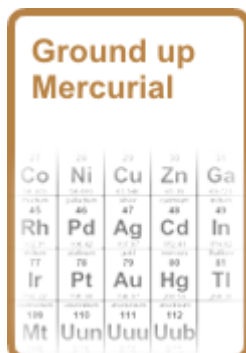


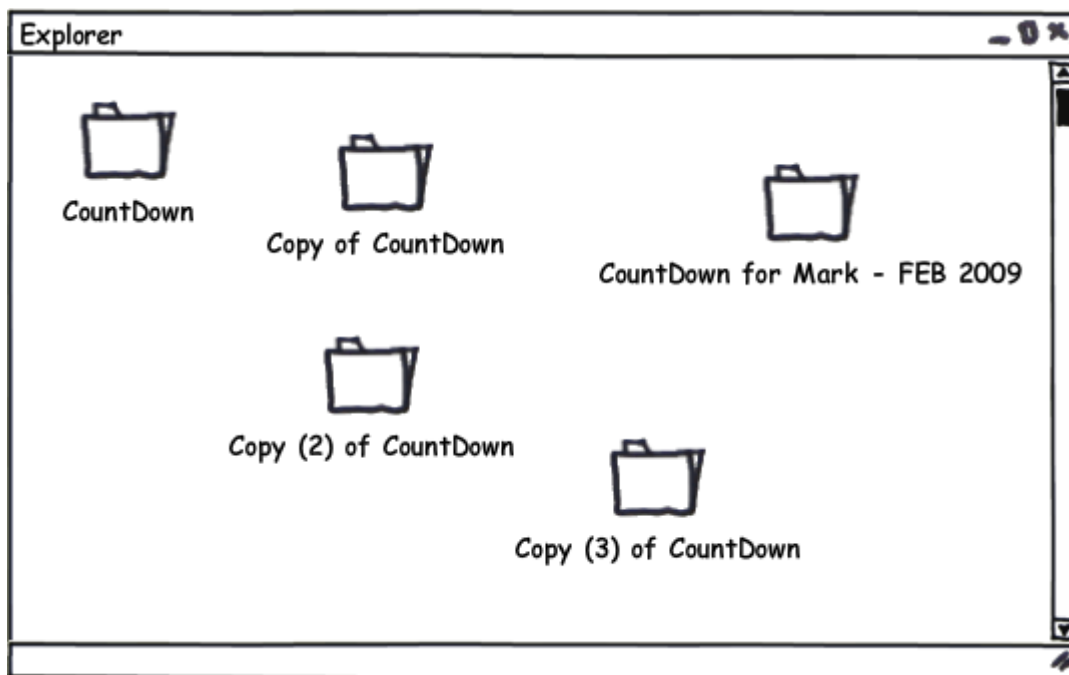
Часть 2. Основы Mercurial



Mercurial — это *система контроля версий*. Разработчики используют ее для администрирования исходного кода. У нее два основных назначения:

1. Она хранит все предыдущие версии каждого файла
2. Она может объединить разные версии вашего кода, то есть сотрудники могут независимо работать над кодом и затем объединять свои изменения

Без Mercurial вы могли бы попытаться сохранить предыдущие версии, просто создавая много копий каталога с кодом:



Это муторно, требует много места на диске и создает путаницу. Лучше использовать систему контроля версий.

Большинство людей работают с Mercurial через интерфейс командной строки. Так можно работать в Windows, Unix, и Mac. Команда для Mercurial — это `hg`:

```
Command Prompt
c:\hginit> hg
Mercurial Distributed SCM

basic commands:

add          add the specified files on the next commit
annotate    show changeset information by line for each file
clone       make a copy of an existing repository
commit      commit the specified files or all outstanding changes
diff        diff repository (or selected files)
export      dump the header and diffs for one or more changesets
forget      forget the specified files on the next commit
init        create a new repository in the given directory
log         show revision history of entire repository or files
merge       merge working directory with another revision
pull        pull changes from the specified source
push        push changes to the specified destination
remove      remove the specified files on the next commit
serve       export the repository via HTTP
status      show changed files in the working directory
summary     summarize working directory state
update      update working directory

use "hg help" for the full list of commands or "hg -v" for details
```

Если выполнить `hg` без параметров, то вы получите список наиболее часто используемых команд. Вы можете также попробовать выполнить `hg help` для получения полного списка команд.

Для того чтобы воспользоваться преимуществами контроля версий, вам нужен *репозиторий*. Репозиторий хранит все предыдущие версии всех ваших файлов. На самом деле, для экономии места на диске, все предыдущие версии не будут храниться — будет храниться только компактный список изменений.

В старые времена занять репозиторий было большим делом. Вам было необходимо иметь центральный сервер и нужно было установить на него программное обеспечение. Mercurial — *распределенная* система, так что вы можете использовать ее без модного центрального сервера. Вы можете пользоваться Mercurial, используя один только свой компьютер. А заполнить репозиторий супер просто: вы просто идете в каталог с вашим кодом...

```
Command Prompt

c:\hginit> cd Countdown

c:\hginit\CountDown> dir /w
Volume in drive C has no label.
Volume Serial Number is 9862-36C5

Directory of c:\hginit\CountDown

[.]                [..]                a.txt
AlternateMessages.xml  App.config           App.xaml
App.xaml.cs         Countdown.xaml       Countdown.xaml.cs
DevDaysCountDown.csproj  favicon.ico         [Images]
[Properties]        [TweetSharp]
                    9 File(s)          155,932 bytes
                    5 Dir(s)  76,083,609,600 bytes free
```

... и выполняете команду `hg init`:

```
Command Prompt

c:\hginit\CountDown> hg init

c:\hginit\CountDown>
```

`hg init`
создает репозиторий.

