

Основні команди LINUX

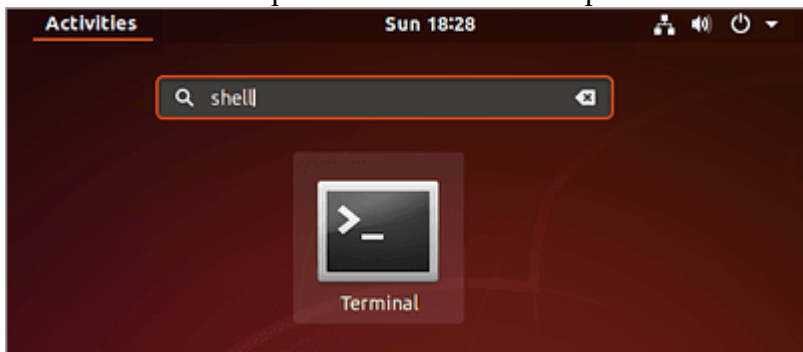
1. [Повний список команд Linux](#)
2. [Основні команди Linux](#)
3. [Команди Linux для відображення дати, часу, календаря](#)
4. [Команди для навігації файловою системою Linux](#)
5. [Інші корисні команди Linux](#)
6. [Висновок](#)
7. [Рекомендована література](#)

Linux належить до Unix-подібного сімейства ОС. Операційні системи Linux засновані на ядрі Linux. Усі команди Linux виконуються в терміналі, наданому ОС Linux. Термінал дуже схожий на командний рядок Windows. Його також називають оболонкою або консоллю Linux. Інтерфейс користувача терміналу Linux допомагає виконувати всі адміністративні операції, включаючи встановлення пакунків, маніпулювання файлами, управління користувачами.

Усі команди Linux чутливі до регістру . Команди виконуються при натисканні клавіші Enter після введення команди. Результат команди буде показаний у самому терміналі.

Linux включає багато команд. У цій шпаргалці команд Linux ми проведемо вас через найважливіші, щоб ви ознайомилися з ними та могли легко працювати на терміналі Linux.

Для виконання команд вам знадобиться комп'ютер під управлінням Ubuntu або будь-яким іншим дистрибутивом Linux. В Ubuntu ви можете запустити термінал, натиснувши кнопку Ctrl-Alt-T або натиснувши кнопку «Діяльність» у верхньому лівому куті екрана та ввівши «оболонка» або «термінал» або інші поширені синоніми.



Ads by optAd360

Якщо у вас немає ОС Linux, але ви хочете відпрацювати команди Linux, ви можете використовувати будь-який онлайн-термінал Linux. Багато терміналів Linux доступні в Інтернеті безкоштовно для відпрацювання команд Linux.

Що ви дізнаєтесь:

1. Основні команди Linux

1) pwd: Команда `pwd` друкує поточний робочий каталог. Це дасть вам абсолютний шлях, тобто від кореня до поточного каталогу, в якому ви знаходитесь. Ця команда допомагає уникнути загублення у вікні терміналу. Ви будете знати, в якому каталозі ви знаходитесь. За замовчуванням поточний робочий каталог є домашнім каталогом, коли ви вперше відкрили термінал.

Команда: `pwd`

```
-sh-4.1$ pwd
/home/cpf657/kaushapx
-sh-4.1$
```

2) **whoami**: Ця команда відобразить ім'я користувача для користувача, який наразі ввійшов до терміналу Linux.

Команда: *хто я*

```
-sh-4.1$ whoami
kaushapx
-sh-4.1$
```

3) **ясно**: Коли термінал заповнюється великою кількістю команд та їх виведенням, ви можете використовувати команду `clear`, щоб очистити термінал.

4) **вихід**: команда виходу просто закриє сеанс терміналу.

Команди Linux для відображення дати, часу, календаря

1) **дата**: Ця команда відображає поточний час і дату.

Ads by **optAd360**

Команда: *дата*

```
1 date

Result

$bash -f main.sh
Mon Mar 23 09:27:18 UTC 2020
```

2) **кал**: кал означає календар. За замовчуванням команда `cal` відображає календар поточного місяця. На момент написання цієї статті поточним місяцем був березень 2020 року, а отже, у вихідних даних ви бачите, що він відобразив календар березня 2020 року.

Команда: *кал*

```
1 cal

Result

$bash -f main.sh
    March 2020
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

- Якщо ви хочете відобразити календар усього поточного року, ви можете дати команду «**Cal - у**» .
- Ви також можете вказати рік. **Наприклад**, якщо ви хочете побачити календар 2019 року, ви можете дати команду '**Cal 2019**' - Тут відобразиться календар 2019 року.
- Крім того, якщо ви хочете переглянути календар конкретного місяця певного року, скажімо, січень 2019 року, ви можете виконати команду '**Cal 1 2019**' - це покаже календар січня 2019 року. Тут «1» позначає перший місяць року, тобто січень.

- Ви також можете побачити календар трьох місяців разом - попереднього місяця, поточного місяця та наступного місяця. Команда для цієї операції: **'Cal -3'**

