



# دوره آموزشی طراحی و توسعه وب

Web Design and Development Training Course



مهارت نرم  
Soft Skill

مهارت فنی  
Tech. Skill

جلسه پنجم | Fifth Session

## برگه تقلب گیت

Git Cheat Sheet

محمد صالح گواهی

Mohammad Saleh Govahi

توسعه دهنده در مرکز مطالعات گروه نرم افزاری پارت  
Developer in Part Software Group Research Institute

01.

# گرفتن اسنپ شات از وضعیت

Creating Snapshot





## ساختن یک repository

کاربرد	دستور
ساختن یک ریپازیتوری	git init



## بردن فایل ها به وضعیت استیج

دستور	کاربرد
<code>git add &lt;&lt;filename&gt;&gt;</code> ( <code>git add index.html</code> )	بردن یک فایل با استفاده از نام فایل
<code>git add &lt;&lt;filename&gt;&gt; &lt;&lt;filename&gt;&gt;</code> ( <code>git add index.html a.html b.html</code> )	بردن چند فایل با استفاده از نام فایل ها
<code>git add &lt;&lt;pattern&gt;&gt;</code> ( <code>git add *.html</code> )	بردن یک یا چند فایل با استفاده از پترن ها
<code>git add .</code>	بردن تمامی فایل های موجود در یک پوشه



## مشاهده وضعیت فایل ها

کاربرد	دستور
مشاهده وضعیت کامل فایل ها (بیشترین دستوری که تو گیت استفاده میشه)	git status
مشاهده خلاصه وضعیت فایل ها	git status -s



## کامیت کردن فایل‌ها

دستور	کاربرد
<code>git commit -m "message"</code> <code>git commit -m "add index.html"</code>	کامیت کردن فایل‌هایی که وضعیت استیج دارن به همراه یک پیام
<code>git commit</code>	کامیت کردن فایل‌هایی که وضعیت استیج دارن (یک تکست ادیتور باز میشه که میتونین پیام کامیتتون رو داخل اون بنویسین)
<code>git commit -am "message"</code> <code>git commit -am "add css files"</code>	کامیت کردن فایل‌های تغییر یافته به صورت مستقیم (ردشدن از وضعیت استیج)



## حذف، تغییر نام و جابجایی فایل‌ها

کاربرد	دستور
حذف یک فایل از پوشه در حال کار و وضعیت استیج	<code>git rm &lt;&lt;filename&gt;&gt;</code> <code>git rm index.html</code>
حذف یک فایل فقط از وضعیت استیج	<code>git rm --cached &lt;&lt;filename&gt;&gt;</code> <code>git rm --cached index.html</code>
تغییر نام فایل	<code>git mv &lt;&lt;oldName&gt;&gt; &lt;&lt;newname&gt;&gt;</code> <code>git mv 1.html index.html</code>
جابجایی فایل	<code>git mv &lt;&lt;oldPath&gt;&gt; &lt;&lt;newPath&gt;&gt;</code> <code>git mv 1.html updated/index.html</code>



## مشاهده تغییرات

کاربرد	دستور
مشاهده تغییرات فایل های استیج نشده	git diff
مشاهده تغییرات فایل های در وضعیت استیج	git diff --staged git diff --cached





## برگرداندن یک فایل از وضعیت stage

کاربرد	دستور
برگرداندن یک فایل از حالت استیج	<code>git restore --staged &lt;&lt;filename&gt;&gt;</code> <code>git restore --staged index.html</code>



## برگرداندن فایل به وضعیت قبل از تغییرات انجام شده جدید در محیط `working directory`

کاربرد	دستور
برگرداندن یک فایل به وضعیت قبل از تغییرات جدید	<code>git restore &lt;&lt;filename&gt;&gt;</code> <code>git restore index.html</code>
برگرداندن چند فایل به وضعیت قبل از تغییرات جدید	<code>git restore &lt;&lt;filename&gt;&gt; &lt;&lt;filename&gt;&gt;</code> <code>git restore index.html a.html b.html</code>
برگرداندن تمامی فایل های به وضعیت قبل از تغییرات جدید	<code>git restore .</code>