Минуточку, а разве что-то произошло? Выглядит так, словно ничего не поменялось. Нет, если вы присмотритесь получше, то увидите, что появился новый каталог `.hg`:

```
Command Prompt
c:\hginit\CountDown> dir /w
Volume in drive C has no label.
Volume Serial Number is 9862-36C5

Directory of c:\hginit\CountDown

[.]                [..]                [.hg]
a.txt              AlternateMessages.xml  App.config
App.xaml           App.xaml.cs           Countdown.xaml
CountDown.xaml.cs DevDaysCountDown.csproj favicon.ico
[Images]           [Properties]           [TweetSharp]
                   9 File(s)             155,932 bytes
                   6 Dir(s)  76,083,650,560 bytes free
```

Это и есть репозиторий! Это каталог со всем, что нужно Mercurial для работы. Настройки, предыдущие версии файлов, теги, лишняя пара носков на случай дождя, и прочее. *Не лезьте туда*. Вам почти всегда не стоит возиться с этим каталогом самостоятельно.

Ну хорошо, раз у нас теперь есть свежеспеченный репозиторий, то мы захотим добавить в него все исходные файлы. Это тоже просто: нужно лишь выполнить `hg add`.

```
Command Prompt
c:\hginit\CountDown> hg add
adding AlternateMessages.xml
adding App.config
adding App.xaml
adding App.xaml.cs
adding Countdown.xaml
adding Countdown.xaml.cs
adding DevDaysCountDown.csproj
adding Images\background_city.jpg
adding Images\carsonified_presents.png
adding Images\darkpanel.png
adding Images\devdays.png
adding Images\failwhale.png
adding Images\holding_image.jpg
adding Images\jeff_atwood.jpg
adding Images\joel_spolsky.jpg
adding Images\logo_stackoverflow.png
adding Images\matt_lacey.jpg
adding Images\sideDarkpanel.png
adding Images\vertical_lines2.png
adding Properties\AssemblyInfo.cs
adding Properties\Resources.Designer.cs
adding Properties\Resources.resx
adding Properties\Settings.Designer.cs
adding Properties\Settings.settings
adding TweetSharp\Dimebrain.TweetSharp.dll
adding TweetSharp\Dimebrain.TweetSharp.xml
adding TweetSharp\Newtonsoft.Json.dll
adding a.txt
adding favicon.ico
```

hg add

помечает файлы как запланированные для добавления в репозиторий. Файлы на самом деле не будут добавлены до тех пор, пока вы не зафиксируете изменения.

Остался еще один шаг... вам нужно *зафиксировать* ваши изменения. Какие изменения? Факт добавления всех этих файлов.

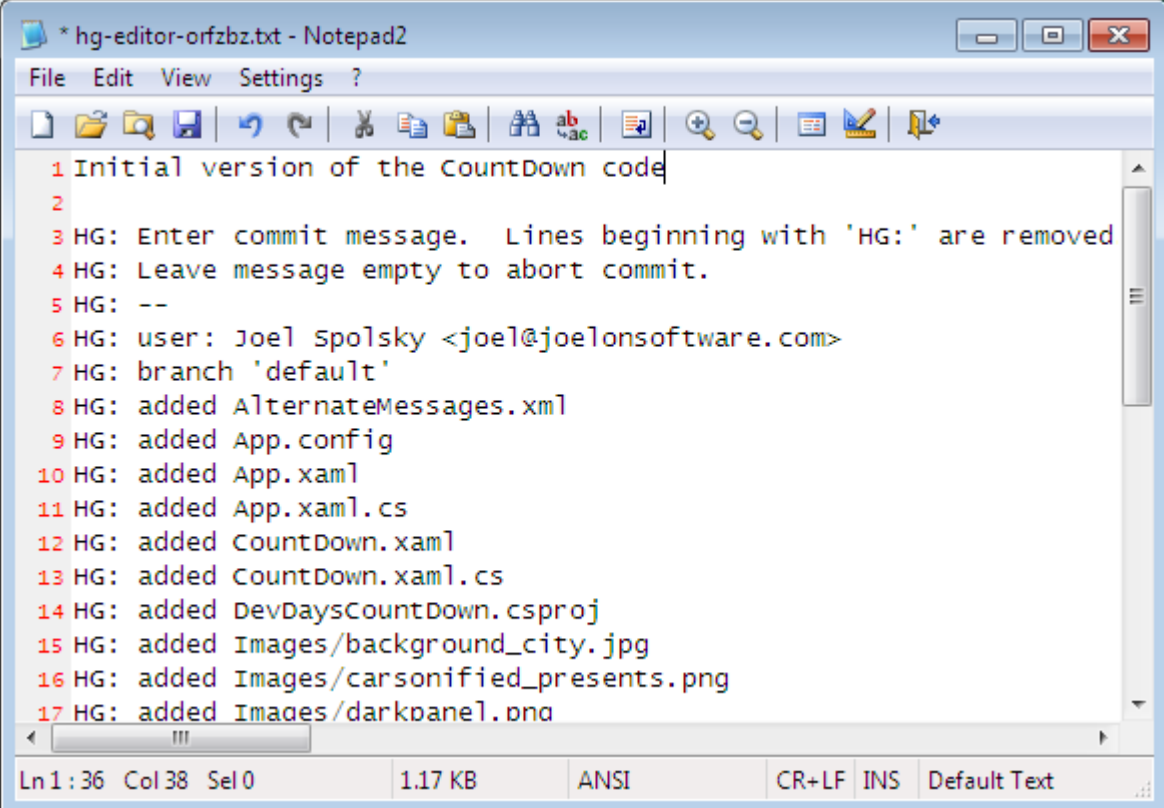
Почему вы вынуждены фиксировать изменения? При работе с Mercurial фиксирование изменений означает: «Эй, вот как файлы сейчас выглядят, так их и запомни, пожалуйста». Это как создание копии всего каталога... каждый раз, когда у вас есть что-то измененное, и оно вам типа нравится, то вы фиксируете изменения.