```
1 cal -3
```

Result

```
$bash -f main.sh
```

February 2020							March 2020							April 2020						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
						1	1	2	3	4	5	6	7				1	2	3	4
2	3	4	5	6	7	8	8	9	10	11	12	13	14	5	6	7	8	9	10	11
9	10	11	12	13	14	15	15	16	17	18	19	20	21	12	13	14	15	16	17	18
16	17	18	19	20	21	22	22	23	24	25	26	27	28	19	20	21	22	23	24	25
23	24	25	26	27	28	29	29	30	31					26	27	28	29	30		

Команди для навігації файловою системою Linux

Файлова система Linux має деревоподібну структуру, яку також називають деревом каталогів. Він використовується для упорядкування інформації.

Дерево каталогів дуже схоже на генеалогічне дерево, тому це каталог або файл, що має батьківський статус. Однак кожен каталог або файл має рівно одного батька.

Тепер ми подивимось, як структурована файлова система Linux. По-перше, у нас є кореневий каталог, який є першим або найпопулярнішим каталогом у нашому дереві. Він представлений косою рисою (/). Під коренем ви знайдете багато підкаталогів.

Давайте розглянемо деякі важливі підкаталоги:

1. **Am** : Цей каталог містить виконувані програми та команди, якими можуть користуватися всі користувачі системи.
2. **Вибрати**: Це означає необов'язковий. Цей каталог містить комерційні програмні продукти, які не встановлені за замовчуванням у вашій системі. **Наприклад**, якщо ви використовуєте ubuntu, ви, мабуть, помітили, що Google Chrome за замовчуванням не встановлений у вашій системі, і якщо ви хочете встановити Google Chrome, ви знайдете його під каталогом Opt.
3. **Температура**: temp означає тимчасовий. У цьому каталозі ви знайдете тимчасові файли.
4. **Де**: Він містить змінні дані, які часто змінюються з часом. Сюди входять файли журналів, пули електронної пошти та бази даних користувачів.
5. **Головна Linux** - це багатокористувацьке середовище, що означає, що багато користувачів можуть отримати доступ до системи одночасно. Кожному користувачеві призначено дістичне харчування всередині домашнього каталогу.
- 6.

Команди, які допоможуть орієнтуватися в дереві каталогів.

1. **pwd**: Ми вже детально обговорювали цю команду вище.
2. **ls**: ls означає список.

Синтаксис команди ls такий:

Ця команда відображає вміст каталогу. Якщо ви просто виконаєте команду ls, не вказавши жодних параметрів та файлів, вона перелічить в алфавітному порядку всі каталоги та файли, що знаходяться під вашим поточним робочим каталогом.

```
$ ls
```

Output

cache db empty games lib local lock log mail opt run spool tmp

Ви можете використовувати команду **ls** з різними опціями та аргументами:

а) **ls -l** : Тут відображається вміст із докладним переліком. -l означає довгий варіант.

б) **ls -lh**: Він використовується для відображення зручних для людини розмірів файлів. -h означає люди.

в) **ls -lha** :Ця команда використовується для відображення всіх прихованих файлів. -a включає приховані файли.

Приклад для цих команд:

```
-sh-4.1$ ls
0 console.txt lock_1.txt lockfile.txt test1
-sh-4.1$ ls -l
total 32
-rw-rw-r-- 1 kaushapx acsgrp 0 Jan 9 2017 0
-rw-r--r-- 1 kaushapx acsgrp 816 Apr 19 22:18 console.txt
-rw-r--r-- 1 kaushapx acsgrp 8 Jan 6 2017 lock_1.txt
-rw-r--r-- 1 kaushapx acsgrp 8 Jan 6 2017 lockfile.txt
drwxr-xr-x 2 kaushapx acsgrp 1024 Apr 20 05:00 test1
-sh-4.1$ ls -lh
total 32K
-rw-rw-r-- 1 kaushapx acsgrp 0 Jan 9 2017 0
-rw-r--r-- 1 kaushapx acsgrp 816 Apr 19 22:18 console.txt
-rw-r--r-- 1 kaushapx acsgrp 8 Jan 6 2017 lock_1.txt
-rw-r--r-- 1 kaushapx acsgrp 8 Jan 6 2017 lockfile.txt
drwxr-xr-x 2 kaushapx acsgrp 1.0K Apr 20 05:00 test1
-sh-4.1$ ls -lha
total 48K
drwxr-xr-x 3 kaushapx acsgrp 1.0K Apr 20 05:00 .
drwxr-xr-x 5 kaushapx acsgrp 6.0K Apr 19 21:40 ..
-rw-rw-r-- 1 kaushapx acsgrp 0 Jan 9 2017 0
-rw-r--r-- 1 kaushapx acsgrp 816 Apr 19 22:18 console.txt
-rw-r--r-- 1 kaushapx acsgrp 8 Jan 6 2017 lock_1.txt
-rw-r--r-- 1 kaushapx acsgrp 8 Jan 6 2017 lockfile.txt
drwxr-xr-x 2 kaushapx acsgrp 1.0K Apr 20 05:00 test1
-sh-4.1$
```

Ads by **optAd360**

г) **ls /**: Оскільки / представляє кореневий каталог, ця команда покаже всі файли та папки, що знаходяться всередині кореневого каталогу.

д) **ls ~**: Оскільки ~ представляє домашній каталог, то ця команда відобразить вміст вашого домашнього каталогу.

Команда: **ls ~**

```
-sh-4.1$ ls ~
comment_body.tgp console.txt pclwip test test1
-sh-4.1$
```

ф) **ls ..**: Ця команда відобразить вміст батьківського каталогу, оскільки .. представляє батьківський каталог. Припустимо, право пось я перебуваю в тестовому каталозі всередині мого домашнього каталогу. Якщо я виконаю команду **ls ..** тут, то вона відобразить вміст батьківського каталогу тесту, який є моїм домашнім каталогом.

Ads by **optAd360**

Команди, виконані в наведеному нижче прикладі:

ls
cd-тест
pwd
ls ..

