awesome!”

— אני

וברצינות, יש ל Vim יתרונות על עורכים אחרים

- לאחר שמתרגלים לתוכנה עריכת קוד נהיית מהירה יותר
- פחות הסחות דעת (אולם חלק אוהבים את ה-IntelliSense, אני פחות)
- אפשר להוציא מ Vim מלא פיצ'רים קטנים שמשפרים את איכות החיים
- בהשוואה לכמעט כל עורך אפשרי, Vim יותר מהיר
- Vim הוא תוכנה מהטרמינל בלבד, דבר שמאפשר לשמור, לקמפל ולהריץ את התוכנית מאותו המקום בדיוק
- ניתן לקנפרג כמעט כל דבר!
- ניתן להשוויץ בכך שאתם יודעים לצאת מ Vim

דאָוואַ, הבה נלמד!

A terminal window with a dark blue background. The title bar at the top shows three colored circles (red, yellow, green) and the text 'talbi@talbimbp:~/Desktop/ok'. The terminal content shows a prompt 'talbi@talbimbp ~/Desktop/ok' followed by the command '\$ vim myFile.cpp' with a white cursor at the end of the line.

```
talbi@talbimbp:~/Desktop/ok
talbi@talbimbp ~/Desktop/ok
$ vim myFile.cpp
```

Introduction

- כפי שאמרנו, Vim מהטרמינל בלבד!
 1. פתרו את הטרמינל (בווינדוס, השתמשו בWSL כמובן)
 2. כתבו `vim myFile.cpp` כדי ליצור קובץ חדש בשם `myFile.cpp`



Introduction

- כפי שאמרנו, Vim מהטרמינל בלבד!

1. פתרו את הטרמינל (בווינדוס, השתמשו בWSL כמובן)

2. כתבו `vim myFile.cpp` כדי ליצור קובץ חדש בשם `myFile.cpp`

3. יש! יצרנו קובץ אבל:

- אי אפשר לערוך אותו?

- אי אפשר לכתוב לו כלום?

- איך שומרים???

- איך יוצאים?????



```
vim myFile.cpp
WOW!
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --          1,5          A11
```

Introduction

1. כדי לערוך קובץ, נלחץ i (שימו לב! i קטנה)
2. זה מסמל Insert Mode, המצב המאפשר כתיבה!
3. כאשר פתחנו את Vim, הוא בא במצב Normal Mode המאפשר לנו להכניס פקודות כדי לערוך ("פיזית") את הקובץ – נלמד על פקודות אלו בהמשך
4. בעצם, כמעט תמיד תהיו במצב Normal Mode!

Introduction

הבה נכתוב קוד אמיתי!

- נשים לב שVim לבד זיהה את הקוד שאנו כותבים (לפי סיומת הקובץ).
- ייתכן שאצלכם הצבעים יראו שונה, זה בסדר.
- בנוסף, ייתכן שאצלכם יהיו Tabs במקום Spaces, או לא מופיעים המספרים בתחתית המסך (5, 35) – בכל הדברים הללו נטפל בהמשך!
- לאחר שכתבנו תוכנית, נרצה לשמור אותה. כיצד?

Introduction

כדי לשמור קובץ (או לעשות דבר שהוא
לא כתיבה), תחילה נצטרך לצאת
מ־Insert Mode

1. נלחץ על Escape במקלדת

Introduction

כדי לשמור קובץ (או לעשות דבר שהוא
לא כתיבה), תחילה נצטרך לצאת
מ־Insert Mode

1. נלחץ על Escape

2. נלחץ על :