hg commit

сохраняет текущее состояние всех файлов в репозиторий.

```
Command Prompt
c:\hginit\CountDown> hg commit
```

Mercurial запустит редактор для того, чтобы вы могли напечатать комментарий к изменениям. Вам нужно просто написать что-нибудь, что напомним вам о сделанных изменениях.



The screenshot shows a Notepad2 window titled "hg-editor-orfzbx.txt - Notepad2". The text inside the editor is a commit message for Mercurial. The message starts with "Initial version of the Countdown code" and lists several files that were added to the repository. The status bar at the bottom indicates the cursor is at line 1, column 36, and the file size is 1.17 KB.

```
* hg-editor-orfzbx.txt - Notepad2
File Edit View Settings ?
Initial version of the Countdown code
HG: Enter commit message. Lines beginning with 'HG:' are removed
HG: Leave message empty to abort commit.
HG: --
HG: user: Joel Spolsky <joel@joelonsoftware.com>
HG: branch 'default'
HG: added AlternateMessages.xml
HG: added App.config
HG: added App.xaml
HG: added App.xaml.cs
HG: added Countdown.xaml
HG: added Countdown.xaml.cs
HG: added DevDaysCountDown.csproj
HG: added Images/background_city.jpg
HG: added Images/carsonified_presents.png
HG: added Images/darkpanel.png
Ln1:36 Col38 Sel0 1.17 KB ANSI CR+LF INS Default Text
```

После сохранения и выхода из редактора ваши изменения будут зафиксированы.

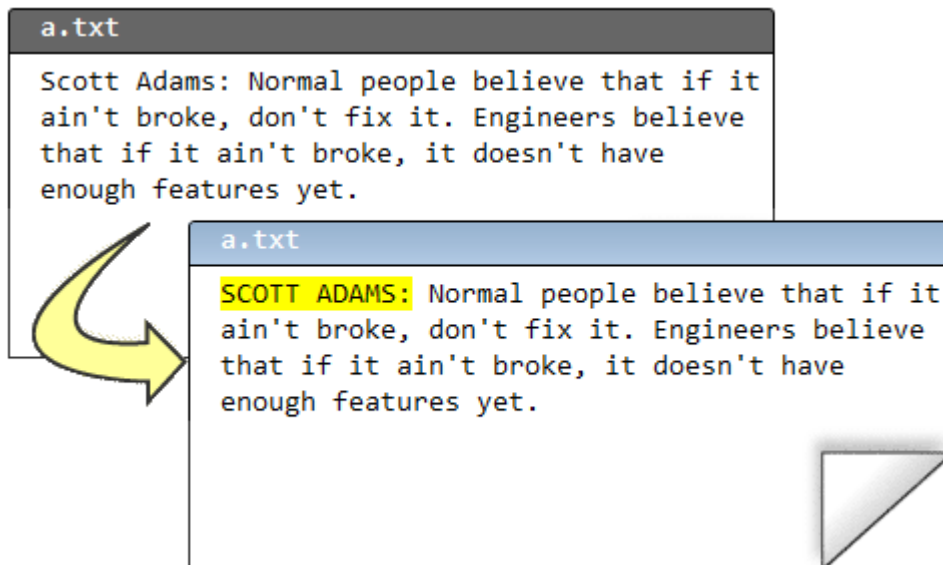
Вы можете выполнить `hg log` для просмотра истории изменений. Это как блог для вашего репозитория:

```
Command Prompt
c:\hginit\CountDown> hg log
changeset: 0:da5f372c3901
tag: tip
user: Joel Spolsky <joel@joelonsoftware.com>
date: Fri Feb 05 13:04:30 2010 -0500
summary: Initial version of the Countdown code
```

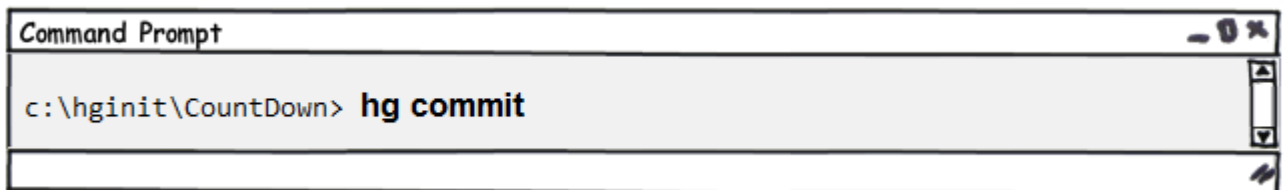
hg log

отображает историю изменений, зафиксированных в репозитории.