```
-sh-4.1$ ls ~  
comment_body.tgp  console.txt  pclwip  test  test1  
-sh-4.1$ cd test  
-sh-4.1$ pwd  
/home/cpf657/kaushapx/test  
-sh-4.1$ ls ..  
comment_body.tgp  console.txt  pclwip  test  test1  
-sh-4.1$
```

g) Ви також можете перерахувати вміст файлу, вказавши його абсолютну назву шляху.

Команда: `ls / home / cpf657 / kaushapx / test / parent`

Ads by optAd360

```
-sh-4.1$ ls /home/cpf657/kaushapx/test/parent  
child  directory1  directory2  
-sh-4.1$
```

Так само ви можете використовувати відносну назву шляху.

(iii) **cd:** Ще однією корисною командою для навігації по файловій системі є `cd`. `cd` означає каталог змін.

Припустимо, я зараз перебуваю у своєму домашньому каталозі і хочу перейти до каталогу з назвою „test”, який присутній у моєму домашньому каталозі, тому я хочу використувати **cd-тест c ommand:**

pwd
cd-тест
pwd

Ads by optAd360

```
-sh-4.1$ pwd  
/home/cpf657/kaushapx  
-sh-4.1$ cd test  
-sh-4.1$ pwd  
/home/cpf657/kaushapx/test  
-sh-4.1$
```

Щоб перейти на рівень вище, вам потрібно виконати **компакт-диск .. команда:**

pwd
Компакт-диск
pwd

```
-sh-4.1$ pwd  
/home/cpf657/kaushapx/test  
-sh-4.1$ cd ..  
-sh-4.1$ pwd  
/home/cpf657/kaushapx  
-sh-4.1$
```

Ads by optAd360

Ви також можете перейти до будь-якого каталогу, використовуючи його абсолютну назву шляху. **Наприклад,** припустимо, що зараз я перебуваю у тестовому каталозі під своїм домом. Тепер я хочу перейти в інший каталог, який називається `test1`, який присутній у моєму

домашньому каталозі. Отже, я буду використовувати команду `cd` разом із абсолютним шляхом до каталогу `test1`.

Команда:

`pwd`

`cd /home/cpf657/kaushapx/test1`

`pwd`

```
-sh-4.1$ pwd
/home/cpf657/kaushapx/test
-sh-4.1$ cd /home/cpf657/kaushapx/test1
-sh-4.1$ pwd
/home/cpf657/kaushapx/test1
-sh-4.1$
```

`ad`

Є багато варіантів, які ми можемо використовувати з командою `cd`.

CD /: Перший варіант, який є дуже корисним, - це пробіл уперед. Отже, як ви знаєте, що `Slash` представляє кореневий каталог, **CD /** перенесе вас безпосередньо до кореневого каталогу.

Команда:

`pwd`

`CD`

`pwd`

```
-sh-4.1$ pwd
/home/cpf657/kaushapx/test1
-sh-4.1$ cd /
-sh-4.1$ pwd
/
-sh-4.1$ ls
acs      collection-new  elsabin  fpdf  icce-app  java.ps  l-n      mnt  nfd2  proc  selinux  sys      transformat
bin      dev            etc      fpdf2  icce-data  jout     lost+found  net  nfs   release  services  sys      usr
boot     dsar          fab      glp    icce_data_share  lib      media     nfd  nj    root    services-new  transformation  var
collection  elsa          fpdf     home  info-data  lib64    misc      nfd1 opt  sbin   srv      services-new  transformation-local  xml
-sh-4.1$
```

cd ~: Другий варіант, який ви можете використовувати, - це `CD Tilda`, тобто `cd ~`. Це призведе до домашнього каталогу, як ви можете бачити нижче скріншот.

Команда:

`cd-тест`

`pwd`

`cd`

`pwd`

```
-sh-4.1$ cd test
-sh-4.1$ pwd
/home/cpf657/kaushapx/test
-sh-4.1$ cd ~
-sh-4.1$ pwd
/home/cpf657/kaushapx
```

компакт-диск ..: Іншим варіантом, який ви можете використовувати з командою `cd`, є `cd` дві крапки, тобто `cd ..`

Ads by optAd360

Дві крапки представляють батьківський каталог, тому ви перейдете на крок назад у дереві каталогів за допомогою `cd ..` **команда:**

компакт-диск

`pwd`

```
/home/cpf657/kaushapx/test/parent/child
-sh-4.1$ cd ..
-sh-4.1$ pwd
/home/cpf657/kaushapx/test/parent
-sh-4.1$
```

Якщо ви наберете `cd` дві крапки знову і знову і знову, зрештою, ви перейдете до свого кореневого каталогу.

Якщо ви хочете перейти в інший каталог всередині батьківського каталогу, ви також можете використовувати цю команду, як показано нижче:

Ads by optAd360

```
pwd
```

```
cd ../directory1
```

```
pwd
```

```
-sh-4.1$ pwd
/home/cpf657/kaushapx/test/parent/directory2
-sh-4.1$ cd ../directory1
-sh-4.1$ pwd
/home/cpf657/kaushapx/test/parent/directory1
-sh-4.1$ █
```

компакт-диск - команда `cd space dash` переводить вас до попереднього робочого каталогу.

Інші корисні команди Linux

(i) **w**: Ця команда відображатиме інформацію про зареєстрованих користувачів.

Команда: w

```
ad
```

```
-sh-4.1$ w
 04:28:38 up 57 days, 15:06,  1 user,  load average: 0.21, 0.22, 0.24
USER      TTY      FROM          LOGIN@      IDLE   JCPU   PCPU WHAT
kaushapx pts/0    172.29.192.3   04:04       0.00s  0.04s  0.01s w
-sh-4.1$ █
```

(ii) **невідомий**: Ця команда відображає інформацію про комп'ютер Linux, на якому ви працюєте. Існують різні варіанти, за допомогою яких ви можете використовувати цю команду.

- **uname -a**: Для відображення всієї інформації про комп'ютер Linux.

Команда: uname -a