# 02.

## مرور تاریخچه کامیت ها

Browsing Commits History





## مشاهده تاریخچه کامیت ها

کاربرد	دستور
مشاهده تمامی کامیت ها	<code>git log</code>
مشاهده کامیت ها و فایل های تغییر یافته مربوط به هر کامیت	<code>git log --stat</code>
مشاهده کامیت ها (ریز تغییرات)	<code>git log --patch</code>



## فیلترکردن مشاهده تاریخچه کامیت ها

کاربرد	دستور
مشاهده آخرین کامیت ها به اندازه عدد وارد شده	<code>git log -&lt;number&gt;</code> <code>git log -۳</code>
مشاهده کامیت هایی که نام نویسندشون با نامی که وارد کردیم یکی هست	<code>git log --author="&lt;name&gt;"</code> <code>git log --author="saleh"</code>
مشاهده کامیت هایی که قبل از تاریخ وارد شده زده شدن	<code>git log --before="&lt;date&gt;"</code> <code>git log --before="۲۰۲۴-۰۲-۱۲"</code>
مشاهده کامیت هایی که بعد از تاریخ وارد شده زده شدن	<code>git log --after="&lt;date&gt;"</code> <code>git log --after="۲۰۲۴-۰۲-۱۲"</code>



## فیلترکردن مشاهده تاریخچه کامیت ها

دستور	کاربرد
<code>git log --grep="&lt;&lt;text&gt;&gt;"</code> <code>git log --grep="update"</code>	مشاهده کامیت هایی که متنی که وارد کردیم رو توی پیام مربوط به خودشون دارن
<code>git log &lt;&lt;startCommit&gt;&gt; ..&lt;&lt;endCommit&gt;&gt;</code> <code>git log 1cFdcae4..4Rtyi8uy</code>	مشاهده کامیت هایی از کامیت شروع بازه تا کامیت پایان بازه هستن (البته کامیت پایان بازه رو نشون نمیده)
<code>git log &lt;&lt;filename&gt;&gt;</code> <code>git log index.html</code>	مشاهده کامیت های مربوط به یک فایل



## برخی دستورات برای ساده تر شدن کار

کاربرد	دستور
مشاهده تاریخچه با فرمت دلخواه	<pre>git log --pretty=format:"&lt;pattern&gt;" git log --pretty=format:"%an committed %h"</pre>
تعریف کردن دستور دلخواه	<pre>git config --global alias.&lt;wantedCommand&gt; "&lt;actualCommand&gt;" git config --global alias.lg "log --oneline"</pre>
مشاهده کامیت های مربوط به یک شخص	<pre>git shortlog</pre>



## مشاهده یک کامیت

کاربرد	دستور
مشاهده یک کامیت	<code>git show &lt;&lt;commitHash&gt;&gt;</code> <code>git show 1Rtfvft5Ty</code>
مشاهده نسخه ای از فایل ذخیره شده در یک کامیت	<code>git show &lt;&lt;commitHash&gt;&gt;:&lt;&lt;filename&gt;&gt;</code> <code>git show 1Rtfvft5Ty:index.html</code>





## مقایسه کامیت ها

کاربرد	دستور
مشاهده تغییرات بین دو کامیت	<code>git diff &lt;&lt;startCommit&gt;&gt; &lt;&lt;endCommit&gt;&gt;</code> <code>git diff Adrv۵e Auni۹o</code>
مشاهده کردن تغییرات بین دو کامیت در یک فایل مشخص	<code>git diff &lt;&lt;startCommit&gt;&gt; &lt;&lt;endCommit&gt;&gt;</code> <code>&lt;&lt;filename&gt;&gt;</code> <code>git diff Adrv۵e Auni۹o index.html</code>



## مشاهده کامیت ها

کاربرد	دستور
مشاهده کامیت	<code>git show &lt;&lt;commitHash&gt;&gt;</code> <code>git show Adc۲Rc</code>
مشاهده کردن تغییرات بین دو کامیت	<code>git diff &lt;&lt;startCommit&gt;&gt; &lt;&lt;endCommit&gt;&gt;</code> <code>git diff Adrv۵e Auni۹o</code>
مشاهده کردن تغییرات بین دو کامیت در یک فایل مشخص	<code>git diff &lt;&lt;startCommit&gt;&gt; &lt;&lt;endCommit&gt;&gt;</code> <code>&lt;&lt;filename&gt;&gt;</code> <code>git diff Adrv۵e Auni۹o index.html</code>