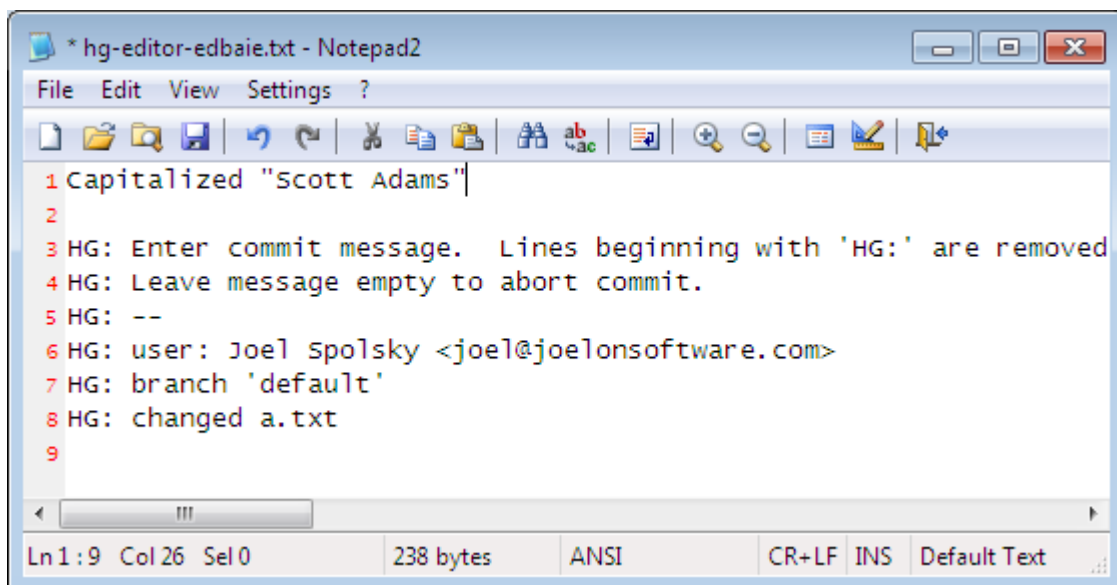
Давайте изменим файл и посмотрим, что получится.



Так как мы сделали изменение, то можем его зафиксировать при помощи `hg commit`:



Обратите внимание, что Mercurial догадался, что только один файл, a.txt, изменен:



А теперь, после того как я закоммитил (зафиксировал изменения), давайте посмотрим на

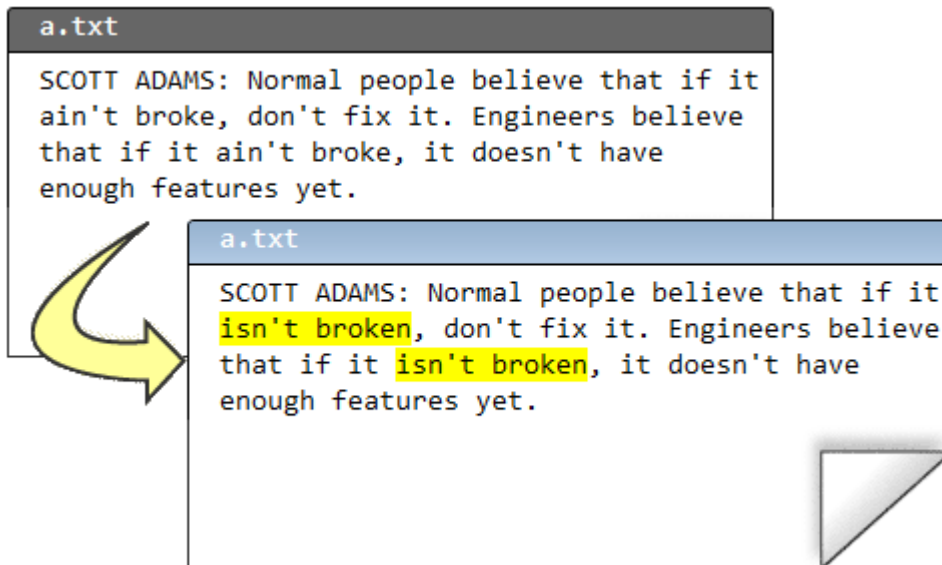
историю:

```
Command Prompt
c:\hginit\CountDown> hg log
changeset: 1:a9497f468dc3
tag:       tip
user:      Joel Spolsky <joel@joelonsoftware.com>
date:      Fri Feb 05 13:26:13 2010 -0500
summary:   Capitalized "Scott Adams"

changeset: 0:da5f372c3901
user:      Joel Spolsky <joel@joelonsoftware.com>
date:      Fri Feb 05 13:04:30 2010 -0500
summary:   Initial version of the Countdown code
```

Как любой нынешний блоггер, Mercurial помещает самое новое в начало.