```
-sh-4.1$ uname -a
Linux src4713 2.6.32-754.25.1.el6.x86_64 #1 SMP Mon Dec 23 15:19:53 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
-sh-4.1$ █
```

- **uname -s**: Для відображення типу ядра.

Команда: uname -s

Ads by optAd360

```
-sh-4.1$ uname -s
Linux
-sh-4.1$ █
```

- **uname -r**: Відображення інформації про випуск ядра.

Команда: uname -r

```
-sh-4.1$ uname -r
2.6.32-754.25.1.el6.x86_64
-sh-4.1$ █
```

- **uname -v**: Відображення версії ядра.

Команда: uname -v

```
-sh-4.1$ uname -v
#1 SMP Mon Dec 23 15:19:53 UTC 2019
-sh-4.1$ █
```

(iii) **зверху**: Ця команда відображатиме дані про вашу машину Linux у режимі реального часу. Він показує підсумок стану, скільки часу працює комп'ютер, середнє навантаження, кількість

запущених завдань, інформацію про процесор, інформацію про пам'ять, стан процесів тощо. Вам потрібно натиснути Q, щоб вийти з верхньої команди.

ad

Команда: *зверху*

```
-sh-4.1$ top
top - 05:01:14 up 57 days, 15:39, 1 user, load average: 0.34, 0.29, 0.26
Tasks: 635 total, 1 running, 634 sleeping, 0 stopped, 0 zombie
Cpu(s): 3.1%us, 3.1%sy, 0.0%ni, 93.8%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 65957544k total, 18329060k used, 47628484k free, 528608k buffers
Swap: 67108856k total, 0k used, 67108856k free, 10907792k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
19027	kaushapx	20	0	19660	1828	1048	R	94.3	0.0	0:00.53	top
1	root	20	0	19364	1552	1228	S	0.0	0.0	1:11.75	init
2	root	20	0	0	0	0	S	0.0	0.0	0:04.04	kthreadd
3	root	RT	0	0	0	0	S	0.0	0.0	0:13.44	migration/0
4	root	20	0	0	0	0	S	0.0	0.0	1:14.07	ksoftirqd/0
5	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	stopper/0
6	root	RT	0	0	0	0	S	0.0	0.0	0:04.29	watchdog/0
7	root	RT	0	0	0	0	S	0.0	0.0	0:14.26	migration/1
8	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	stopper/1
9	root	20	0	0	0	0	S	0.0	0.0	1:30.36	ksoftirqd/1
10	root	RT	0	0	0	0	S	0.0	0.0	0:03.91	watchdog/1
11	root	RT	0	0	0	0	S	0.0	0.0	0:05.09	migration/2
12	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	stopper/2
13	root	20	0	0	0	0	S	0.0	0.0	0:32.56	ksoftirqd/2
14	root	RT	0	0	0	0	S	0.0	0.0	0:03.63	watchdog/2
15	root	RT	0	0	0	0	S	0.0	0.0	0:06.05	migration/3

ad

(iv) смола: tar - це стрічковий архів. Ця команда використовується для створення архівного файлу, що складається з багатьох інших файлів, або також може використовуватися для вилучення файлів з архівного файлу.

Спочатку давайте подивимося, як створити заархівований файл. Припустимо, я наразі перебуваю в каталозі під назвою 'батьківський'. Всередині цього батьківського каталогу у мене є ще один каталог, який називається „directory1“, який містить два текстові файли, як показано на знімку екрана нижче.

Команда:

pwd

ls

каталог ls1

ad

```
-sh-4.1$ pwd
/home/cpf657/kaushapx/test/parent
-sh-4.1$ ls
child directory1 directory2
-sh-4.1$ ls directory1
DemoFile.txt LearnLinuxCommands.txt
-sh-4.1$
```

Тепер я буду використовувати команду tar, щоб архівувати всі файли в directory1, використовуючи наведену нижче команду:

Команда: tar -cvf *каталог testarchive.tar1 /*

```
-sh-4.1$ tar -cvf testarchive.tar directory1/
directory1/
directory1/DemoFile.txt
directory1/LearnLinuxCommands.txt
-sh-4.1$ pwd
/home/cpf657/kaushapx/test/parent
-sh-4.1$ ls
child directory1 directory2 testarchive.tar
```

ad

Як ви можете бачити на наведеному вище скріншоті, я використав три **варіанти: -c -v та -f** .

- **-c** - це параметр create, який використовується для створення файлу архіву.
- **-v** це багатослівна опція, яка перелічить файли у вікні терміналу, які були додані до вікна архіву, як ви можете бачити на наведеному вище скріншоті.
- **-f** параметр використовується для надання бажаного імені архівному файлу. Назва файлу має слідувати відразу після параметра -f.

У наведеному вище прикладі файл архіву створюється, але він не стискається.

Існує два варіанти стиснення файлу архіву. Це:

-z : Це опція gzip. Він вказує tar використовувати утиліту gzip для стиснення файлу архіву. Він пропонує гідне стиснення та розумну швидкість. Краще додати суфікс „.gz“ після назви файлу, щоб, коли хтось намагається витягти файли з архіву, він міг легко дізнатися про вид стиснення.

Команда: `tar -cvzf testarchive.tar.gz каталог1 /`

ad

```
-sh-4.1$ tar -cvzf testarchive.tar.gz directory1/
directory1/
directory1/DemoFile.txt
directory1/LearnLinuxCommands.txt
-sh-4.1$ ls
child directory1 directory2 testarchive.tar testarchive.tar.gz
```

-j: Це опція bzip2. Він використовує чудовий алгоритм стиснення для стиснення. Він пропонує краще стиснення та меншу швидкість. Краще додати суфікс ‘.bz2’ після імені файлу, щоб, коли хтось намагається витягти файли з архіву, він міг легко дізнатися про тип стиснення.