## اعمال تغییرات کامیت ها روی محیط

دستور	کاربرد
<code>git checkout &lt;&lt;commitHash&gt;&gt;</code> <code>git checkout Adc۲Rc</code>	اعمال تغییرات کامیت داده شده در محیط working directory
<code>git checkout master</code>	اعمال تغییرات آخرین کامیت شاخه اصلی در working directory



## مشاهده تاریخچه مربوط به یک فایل

دستور	کاربرد
<code>git log &lt;&lt;fileName&gt;&gt;</code> <code>git log index.html</code>	مشاهده کامیت هایی مربوط به یک فایل (کامیت هایی که فایل در آن ساخته یا آپدیت شده)
<code>git log --stat &lt;&lt;fileName&gt;&gt;</code> <code>git log --stat index.html</code>	مشاهده تعداد تغییراتی که در هر کامیت مربوط به فایل اومده
<code>git log --patch &lt;&lt;fileName&gt;&gt;</code> <code>git log --patch index.html</code>	مشاهده تغییراتی که در هر کامیت مربوط به فایل اومده
<code>git blame &lt;&lt;fileName&gt;&gt;</code> <code>git blame index.html</code>	مشاهده نویسندگان هر خط از فایل

# 03.

## شاخه ها و ادغام

Branches and Merge





## مدیریت شاخه ها

کاربرد	دستور
ساختن یک شاخه	<code>git branch &lt;&lt;branchName&gt;&gt;</code> <code>git branch fixBug</code>
تغییر شاخه به شاخه داده شده	<code>git checkout &lt;&lt;branchName&gt;&gt;</code> <code>git checkout fixBug</code>
تغییر شاخه به شاخه داده شده	<code>git switch &lt;&lt;branchName&gt;&gt;</code> <code>git switch fixBug</code>
تغییر شاخه به شاخه جدید به همراه ساخت شاخه	<code>git switch -C &lt;&lt;branchName&gt;&gt;</code> <code>git switch -C fixBug</code>
حذف یک شاخه	<code>git branch -d &lt;&lt;branchName&gt;&gt;</code> <code>git branch -d fixBug</code>



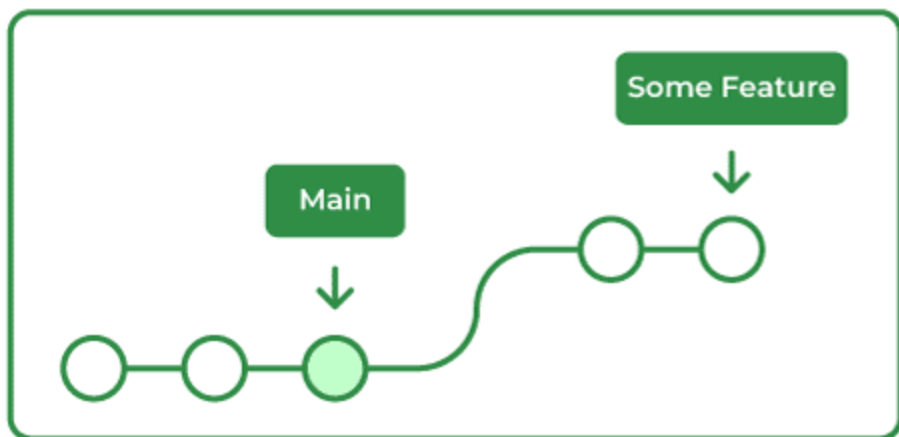
## مقایسه شاخه ها

کاربرد	دستور
مشاهده کامیت هایی که در شاخه دومی هستند ولی در شاخه اولی نیستند	<code>git log &lt;&lt;firstBranch&gt;&gt;..&lt;SecondBranch&gt;&gt;</code> <code>git log master..updateFile</code>
مشاهده خلاصه تغییرات بین دو شاخه	<code>git diff &lt;&lt;firstBranch&gt;&gt;..&lt;SecondBranch&gt;&gt;</code> <code>git diff master..updateFile</code>