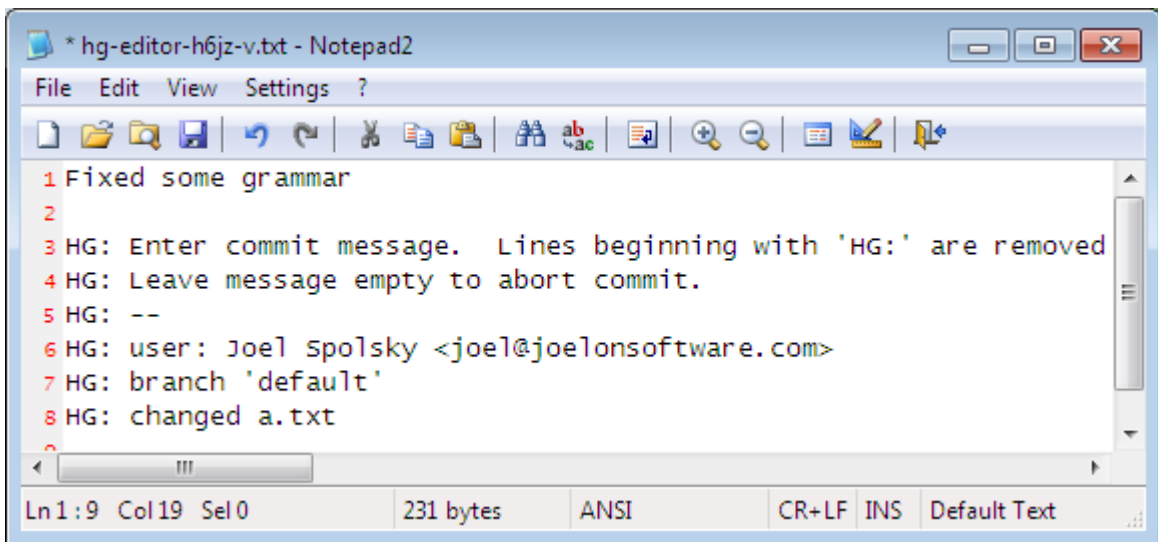
Я собираюсь сделать еще одно изменение, просто чтобы потешить себя.



Коммичу (фиксирую изменения):

```
Command Prompt
c:\hginit\CountDown> hg commit
```

Мое сообщение к коммиту:



```
* hg-editor-h6jz-v.txt - Notepad2
File Edit View Settings ?
1 Fixed some grammar
2
3 HG: Enter commit message. Lines beginning with 'HG:' are removed
4 HG: Leave message empty to abort commit.
5 HG: --
6 HG: user: Joel Spolsky <joel@joelonsoftware.com>
7 HG: branch 'default'
8 HG: changed a.txt
Ln1:9 Col19 Sel0 231 bytes ANSI CR+LF INS Default Text
```

А что у нас теперь в истории?



```
Command Prompt
c:\hginit\CountDown> hg log
changeset: 2:55490459b740
tag: tip
user: Joel Spolsky <joel@joelonsoftware.com>
date: Fri Feb 05 13:47:43 2010 -0500
summary: Fixed some grammar

changeset: 1:a9497f468dc3
user: Joel Spolsky <joel@joelonsoftware.com>
date: Fri Feb 05 13:26:13 2010 -0500
summary: Capitalized "Scott Adams"

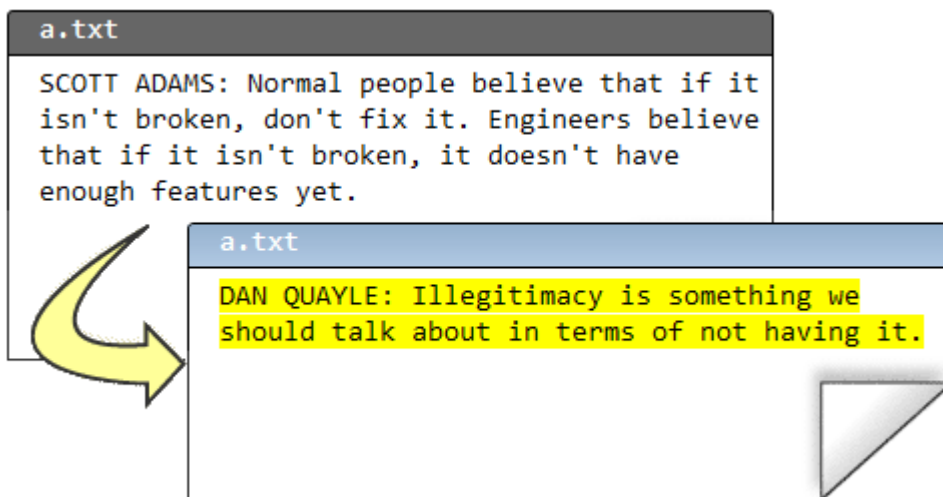
changeset: 0:da5f372c3901
user: Joel Spolsky <joel@joelonsoftware.com>
date: Fri Feb 05 13:04:30 2010 -0500
summary: Initial version of the Countdown code
```

Хорошо, это было весело. Я сделал несколько изменений и каждый раз, сделав значимое изменение, я фиксировал его в репозитории.