Команда: `tar -cvjf testarchive.tar.bz2 каталог1 /`

```
-sh-4.1$ tar -cvjf testarchive.tar.bz2 directory1/
directory1/
directory1/DemoFile.txt
directory1/LearnLinuxCommands.txt
-sh-4.1$ ls -sh
total 40K
0 child  0.0K directory1  0 directory2  16K testarchive.tar  8.0K testarchive.tar.bz2  8.0K testarchive.tar.gz
-sh-4.1$
```

Тепер обговоримо команду для вилучення файлів з архівного файлу. Нам потрібно використовувати **-x** тобто витягнути варіант для цієї мети. Параметри **-v** та **-f** поведуться в цьому випадку так само, як і для архіву.

ad

Команда: `tar -xvf testarchive.tar`

```
-sh-4.1$ ls
child directory1 directory2 testarchive.tar testarchive.tar.bz2 testarchive.tar.gz
-sh-4.1$ tar -xvf testarchive.tar
directory1/
directory1/DemoFile.txt
directory1/LearnLinuxCommands.txt
```

Для вилучення файлів з архіву .tar.gz вам потрібно використовувати опцію **-z** та для вилучення файлів з .tar.bz2, вам потрібно скористатися опцією **-j**.

Команди:

```
tar -xvzf testarchive.tar.gz
tar -xvjf testarchive.tar.bz2
```

```
-sh-4.1$ tar -xvzf testarchive.tar.gz
directory1/
directory1/DemoFile.txt
directory1/LearnLinuxCommands.txt
-sh-4.1$ tar -xvjf testarchive.tar.bz2
directory1/
directory1/DemoFile.txt
directory1/LearnLinuxCommands.txt
-sh-4.1$
```

Ads by optAd360

(V) tail: Ця команда використовується для введення останніх 10 рядків файлу. За замовчуванням він відображає останні 10 рядків, але якщо ви хочете вказати кількість рядків, тоді ви можете використовувати параметр -n.

Синтаксис є **хвіст** , або **хвіст -n**

Команда: *хвіст DemoFile.txt*

```
-sh-4.1$ tail DemoFile.txt
today is a beautiful day
I am thinking to go out for a vacation
let me try to find out a best place to visit
I am fond of beaches
I really like going to sea shores and watching the waves
And, I also like exploring tea cafes
My favorite past time is to visit a tea cafe
And doing chit chat with my friends
And, ofcourse I like to do shopping
I prefer online shopping for buying makeup-sh-4.1$
```

Ads by optAd360

Команда: *хвіст -n 5 DemoFile.txt*

```
-sh-4.1$ tail -n 5 DemoFile.txt
And, I also like exploring tea cafes
My favorite past time is to visit a tea cafe
And doing chit chat with my friends
And, ofcourse I like to do shopping
I prefer online shopping for buying makeup-sh-4.1$
```

(vi) sudo: sudo означає суперкористувач do. Ця команда використовується для виконання дій, для яких потрібні права адміністратора або суперкористувача, наприклад, зміна пароля для іншого користувача тощо.

(vii) ps: Ця команда включає процеси, які запущені в поточній оболонці.

Команда: *ps*

```
-sh-4.1$ ps
  PID TTY          TIME CMD
  9872 pts/0    00:00:00 sh
 20747 pts/0    00:00:00 ps
-sh-4.1$
```

ad

Якщо ви хочете побачити процеси, що працюють для певного користувача, вам потрібно використовувати **-u** (користувач) в команді.

Команда: *ps -u kausharx*

```
-sh-4.1$ ps -u kaushapx
  PID TTY          TIME CMD
  9864 ?            00:00:00 sshd
  9872 pts/0        00:00:00 sh
 22839 pts/0        00:00:00 ps
-sh-4.1$
```

Щоб відобразити кожен запущений процес, потрібно використовувати -e (кожен) варіант, тобто **ps -e** в команді .

(viii) ping: Це дуже корисна команда для усунення несправностей мережі. Це дозволяє перевірити, чи підключені ви до мережі з іншою мережевою машиною.

Ads by optAd360

Синтаксис цієї команди: **ping**

Вам потрібно натиснути Ctrl + C, щоб зупинити вихід команди ping.

Наприклад, спробуймо перевірити на пінг веб-сайт Google, щоб перевірити, доступний він чи ні.

Команда: *ping www.google.com*

```
-sh-4.1$ ping www.google.com
PING www.google.com (216.58.192.196) 56(84) bytes of data.
64 bytes from ord30s25-in-f196.1e100.net (216.58.192.196): icmp_seq=1 ttl=48 time=17.4 ms
64 bytes from ord30s25-in-f196.1e100.net (216.58.192.196): icmp_seq=2 ttl=48 time=17.3 ms
64 bytes from ord30s25-in-f196.1e100.net (216.58.192.196): icmp_seq=3 ttl=48 time=17.3 ms
64 bytes from ord30s25-in-f196.1e100.net (216.58.192.196): icmp_seq=4 ttl=48 time=17.3 ms
64 bytes from ord30s25-in-f196.1e100.net (216.58.192.196): icmp_seq=5 ttl=48 time=17.3 ms
64 bytes from ord30s25-in-f196.1e100.net (216.58.192.196): icmp_seq=6 ttl=48 time=17.3 ms
64 bytes from ord30s25-in-f196.1e100.net (216.58.192.196): icmp_seq=7 ttl=48 time=17.3 ms
64 bytes from ord30s25-in-f196.1e100.net (216.58.192.196): icmp_seq=8 ttl=48 time=17.3 ms
64 bytes from ord30s25-in-f196.1e100.net (216.58.192.196): icmp_seq=9 ttl=48 time=17.2 ms
64 bytes from ord30s25-in-f196.1e100.net (216.58.192.196): icmp_seq=10 ttl=48 time=17.3 ms
64 bytes from ord30s25-in-f196.1e100.net (216.58.192.196): icmp_seq=11 ttl=48 time=17.3 ms
64 bytes from ord30s25-in-f196.1e100.net (216.58.192.196): icmp_seq=12 ttl=48 time=17.3 ms
64 bytes from ord30s25-in-f196.1e100.net (216.58.192.196): icmp_seq=13 ttl=48 time=17.3 ms
^C
--- www.google.com ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12526ms
rtt min/avg/max/mdev = 17.292/17.338/17.430/0.167 ms
-sh-4.1$
```