## شیوه های merge – fast forward

Before Merging



After a Fast-Forward Merge

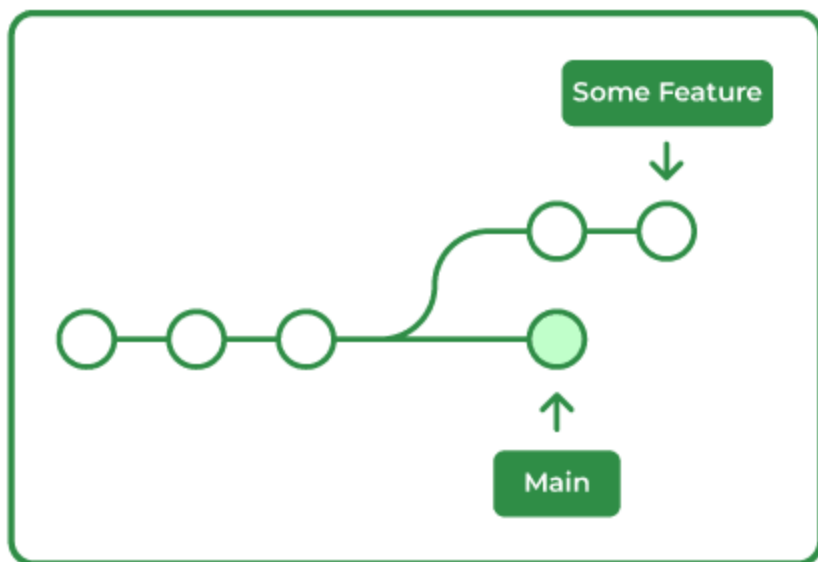




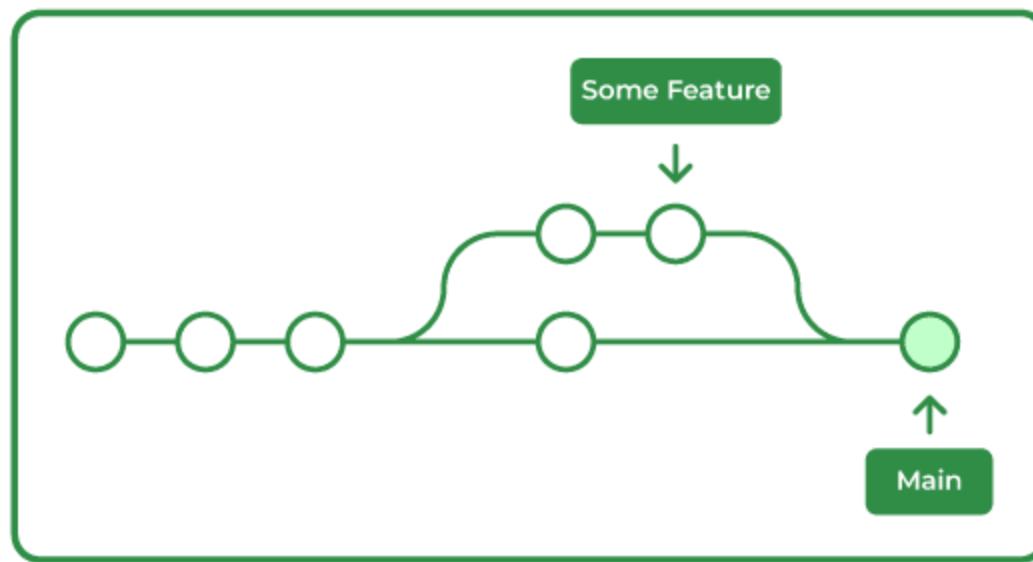


## شیوه های merge - three way

Before Merging



After a 3-way Merge





## ادغام شاخه ها

کاربرد	دستور
ادغام شاخه داده شده به شاخه کنونی	<code>git merge &lt;&lt;wantedBranch&gt;&gt;</code> <code>git merge updateFile</code>
ادغام به همراه کامیت حتی اگر فست فروارد باشد	<code>git merge --no-ff &lt;&lt;branchName&gt;&gt;</code> <code>git merge --no-ff updateFile</code>
تغییر وضعیت شاخه کنونی به قبل از ادغام	<code>git merge --abort</code>
مشاهده شاخه های ادغام شده	<code>git branch --merged</code>
مشاهده شاخه های ادغام نشده	<code>git branch --no-merged</code>
اعمال کامیت داده شده بر روی شاخه کنونی	<code>git cherry-pick &lt;&lt;commitHash&gt;&gt;</code>