Я знаю, вы сейчас думаете: «ДЖОЭЛЬ, ВСЕ ЭТО ВЫГЛЯДИТ ПУСТОЙ ТРАТОЙ ВРЕМЕНИ». К чему вся эта катавасия с коммитами?

Терпение, мой юный друг. Ты вот-вот узнаешь, как извлечь выгоду из всего этого.

Для начала: скажем, вы сделали большую ошибку в процессе редактирования.



А затем, боже ж мой, в дополнение ко всему вы удалили пару очень важных файлов.

```
Command Prompt
c:\hginit\CountDown> del favicon.ico
c:\hginit\CountDown> del App.xaml
```

Во времена, когда не было Mercurial, это все могло бы стать неплохим поводом для похода к системному администратору. Там, со слезами на глазах, вы бы задали ему пронзительно грустный вопрос: «Почему система резервного копирования „временно“ не работает все последние восемь месяцев».

Системный администратор, которого все зовут [Тако](#), слишком скромн и не обедает с остальной командой. В тех редких случаях, когда он не сидит в своем кресле на колесиках, можно увидеть треугольное пятно цвета [сальсы](#) там, где падали, пролетев между ног, капли от его мексиканских закусок. Эти капли гарантируют, что никто не возьмет его кресло, хотя это [одно из превосходных кресел от Herman Miller](#), купленных учредителями компании для себя любимых, а не обычное бюджетное офисное нечто, из-за которого у всех болят спины.

В любом случае, резервной копии нет, ага.

Благодаря Mercurial, если вам не нравится то, что вы сделали, вы можете выполнить удобную команду `hg revert`, которая немедленно вернет ваш каталог к виду, в котором он был в момент последнего коммита.

```
Command Prompt
c:\hginit\CountDown> hg revert --all
reverting App.xaml
reverting a.txt
reverting favicon.ico

c:\hginit\CountDown> type a.txt
SCOTT ADAMS: Normal people believe that if it isn't
broken, don't fix it. Engineers believe that if it
isn't broken, it doesn't have enough features yet.
```

`hg revert`

возвращает измененные файлы к зафиксированному в репозитории виду.

Я использовал аргумент `--all`, потому что хотел вернуть *все* файлы к прежнему состоянию.

Таким образом, когда вы работаете над исходным кодом, используя Mercurial, вы:

1. Делаете изменения
2. Оцениваете, подходят ли они
3. Если подходят, то выполняете `commit`
4. Если не подходят, то выполняете `revert`
5. GOTO 1

(Я знаю. Пользуюсь командной строкой, в Windows, да еще и оператором GOTO — я *самый стрёмный программист* из всех, когда-либо живших.)

Со временем вы можете забыть, на чем остановились и что поменяли с момента прошлого коммита. Mercurial отслеживает все это для вас. Все что вам нужно, это выполнить `hg status` и Mercurial даст вам список измененных файлов.

`hg status`

отображает список измененных файлов.

Предположим, я создал один файл, отредактировал другой и удалил третий.

```
Command Prompt

c:\hginit\CountDown> copy a.txt b.txt
1 file(s) copied.

c:\hginit\CountDown> notepad2 a.txt

c:\hginit\CountDown> del favicon.ico

c:\hginit\CountDown> hg status
M a.txt
! favicon.ico
? b.txt
```

hg status отображает список измененных файлов, добавляя значок в начале каждой строки. Этот значок сообщает о том, что же произошло. «M» означает «Modified» — файл был изменен. «!» означает отсутствие — файл должен быть здесь, но куда-то делся. «?» означает что, состояние не определено — Mercurial ничего не знает про этот файл. Пока не знает.

Давайте разбираться с изменениями поочередно. Вот что поменялось в файле **a.txt**? Вы ведь можете забыть, что изменили. Чёрт, да я едва помню что я обычно ем на завтрак. А особенно внушает опасения то, что это всегда хрустящие колечки [CHEERIOS](#). В любом случае, a.txt изменен. Что поменялось?

А есть команда для этого: hg diff сообщит вам в точности, что произошло с файлом с прошлого коммита.

hg diff
показывает, что изменилось в файле.

```
Command Prompt

c:\hginit\CountDown> hg diff a.txt
diff -r 55490459b740 a.txt
--- a/a.txt      Fri Feb 05 13:47:43 2010 -0500
+++ b/a.txt      Fri Feb 05 14:31:18 2010 -0500
@@ -1,3 +1,3 @@
-SCOTT ADAMS: Normal people believe that if it isn't
+SCOTT ADAMS: Civilians believe that if it isn't
  broken, don't fix it. Engineers believe that if it
  isn't broken, it doesn't have enough features yet.
```