Ads by optAd360

Ви можете бачити у наведеному вище прикладі, що IP-адреса Google 216.58.192.196 відповіла на запит ping і почала надсилати пакети по 64 байти.

Порядкові номери через протокол ICMP (seq = 1, seq = 2 тощо) повідомляють вам, що відсутні пропущені відповіді або втрата пакетів. **ttl** настав час жити за пакет, і це в основному для того, щоб уникнути будь-яких проблем із затопленням мережі.

Час - це тривалість, необхідна для надходження запиту з вашого комп'ютера до google, а потім назад до вашого комп'ютера. Чим він коротший, тим швидший. Врешті-решт, виділена частина - це підсумок, який показує статистику пінгу, яка може використовуватися для моніторингу будь-яких проблем із підключенням до мережі.

Якщо ви хочете пінгувати лише певну кількість разів, ви можете використовувати опцію -c (count). І, якщо ви хочете почути пінг, ви можете використовувати -a (звуковий) параметр. **Наприклад,** ping -ac 6 www.google.com

Ads by optAd360

(ix) passwd: Ця команда дозволяє змінити пароль. Якщо ви змінюєте це для себе, вам просто потрібно ввести команду passwd поодиночі. Але, якщо ви змінюєте його на когось іншого, вам потрібно використовувати цю команду разом із командою sudo.

Для зміни власного пароля: **passwd**

Для зміни іншого пароля користувача: **sudo passwd**

Команда: `sudo passwd mary`

```
~/work$ sudo passwd mary
[sudo] password for dave:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
~/work$
```

Ads by **optAd360**

(x) mv: Це дуже важлива команда. `mv` означає переїзд. Ця команда дозволяє переміщувати файли та папки з одного каталогу в інший. Ви також можете перейменувати файли за допомогою цієї команди.

Для переміщення файлу в нове місце у вашій файловій системі,

Синтаксис: `mv [ім'я файлу] [каталог призначення]`

Команда: `mv DemoFile.txt /home/cpf657/kaushapx/test/parent/directory2`

```
-sh-4.1$ pwd
/home/cpf657/kaushapx/test/parent/directory1
-sh-4.1$ ls
DemoFile.txt  LearnLinuxCommands.txt
-sh-4.1$ mv DemoFile.txt /home/cpf657/kaushapx/test/parent/directory2
-sh-4.1$ ls /home/cpf657/kaushapx/test/parent/directory2
DemoFile.txt
-sh-4.1$
```

ad

Якщо ви просто хочете перейменувати файл, ви можете використовувати наступний синтаксис.

Синтаксис: `mv [ім'я файлу] [ім'я_нового_файла]`

Команда: `mv DemoFile.txt Hello.txt`

```
-sh-4.1$ pwd
/home/cpf657/kaushapx/test/parent/directory2
-sh-4.1$ ls
DemoFile.txt
-sh-4.1$ mv DemoFile.txt Hello.txt
-sh-4.1$ ls
Hello.txt
-sh-4.1$
```

Ads by **optAd360**

Ви також можете здійснити переміщення та перейменування файлів за один раз, як це видно з наведеного нижче прикладу:

Команда: `mv Hello.txt /home/cpf657/kaushapx/test/parent/directory1/DemoFile.txt`

```
-sh-4.1$ pwd
/home/cpf657/kaushapx/test/parent/directory2
-sh-4.1$ ls
Hello.txt
-sh-4.1$ mv Hello.txt /home/cpf657/kaushapx/test/parent/directory1/DemoFile.txt
-sh-4.1$ ls /home/cpf657/kaushapx/test/parent/directory1/
DemoFile.txt  LearnLinuxCommands.txt
-sh-4.1$
```

(xi) mkdir: `mkdir` розшифровується як *make directory*. Ця команда використовується для створення нових каталогів у файловій системі.

Синтаксис: `mkdir` [параметри...] [каталоги...]

ad

Якщо ви хочете створити новий каталог всередині поточного каталогу, ви можете вказати лише ім'я каталогу в команді. В іншому випадку вам потрібно вказати шлях до нового каталогу.

```
-sh-4.1$ pwd
/home/cpf657/kaushapx/test/parent/directory2
-sh-4.1$ mkdir newTestDir
-sh-4.1$ ls
newTestDir
-sh-4.1$ mkdir /home/cpf657/kaushapx/test/parent/directory1/newTest2Dir
-sh-4.1$ ls /home/cpf657/kaushapx/test/parent/directory1/
DemoFile.txt LearnLinuxCommands.txt newTest2Dir
-sh-4.1$
```

Деякі параметри, які можна використовувати з командою `mkdir`, такі:

- **–Версія** : Відображається інформація про версію та ліцензію.

