

به نام خدا



git

آشنایی با گیت

مقدمه

یکی از مهم‌ترین ابزارهایی که یک برنامه‌نویس با آن سروکار دارد، گیت (git) است. کاربرد گیت کمک به ثبت روند تغییرات کدها و فایل‌های برنامه و همچنین مدیریت و کنترل نسخه‌ی نرم‌افزار (version control) می‌باشد. با استفاده از گیت می‌توان به صورت گروهی برنامه‌نویسی کرد. افراد گروه هم‌زمان روی سیستم‌های خود روی یک فایل مشترک یا چند فایل کار می‌کنند و در آخر به کمک این ابزار کدهایشان را با یکدیگر ترکیب می‌کنند.

مسئله

فرض کنید قرار است پروژه‌ای با موضوع مبارزه با کرونا را از جایی در اینترنت به کمک گیت دریافت کرده و آن را چند مرحله به پیش ببریم.

گام‌هایی که قرار است طی کنیم به این صورت است:

۱. دریافت پروژه از اینترنت
۲. اصلاح کد و مدیریت خطا
۳. اضافه کردن قابلیت جدید «منو» در کپی دیگری از آن کد به صورت موقت (کپی ۱)
۴. تکمیل و انجام اصلاحات کامل در آن (در کپی ۱)
۵. ذخیره‌ی لاگ در
۶. روش تشخیص بر اساس سرفه
۷. ترکیب کدهای کپی ۱ با کدهای اصلی
۸. افزودن گزینه‌ی توصیه‌های پزشکی به منوها

راه اندازی

برای نصب گیت از طریق ترمینال اقدام می کنیم:

```
→ Desktop sudo apt-get install git
```

پس از نصب گیت، دستور «git» به مجموعه دستورات ما اضافه می شود.

پیاده سازی

برای دریافت پروژه از سایت [گیت هاب](https://github.com/alikhezri/koronavarzi) که محلی برای به اشتراک گذاری پروژه هایی است که از گیت استفاده کرده اند، از دستور زیر استفاده کرده و آدرس پروژه به عنوان ورودی می دهیم:

```
→ Desktop git clone https://github.com/alikhezri/koronavarzi.git
Cloning into 'koronavarzi'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
→ Desktop
```

وارد پوشه ی پروژه که ایجاد شده می شویم:

```
→ Desktop cd koronavarzi
→ koronavarzi git:(master)
```

بعد از مطالعه ی کد تصمیم به اصلاح آن و مدیریت حالتی که کاربر ورودی نامعتبر وارد کرد، می کنیم.

برای تغییر فایل از ویرایشگر «nano» استفاده می کنیم:

```
KORONA_TEMPRETURE = 37.5

temp = int(input("Enter Your Body Tempreature: "))

def tab_sanj():
    try:
        if temp < KORONA_TEMPRETURE:
            print("You are in safe range.")
        else:
            print("Danger! Visit a doctor.")
    except ValueError:
        print("Wrong value entered. Try again later.")

if __name__ == '__main__':
    tab_sanj()
```

برای دیدن تغییرات انجام شده بر روی فایل از دستور «git diff» زیر استفاده می‌کنیم. خروجی به شکل زیر می‌باشد:

```
diff --git a/korona_diagnose.py b/korona_diagnose.py
index c6c568a..bbde92e 100644
--- a/korona_diagnose.py
+++ b/korona_diagnose.py
@@ -3,10 +3,13 @@ KORONA_TEMPRETURE = 37.5
temp = int(input("Enter Your Body Tempreature: "))

def tab_sanj():
-     if temp < KORONA_TEMPRETURE:
-         print("You are in safe range")
-     else:
-         print("Danger! Visit a doctor")
+     try:
+         if temp < KORONA_TEMPRETURE:
+             print("You are in safe range.")
+         else:
+             print("Danger! Visit a doctor.")
+     except ValueError:
+         print("Wrong value entered. Try again later.")

if __name__ == '__main__':
(END)
```

برای ثبت تغییرات ابتدا باید فایل‌ها را به گیت اضافه کنیم. برای این کار از دستور زیر استفاده می‌کنیم:

```
→ koronavarzi git:(master) X git add korona_diagnose.py
→ koronavarzi git:(master) X
```

برای اینکه ببینیم وضعیت فایل‌ها چگونه است، کدام‌ها تغییر کرده و کدام‌ها اضافه شدند، باید از دستور زیر استفاده کرد:

```
→ koronavarzi git:(master) X git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   korona_diagnose.py

→ koronavarzi git:(master) X
```

حال باید تغییراتی را که اضافه کرده‌ایم، ثبت کنیم:

```
→ koronavarzi git:(master) X git commit -m "Add try-except to handle errors."
[master 176f8e2] Add try-except to handle errors.
1 file changed, 7 insertions(+), 4 deletions(-)
→ koronavarzi git:(master) X
```

حال می‌خواهیم قابلیت جدید «منو» را، که ممکن است کار زیادی ببرد، به این برنامه اضافه کنیم اما نمی‌خواهیم به کد فعلی دست بزنیم. برای همین، با استفاده از گیت، پیاده‌سازی آن را در یک کپی از کدی که تا به اینجا نوشته‌ایم، شروع می‌کنیم.