# 04.

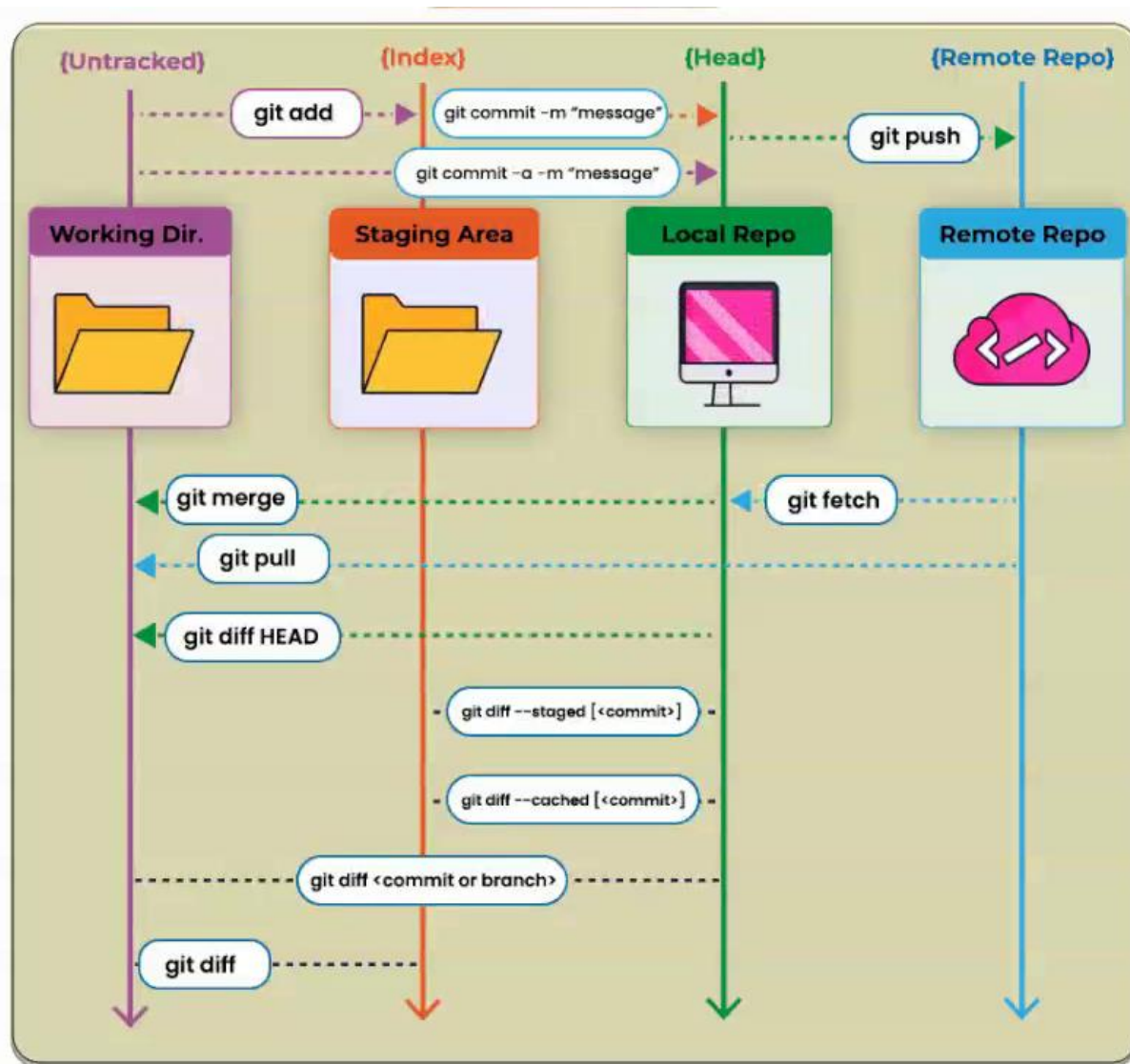
## گیت هاب و کار تیمی

Github and teamwork





# شماتیک کلی فضاها در گیت





## ایجاد یک کپی از repository

کاربرد	دستور
ایجاد یک کپی از ریپازیتوری ریموت در حافظه سیستم	<code>git clone &lt;&lt;url&gt;&gt;</code>