(מצב זה נקרא גם Command Mode)

```
vim myFile.cpp
#include <bits/stdc++.h>
using namespace std;

int main(void) {
    cout << "Hello World!" << endl;
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

[illegible]

Introduction

כדי לשמור קובץ (או לעשות דבר שהוא
לא כתיבה), תחילה נצטרך לצאת
Insert Mode

1. נלחץ על Escape

2. בלחץ על :

(מצב זה נקרא גם Command Mode)

3. נכתוב pq – כל אחת מן האותיות
אומרת פקודה אחרת: write,
quit

Introduction

כדי לשמור קובץ (או לעשות דבר שהוא
לא כתיבה), תחילה נצטרך לצאת
מ־Insert Mode

1. נלחץ על Escape

2. בלחץ על :

(מצב זה נקרא גם Command Mode)

3. נכתוב pq – כל אחת מן האותיות
אומרת פקודה אחרת: write,
quit

4. נלחץ Enter וזהו! כתבתם תוכנית



רגע, זה לא הרבה פחות יעיל?

- ובכן, כמתחילים זה אכן נראה כאילו הרבה פחות נוח ויעיל להשתמש בVim
- גם כשאני התחלתי להשתמש בVim לקח לי חצי שעה לכתוב תוכנית אחת, אבל ככל שמשתמשים יותר ויותר מתרגלים ואף מעדיפים את Vim על פני עורכים אחרים
- כמובן, שלהשתמש רק במה שהראתי בשקופיות הקודמות יהיה חסר כל הגיון, ולכן כעת נראה דרכים ליעל את העריכה (ובעצם, אלו הם הסיבות האמיתיות ללמה משתמשים בVim)
- אבל לפני זאת, הבה נראה קצת קינפרוגים סטנדרטיים לVim!

קינפרוגים של Vim

- כדי לקבוע הגדרות נערוך את הקובץ `~/.vimrc`
- ברוב התוכנות בLinux הקונבנציה היא ליצור קובץ בבית המשתמש (`home user` directory) אשר נקרא `<program name>.rc` – וגם Vim עושה זאת!
- בקובץ הזה נשים את כל הקינפרוגים המגניבים שלנו
- נשים לב, כאשר אנו פותחים את Vim התוכנה תריץ את הקובץ וטטעון את הקינפרוגים שלנו אוטומטית, ולכן אולי תרצו להיזהר טיפה ולא לכתוב דברים מצחיקים שם

Basic Configs

- כדי לקבוע הגדרה ספציפית נשתמש ב-set. זוהי מילה בשפת התכנות Vim, ואומרת לשים ערך כלשהו (או להגדיר) במשתנה.
- לדוגמה, כדי להפעיל את המספרים בתחתית המסך (המייצגים את מספר השורה וכו') נשתמש במשתנה ruler

```
syntax on
set ruler

"set tab to 4 spaces
filetype plugin indent on
set tabstop=4
set shiftwidth=4
set expandtab

"enable backspace (ok)
set backspace=2
set backspace=indent,eol,start
~
~
~
~
~
~
~
~
~
~
~
~
```

Basic Configs

```
syntax on
set ruler

"set tab to 4 spaces
filetype plugin indent on
set tabstop=4
set shiftwidth=4
set expandtab

"enable backspace (ok)
set backspace=2
set backspace=indent,eol,start
```

- כדי לקבוע הגדרה ספציפית נשתמש ב-set. זוהי מילה בשפת התכנות Vim, ואומרת לשים ערך כלשהו (או להגדיר) במשתנה.
- לדוגמה, כדי להפעיל את המספרים בתחתית המסך (המייצגים את מספר השורה וכו') נשתמש במשתנה ruler

1,1

All

Basic Configs

```

syntax on
set ruler

"set tab to 4 spaces
filetype plugin indent on
set tabstop=4
set shiftwidth=4
set expandtab

"enable backspace (ok)
set backspace=2
set backspace=indent,eol,start

```

1, 1

All

- כדי לקבוע הגדרה ספציפית נשתמש ב-set. זוהי מילה בשפת התכנות Vim, ואומרת לשים ערך כלשהו (או להגדיר) במשתנה.
- לדוגמה, כדי להפעיל את המספרים בתחתית המסך (המייצגים את מספר השורה וכו') נשתמש במשתנה ruler
- באותו אופן נוכל להפעיל מספר שורה בעזרת number, או מספר שורה ביחס למיקום הנוכחי בעזרת relativenumber

Basic Configs

• הערות (comments) הם בעזרת התו

//

- כפי שניתן לראות כך נהפוך Tabs לSpaces!

- בינתיים מספיק להעתיק את שורות אלה ובשיעור Vim 2 נלמד מה זה אומר

- למשתמשים בWSL, ייתכן שתצטוו להוסיף גם את השורות הללו: ####

```
syntax on
set ruler

"set tab to 4 spaces
filetype plugin indent on
set tabstop=4
set shiftwidth=4
set expandtab

"enable backspace (ok)
set backspace=2
set backspace=indent,eol,start

~
~
~
~
~
~
~
~
~
~
~
~
~
```