ابتدا باید با استفاده از گیت کدی را که تا اینجا زده شده، در جایی دیگر به‌نام مثلاً «menu» کپی

کنیم. ابتدا باید «menu» را ایجاد کنیم:

```
→ koronavarzi git:(master) git branch menu
```

سپس وارد آن می‌شویم:

```
→ koronavarzi git:(master) git checkout menu
Switched to branch 'menu'
```

کدهای لازم برای اضافه کردن منو:

```
if __name__ == '__main__':
    while:
        item = input("Select from items('tabsanj', 'exit'): ")
        if item == 'tabsanj':
            tab_sanj()
        elif item == 'exit':
            break
```

سپس تغییرات را برای ثبت کردن در گیت اضافه کرده و سپس ثبت می‌کنیم:

```
→ koronavarzi git:(menu) X git add korona_diagnose.py
→ koronavarzi git:(menu) X git commit -m "Add menu to application"
[menu b4d5805] Add menu to application
1 file changed, 7 insertions(+), 1 deletion(-)
→ koronavarzi git:(menu)
```

لازم است خروجی مناسب در حالتی که گزینه‌ی نامعتبر توسط کاربر وارد شد، چاپ شود. گرفتن تغییرات پس از تغییر فایل:

```
diff --git a/korona_diagnose.py b/korona_diagnose.py
index 2d17a73..95ab903 100644
--- a/korona_diagnose.py
+++ b/korona_diagnose.py
@@ -19,4 +19,5 @@ if __name__ == '__main__':
     tab_sanj()
     elif item == 'exit':
         break
-
+         else:
+             print("Wrong item selected. Please enter a correct one!")
(END)
```

تغییرات را اضافه کرده و در گیت ثبت می‌کنیم:

```
→ koronavarzi git:(menu) X git add korona_diagnose.py
→ koronavarzi git:(menu) X git commit -m "Improve handling items."
[menu f9cf682] Improve handling items.
1 file changed, 2 insertions(+), 1 deletion(-)
→ koronavarzi git:(menu)
```

اگر دقت کنیم، می‌بینیم در شرطی که بررسی می‌کند این فایل اجرا شده فایل «__main__» است، به جای ۲ مساوی یک مساوی گذاشتیم. این خطای نحوی در کد اصلی هم وجود دارد. ابتدا این ایراد

کوچک را اینجا برطرف می‌کنیم که توضیح آن لازم نیست. و سپس به شاخه‌ی اصلی که در ابتدا در آن بودیم، می‌رویم و آن را اصلاح می‌کنیم.

```
→ koronavarzi git:(menu) git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)
→ koronavarzi git:(master) nano korona_diagnose.py
```

تغییرات به صورت زیر است.:

```
diff --git a/korona_diagnose.py b/korona_diagnose.py
index bbde92e..796b104 100644
--- a/korona_diagnose.py
+++ b/korona_diagnose.py
@@ -12,5 +12,5 @@ def tab_sanj():
     print("Wrong value entered. Try again later.")

-if __name__ == '__main__':
+if __name__ == '__main__':
     tab_sanj()
(END)
```

تغییرات را اضافه و ثبت می‌کنیم:

```
→ koronavarzi git:(master) X git add korona_diagnose.py
→ koronavarzi git:(master) X git commit -m "Fix syntax error."
[master 5ff051d] Fix syntax error.
1 file changed, 1 insertion(+), 1 deletion(-)
→ koronavarzi git:(master) █
```

حال که همه‌ی تغییرات ثبت شده و «منو» نیز قبلاً تکمیل شده بود می‌توانیم کدهای «منو» را با کد اصلی به کمک گیت ترکیب کنیم. برای این کار از دستور زیر استفاده می‌کنیم:

```
→ koronavarzi git:(master) git merge menu
Auto-merging korona_diagnose.py
CONFLICT (content): Merge conflict in korona_diagnose.py
Automatic merge failed; fix conflicts and then commit the result.
→ koronavarzi git:(master) X █
```