## همگام سازی با ریپازیتوری های ریموت

دستور	کاربرد
<code>git fetch &lt;&lt;remoteName&gt;&gt; &lt;&lt;remoteBranch&gt;&gt; git fetch origin master</code>	واکشی یک شاخه از ریپازیتوری ریموت
<code>git fetch &lt;&lt;remoteName&gt;&gt; git fetch origin</code>	واکشی تمامی آبجکت ها از ریپازیتوری ریموت
<code>git fetch</code>	واکشی تمامی آبجکت ها از ریپازیتوری ریموت



## همگام سازی با ریپازیتوری های ریموت

دستور	کاربرد
git pull	گرفتن تمامی آبجکت ها از ریپازیتوری ریموت و اعمال بر Working directory
git push <<remoteName>> <<localBranch>> git push origin master	ارسال تغییرات جدید یک شاخه از Local repository به ریپازیتوری ریموت
git push	ارسال تغییرات جدید شاخه اصلی از Local repository به ریپازیتوری ریموت



## اشتراک گذاری شاخه ها

دستور	کاربرد
<code>git branch -r</code>	مشاهده شاخه های ریپازیتوری ریموت
<code>git branch --vv</code>	مشاهده تمامی شاخه ها (remote & local repositories)
<code>git push -u &lt;&lt;remoteName&gt;&gt; &lt;&lt;branchName&gt;&gt; git push -u origin updateFile</code>	ارسال شاخه به ریپازیتوری ریموت
<code>git push -d &lt;&lt;remoteName&gt;&gt; &lt;&lt;branchName&gt;&gt; git push -u origin updateFile</code>	حذف شاخه از ریپازیتوری ریموت





## مدیریت ریپازیتوری های ریموت

دستور	کاربرد
git remote	مشاهده ریپازیتوری های ریموت
git remote add <<wantedName>> <<url>>	اضافه کردن یک ریپازیتوری ریموت با نام دلخواه
git remote rm <<remoteName>>	حذف کردن یک ریپازیتوری ریموت

# 05.

## تمرین و منبع پیشنهادی

Exercise





## تمرین

۱. یک نفر از افراد گروه یک repository داخل گیت هاب به اسم crocodile-git-exercises بسازه و بقیه اعضای گروه رو به اون اضافه کنه
۲. یک فایل html دلخواه رو روی شاخه repository main قرار بدید
۳. هر فرد از افراد گروه از روی شاخه main یک شاخه با نام خودش بسازه و یک تگ به فایل html ای که از شاخه main گرفتین اضافه کنه و یک فایل html دلخواه هم با نام خودش روی شاخه ای که با اسمش ساخته اضافه کنه
۴. یک تغییر توی فایل index.html شاخه اصلی ایجاد کنین (ترجیحا با تغییراتی که توی شاخه های خودتون دادین یکی نباشه)
۵. هر کدوم از افراد شاخه ای که ساخته رو در گیت هاب به شاخه main به وسیله pull request ادغام کنه (تو این مرحله به احتمال زیاد، برای ادغام به conflict میخورین که باید خودتون به صورت گروهی با جستجو برطرفش کنید و نحوه برطرف کردنش رو هم حتما مستند سازی کنید)
۶. در نهایت یک نسخه (release) رو با هر اسم ورژنی که خواستین منتشر کنید

- تمرین به صورت گروهی هست
- پیاده سازی هرگونه روش های best practice در هر کدوم از جزئیات گیت نمره اضافی داره
- نوشتن فایل readme.md برای repository که ساختین نمره اضافی داره (هرچی قشنگتر بنویسین نمره اضافیش بیشتر میشه)
- زمان تحویل تمرین تا تاریخ چهارشنبه ۲۱ شهریور هست
- در صورتی که جایی رو متوجه نشدین، بهتره اول خودتون تو اینترنت بگردین و اگر پیدا نکردین، از پشتیبانی سوال بپرسید
- برای یادگیری بیشتر میتونین از دوره گیت Mosh Hamedani استفاده کنین
- مطمئن باشید که میتونین به بهترین وجه این تمرین رو هم انجام بدین. موفق باشین



# ابتدا مسئله رو حل کن و سپس گذش رو بنویس!

John Johnson,  
Programmer

[www.partschool.ir](http://www.partschool.ir)  
[www.part.institute](http://www.part.institute)  
[www.partsoftware.com](http://www.partsoftware.com)