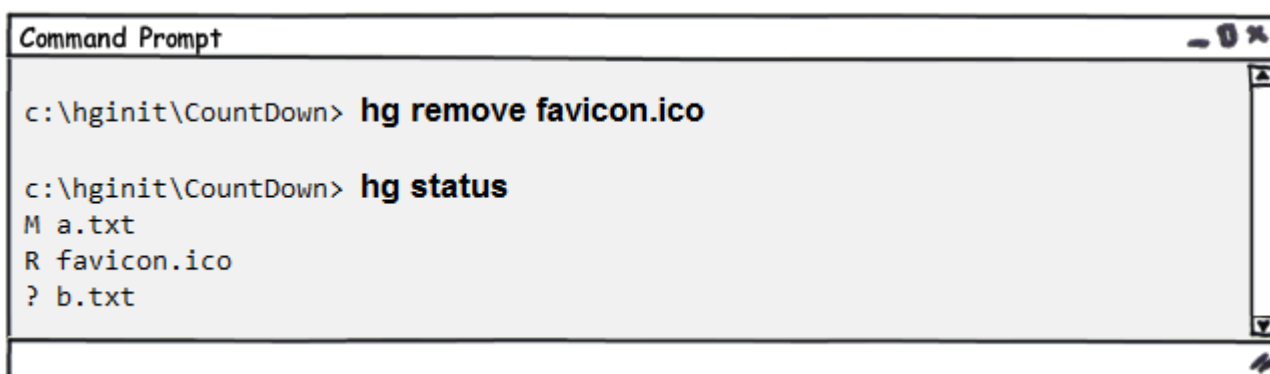
Формат результата немного непонятный, но самое важное, что вы можете видеть строки, начинающиеся с минуса (это то, что было удалено), и строки, начинающиеся с плюса (это

то, что добавлено). Таким образом, вы можете видеть, что «Normal people» было заменено на «Civilians».

Теперь про тот пропавший файл **favicon.ico**. Как уже было сказано, если вы на самом деле не планировали его удалять, то можете выполнить команду `hg revert` для восстановления. Но, предположим, что вы на самом деле хотели его удалить. При каждом удалении (или добавлении) файла вы обязаны сообщить об этом событии Mercurial.

```
hg remove
```

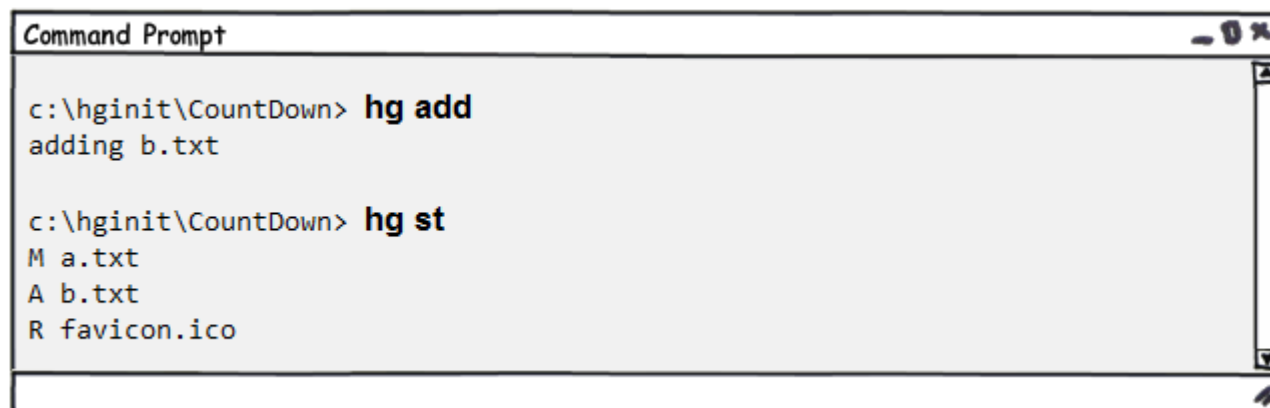
помечает файлы как запланированные для удаления из репозитория. Файлы на диске не будут удалены до тех пор, пока вы не зафиксируете изменения.



```
Command Prompt
c:\hginit\CountDown> hg remove favicon.ico

c:\hginit\CountDown> hg status
M a.txt
R favicon.ico
? b.txt
```

«R» означает «Removed», то есть, помечен для удаления. Во время следующего коммита Mercurial удалит этот файл. (*История* для этого файла сохранится в репозитории, так что, конечно, мы всегда сможем получить его назад.) И, наконец, нужно добавить этот новый файл **b.txt**:



```
Command Prompt
c:\hginit\CountDown> hg add
adding b.txt

c:\hginit\CountDown> hg st
M a.txt
A b.txt
R favicon.ico
```

«A» означает «Added», то есть, помечен для добавления. Заметили, что я разленился писать `hg status` каждый раз? Для Mercurial достаточно минимального набора букв, однозначно определяющего команду, а на `st` начинается только одна команда.