عمل ترکیب کردن به صورت خودکار انجام می‌شود. اما گاهی گیت نمی‌تواند انتخاب درست را تشخیص دهد. در این جور موارد، گیت مواردی را که نتوانست تشخیص دهد، در فایل مشخص می‌کنید تا کاربر خودش انتخاب کند.

```

diff --cc korona_diagnose.py
index 796b104,0d691b3..0000000
--- a/korona_diagnose.py
+++ b/korona_diagnose.py
@@@ -13,4 -13,11 +13,15 @@@ def tab_sanj()

    if __name__ == '__main__':
++<<<<<< HEAD
+     tab_sanj()
++=====
+     while:
+         item = input("Select from items('tabsanj', 'exit'): ")
+         if item == 'tabsanj':
+             tab_sanj()
+         elif item == 'exit':
+             break
+         else:
+             print("Wrong item selected. Please enter a correct one!")
++>>>>>> menu
(END)

```

آن بخشی را که می‌خواهیم نگه می‌داریم و بخشی را که نمی‌خواهیم به همراه خطوطی که در آن

«=»های متوالی و عبارت «HEAD» و «menu» است، حذف می‌کنیم.

پس از انجام تغییرات دوباره فایل را اضافه کرده و سپس تغییرات را ثبت می‌کنیم:

```

→ koronavarzi git:(master) X git add korona_diagnose.py
→ koronavarzi git:(master) X
→ koronavarzi git:(master) X git commit
[master 74c3a1d] Merge branch 'menu'
→ koronavarzi git:(master) 

```

برای دیدن روند کارهایی که تا به الان کرده‌ایم از دستور «git log» استفاده می‌کنیم. خروجی روند به

صورت زیر است:


```
commit 74c3a1d8b43f431d5dcab55895796b8ecc88e421 (HEAD -> master)
Merge: 5ff051d 515651c
Author: null <somebody@domain.com>
Date: Thu Apr 16 18:08:43 2020 +0430

    Merge branch 'menu'

commit 515651cd9e9f2780f8f142462071bf07e31d0718 (menu)
Author: null <somebody@domain.com>
Date: Thu Apr 16 17:43:12 2020 +0430

    Fix syntax error.

commit 5ff051d98aa3db23dddc792066e31aa1a6258786
Author: null <somebody@domain.com>
Date: Thu Apr 16 15:30:13 2020 +0430

    Fix syntax error.

commit f9cf68242bfc8722f688c2abdd65d177c2ad9670
Author: null <somebody@domain.com>
Date: Thu Apr 16 14:19:48 2020 +0430

    Improve handling items.

commit b4d580516121ad50a47c9cda05c4af8b4e7be014
Author: null <somebody@domain.com>
Date: Thu Apr 16 14:06:27 2020 +0430

    Add menu to application

commit 176f8e208fab9dfa2eee8f049e2b18575b6b441b
Author: null <somebody@domain.com>
Date: Thu Apr 16 12:39:46 2020 +0430

    Add try-except to handle errors.

commit 930820351b1ed36fb1669cdc850627032a2875c6 (origin/master, origin/HEAD)
Author: null <somebody@domain.com>
Date: Wed Apr 15 19:31:59 2020 +0430

    Add tab_sanj method for diagnose
(END)
```

نمایی دیگر از کاری که انجام دادیم به کمک دستور:

```
→ koronavarzi git:(master) git log --graph --oneline --decorate
```

و خروجی:

```

* 74c3a1d (HEAD -> master) Merge branch 'menu'
| \
| * 515651c (menu) Fix syntax error.
| * f9cf682 Improve handling items.
| * b4d5805 Add menu to application
| * 5ff051d Fix syntax error.
|/
* 176f8e2 Add try-except to handle errors.
* 9308203 (origin/master, origin/HEAD) Add tab_sanj method for diagnose
(END)

```

برای ذخیره کرد پروژه بر روی سایت گیت‌هاب (جایی که پروژه را از آنجا برداشتیم)، از دستور «git push» استفاده می‌کنیم.

پس از وارد کردن دستور از نام کاربری و گذرواژه‌ی سوال می‌شود. در صورت صحیح بودن، پروژه بر روی آن باگذاری می‌شود:

```

→ koronavarzi git:(master) git push
Username for 'https://github.com': alikhezri
Password for 'https://alikhhezri@github.com':
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Compressing objects: 100% (11/11), done.
Writing objects: 100% (16/16), 1.60 KiB | 1.60 MiB/s, done.
Total 16 (delta 5), reused 0 (delta 0)
remote: Resolving deltas: 100% (5/5), done.
To https://github.com/alikhezri/koronavarzi.git
9308203..74c3a1d master -> master
→ koronavarzi git:(master)

```

مباحثی که ذکر شد، تنها مقدماتی از گیت بود.
موفق باشید.