Команда: `mkdir --version`

```
-sh-4.1$ mkdir --version
mkdir (GNU coreutils) 8.4
Copyright (C) 2010 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by David MacKenzie.
-sh-4.1$
```

Ads by optAd360

- **–Допомога**: Тут відображається вся інформація щодо команди `mkdir`.

Команда: `mkdir --help`

```
-sh-4.1$ mkdir --help
Usage: mkdir [OPTION]... DIRECTORY...
Create the DIRECTORY(ies), if they do not already exist.

Mandatory arguments to long options are mandatory for short options too.
-m, --mode=MODE      set file mode (as in chmod), not a=rwx - umask
-p, --parents         no error if existing, make parent directories as needed
-v, --verbose         print a message for each created directory
-Z, --context=CTX     set the SELinux security context of each created
                      directory to CTX
When COREUTILS_CHILD_DEFAULT_ACLS environment variable is set, -p/--parents
option respects default umask and ACLs, as it does in Red Hat Enterprise Linux 7 by default
--help              display this help and exit
--version            output version information and exit

Report mkdir bugs to bug-coreutils@gnu.org
GNU coreutils home page: <http://www.gnu.org/software/coreutils/>
General help using GNU software: <http://www.gnu.org/gethelp/>
For complete documentation, run: info coreutils 'mkdir invocation'
-sh-4.1$
```

- **-v**: Це багатослівний варіант. Він відобразить повідомлення для створеного каталогу.

Команда: `mkdir -v test3Dir`

ad

```
-sh-4.1$ mkdir -v test3Dir
mkdir: created directory `test3Dir'
-sh-4.1$
```

- **-p**: Цей параметр використовується для створення батьківських каталогів у тому випадку, коли він не існує для новоствореного каталогу.

Команда: `mkdir -vp parent2 / child2`

```
-sh-4.1$ mkdir -vp parent2/child2
mkdir: created directory `parent2'
mkdir: created directory `parent2/child2'
-sh-4.1$
```

- **-m:** Цей параметр використовується для встановлення дозволів для новоствореного каталогу.
Синтаксис: `mkdir -m a = rwx` [каталоги]
Припустимо, ви хочете лише надати дозвіл на читання каталогу, тоді ви можете використовувати таку команду.

Ads by optAd360

Команда: `mkdir -vm a = r readOnlyDir`

```
-sh-4.1$ pwd
/home/cpf657/kaushapx/test/parent/directory1
-sh-4.1$ mkdir -vm a=r readOnlyDir
mkdir: created directory `readOnlyDir'
-sh-4.1$ ls -l
total 16
-rw-r--r-- 1 kaushapx acsgrp 416 Apr  4 09:08 DemoFile.txt
-rw-r--r-- 1 kaushapx acsgrp  50 Apr  4 05:32 LearnLinuxCommands.txt
drwxr-xr-x 2 kaushapx acsgrp  80 Apr  5 09:57 newTest2Dir
drwxr-xr-x 3 kaushapx acsgrp  80 Apr  5 10:36 parent1
drwxr-xr-x 3 kaushapx acsgrp  80 Apr  5 10:37 parent2
dr--r--r-- 2 kaushapx acsgrp  80 Apr  5 10:41 readOnlyDir
drwxr-xr-x 2 kaushapx acsgrp  80 Apr  5 10:33 test3Dir
```

На наведеному вище скріншоті ви можете побачити, що `readOnlyDir` - це каталог із дозволами лише на читання (`dr - r - r-`)

(xii) **псевдонім:** Ця команда використовується для введення вашого імені команді або комбінації команд.

Ads by optAd360

Синтаксис: псевдонім `[-p]` [ім'я [= значення]]...

Наприклад, якщо ви хочете перейменувати команду `mv` для переміщення, ви можете зробити це наступним чином.

Команда: псевдонім `mv = переміщення`

```
-sh-4.1$ alias mv=move
```

(xiii) **кішка:** кішка означає сполучення. Ця команда зчитує дані з файлу та відображає їх вміст як вивід у вікні терміналу.

Давайте побачимо кілька варіантів цієї команди:

- **ім'я файлу `cat`:** Ця команда покаже вміст даного файлу як вихід у вікні терміналу.

Команда: `cat LearnLinuxCommands.txt`

ad

```
-sh-4.1$ cat LearnLinuxCommands.txt
hello , this is a file for learning linux commands-sh-4.1$
-sh-4.1$
```

- **файл `cat1` файл2:** Ця команда об'єднає вміст двох файлів і відобразить у вихідних даних.

Команда: `cat LearnLinuxCommands.txt DemoFile.txt`

```
-sh-4.1$ cat LearnLinuxCommands.txt DemoFile.txt
hello , this is a file for learning linux commandsWelcome , hope you are doing good
today is a beautiful day
I am thinking to go out for a vacation
let me try to find out a best place to visit
I am fond of beaches
I really like goingto sea shores and watching the waves
And, I also like exploring tea cafes
My favorite past time is to visit a tea cafe
And doing chit chat with my friends
And, ofcourse I like to do shopping
I prefer online shopping for buying makeup-sh-4.1$
```

- **cat -n ім'я файлу:** Відображення вмісту файлу разом з номерами його рядків.

Команда: `cat -n Demofile.txt`

ad

```
-sh-4.1$ cat -n DemoFile.txt
 1 Welcome , hope you are doing good
 2 today is a beautiful day
 3 I am thinking to go out for a vacation
 4 let me try to find out a best place to visit
 5 I am fond of beaches
 6 I really like goingto sea shores and watching the waves
 7 And, I also like exploring tea cafes
 8 My favorite past time is to visit a tea cafe
 9 And doing chit chat with my friends
10 And, ofcourse I like to do shopping
11 I prefer online shopping for buying makeup-sh-4.1$
```

- **cat> newFileName:** Ця команда використовується для створення нового файлу.
- **cat [ім'я_файла_джерела]> [ім'я_файлу_назначення]:** Ця команда використовується для копіювання вмісту одного файлу в інший. Зверніть увагу, що він замінить вміст цільового файлу вмістом вихідного файлу.
- **кішка [файл1] >> [файл2]:** Ця команда додасть вміст file1 до кінця вмісту file2.
- **tac ім'я файлу:** Ця команда покаже вміст файлу в зворотному порядку.
- **cat -E ім'я файлу:** Ця команда використовується для виділення кінця рядка.