Разобравшись с вопросиками и восклицательными знаками, я могу продолжить и внести изменения:

```
Command Prompt

c:\hginit\CountDown> hg commit

c:\hginit\CountDown> hg log
changeset: 3:2f4718ee168e
tag:       tip
user:      Joel Spolsky <joel@joelonsoftware.com>
date:      Fri Feb 05 14:54:45 2010 -0500
summary:   A few highly meaningful changes. No favicon.ico no more.

changeset: 2:55490459b740
user:      Joel Spolsky <joel@joelonsoftware.com>
date:      Fri Feb 05 13:47:43 2010 -0500
summary:   Fixed some grammar

changeset: 1:a9497f468dc3
user:      Joel Spolsky <joel@joelonsoftware.com>
date:      Fri Feb 05 13:26:13 2010 -0500
summary:   Capitalized "Scott Adams"

changeset: 0:da5f372c3901
user:      Joel Spolsky <joel@joelonsoftware.com>
date:      Fri Feb 05 13:04:30 2010 -0500
summary:   Initial version of the Countdown code
```

На что еще стоит обратить внимание в выводе `hg log`: строка *changeset* отображает номер каждого коммита. На самом деле даже два номера: короткий удобный вроде «0» для первой ревизии и длинный непонятный шестнадцатеричный, на который вы можете пока не обращать внимания.

Запомните, что Mercurial хранит в репозитории достаточно информации для воссоздания любой версии любого файла.

Прежде всего, при помощи простой команды `hg cat` вы можете вывести содержимое любой версии любого файла в консоль. К примеру, вот как увидеть, что сейчас находится в файле `a.txt`:

```
Command Prompt

c:\hginit\CountDown> hg cat a.txt
SCOTT ADAMS: Civilians believe that if it isn't
broken, don't fix it. Engineers believe that if it
isn't broken, it doesn't have enough features yet.
```

`hg cat`
отображает содержимое любого файла для любой ревизии.

Для того чтобы увидеть как файл выглядел раньше, я могу просто указать нужный номер набора изменений (changeset из лога) при помощи аргумента `-r` («revision», то есть ревизия):

```
Command Prompt
c:\hginit\CountDown> hg cat -r 0 a.txt
Scott Adams: Normal people believe that if it ain't
broke, don't fix it. Engineers believe that if it
ain't broke, it doesn't have enough features yet.
```

Если у файла сложное содержимое или большая длина, а изменился он лишь немного, то я могу использовать команду `hg diff` с аргументом `r` для вывода разницы между двумя ревизиями. К примеру, вот как посмотреть разницу между ревизиями 0 и 1:

```
Command Prompt
c:\hginit\CountDown> hg diff -r 0:1 a.txt
diff -r da5f372c3901 -r a9497f468dc3 a.txt
--- a/a.txt      Fri Feb 05 13:04:30 2010 -0500
+++ b/a.txt      Fri Feb 05 13:26:13 2010 -0500
@@ -1,3 +1,3 @@
-Scott Adams: Normal people believe that if it ain't
+SCOTT ADAMS: Normal people believe that if it ain't
 broke, don't fix it. Engineers believe that if it
 ain't broke, it doesn't have enough features yet.
```

И наконец, - надеюсь, что вы еще не свалились от усталости, - прежде чем закончить эту часть, я хочу показать вам еще *одну маленькую фишку*: при помощи команды `hg update` вы можете перемещаться вперед и назад ко времени создания любой ревизии. Ну, *по сути* в будущее вы перемещаться не сможете, хотя это было бы супер круто, да. Вы, имея всего четыре ревизии, смогли бы выполнить `hg update -r 103994` и получить реально крутую версию ваших исходников. С гравипапой и прочими футуристическими штучками. Но, конечно, это невозможно.

Что на самом деле возможно, так это вернуться к любой ревизии. Следите за руками:

```
Command Prompt
c:\hginit\CountDown> hg update -r 0
2 files updated, 0 files merged, 1 files removed, 0 files unresolved

c:\hginit\CountDown> type a.txt
Scott Adams: Normal people believe that if it ain't
broke, don't fix it. Engineers believe that if it
ain't broke, it doesn't have enough features yet.

c:\hginit\CountDown> hg up -r 1
1 files updated, 0 files merged, 0 files removed, 0 files unresolved

c:\hginit\CountDown> type a.txt
SCOTT ADAMS: Normal people believe that if it ain't
broke, don't fix it. Engineers believe that if it
ain't broke, it doesn't have enough features yet.

c:\hginit\CountDown> hg up
2 files updated, 0 files merged, 1 files removed, 0 files unresolved

c:\hginit\CountDown> type a.txt
SCOTT ADAMS: Civilians believe that if it isn't
broken, don't fix it. Engineers believe that if it
isn't broken, it doesn't have enough features yet.
```

hg update

приводит рабочий каталог к состоянию, как у заданной ревизии.

Фактически, для перемещения вперед и назад между ревизиями `hg update` вносит в файлы запомненные изменения. Если файл был добавлен или удален, то эта команда добавляет или удаляет его на диске. Выполнение `hg update` без дополнительных параметров приводит рабочий каталог к состоянию как у самой свежей ревизии.

Проверь себя

Отлично! Эта часть закончена. Вот все, что вы должны уметь делать на данный момент:

1. Создать репозиторий
2. Добавить или удалить файлы из репозитория
3. После внесения изменений посмотреть, что было сделано, а затем
4. ... зафиксировать изменения, если они вам нравятся
5. ... или откатить изменения, если не нравятся
6. Посмотреть предыдущие версии файлов или привести рабочий каталог к состоянию как у некоторой ревизии.