Ads by optAd360

(xiv) chmod: Ця команда використовується для встановлення дозволів для файлу або каталогу. Якщо ви виконаєте `ls -l` для будь-якого файлу або каталогу, на виході ви побачите початкові символи, такі як `-rwxrwxrwx`. Перший символ '-' тут говорить, що це файл, якщо він відображає 'd', то це означає, що це каталог.

Наступні три символи представляють дозволи для користувача, наступні представляють дозволи групи, а останні три представляють дозволи для інших. `r` означає читати, `w` означає писати, а `x` означає виконувати. Якщо замість `gwx` ви деє побачите `-`, це означає, що дозвіл на файл не надано.

Щоб встановити дозвіл, потрібно скористатися командою `chmod`.

Всього існує вісім цифр, які представляють різні рівні дозволу.

0: Немає дозволу

1: виконати

два: писати

3: писати та виконувати

4: читати

5: читати та виконувати

6: читати і писати

7: читати, писати та виконувати

Ads by optAd360

Припустимо, я хочу використовувати дозвіл на читання, запис і виконання для себе (7), дозвіл лише на читання для групи (4) та дозвіл на читання та виконання для інших (5), я буду використовувати 745.

Команда:

ls -l DemoFile.txt

chmod -R 745 DemoFile.txt

ls -l DemoFile.txt

```
-sh-4.1$ ls -l DemoFile.txt
-rw-r--r-- 1 kausharx acsgrp 50 Apr  5 13:17 DemoFile.txt
-sh-4.1$ chmod -R 745 DemoFile.txt
-sh-4.1$ ls -l DemoFile.txt
-rwxr--r-x 1 kausharx acsgrp 50 Apr  5 13:17 DemoFile.txt
-sh-4.1$
```

(xv) **чоун:** chown означає зміну власника. За допомогою цієї команди ви можете змінити власника та власника групи для файлу.

Ads by optAd360

Синтаксис: chown [ОПЦІЯ]... [ВЛАСНИК] [: [ГРУПА]] ФАЙЛ...

Якщо ви хочете змінити власника файлу, ви можете використовувати цю команду як:

chown [ім'я_власника] [ім'я_файла]

Якщо ви хочете змінити власника групи, вам потрібно буде використовувати цю команду з командою sudo як:

sudo chown [ім'я_власника]: [ім'я_групи] [ім'я файлу]

(xvi) **df:** Ця команда показує розмір, використовуваний простір, доступний простір, відсоток використання та змонтована на деталях для файлових систем вашого комп'ютера.

-h та -x є зручними для читання та виключають параметри відповідно. -h показує розмір у зручному для читання форматі (Гб та Мб). Параметр -x використовується для виключення будь-якої файлової системи, яка вас не цікавить.

ad

У наведеному нижче прикладі я виключив сквош-файли.

Команда: df -h -x кабачки

```
-sh-4.1$ df -h -x squashfs
Filesystem          Size  Used Avail Use% Mounted on
/dev/mapper/localvg-root
                    20G   9.9G   8.8G   53% /
tmpfs                32G    41M   32G    1% /dev/shm
/dev/mapper/mpathapl 976M  190M  736M   21% /boot
/dev/mapper/localvg-tmp
                    4.0G   772M   3.0G   21% /tmp
/dev/mapper/rocketvg-rocket
                    9.9G   236M   9.7G    3% /opt/inform
/dev/mapper/localvg-transformationlocal
                    493G    71M  468G    1% /transformation-local
```

(xvii) **різниця:** diff означає різницю. Ця команда використовується для порівняння двох текстових файлів та показу відмінностей між ними.

Ads by optAd360

Синтаксис: diff [параметр] ім'я файлу1 ім'я файлу2

Команда: `diff file1 file2`

```
-sh-4.1$ diff file1 file2
3,4c3
< this is difference
< this is new line
---
> this is line3
-sh-4.1$
```

Ви можете використовувати певні параметри за допомогою цієї команди, наприклад **-Y** для пліч-о-пліч порівняння, **-IN** для вказівки ширини рядка, щоб уникнути обгортання, **-Придушення-загальні рядки** для придушення загальних рядків між двома файлами у виведенні різниці.

Команда:

Ads by optAd360

`diff -y -W 70 файл1 файл2`

`diff -y -W 70 файл1 файл2 --придушення-загальні рядки`

```
-sh-4.1$ diff -y -W 70 file1 file2
this is line1                this is line1
this is line2                this is line2
this is difference           |   this is line3
this is new line             <
-sh-4.1$ diff -y -W 70 file1 file2 --suppress-common-lines
this is difference           |   this is line3
this is new line             <
-sh-4.1$
```

(xviii) відлуння: Команда `echo` надрукує текст у вікні терміналу. Ви також можете передавати змінні середовища в команді `echo`, як показано нижче.

Команда:

`ad`

`exo`

Привіт

Світ

`echo`

\$

USER

`echo`

\$

PATH

`echo $ HOME`

```
-sh-4.1$ echo Hello World
Hello World
-sh-4.1$ echo $USER
kaushapx
-sh-4.1$ echo $PATH
/usr/lib64/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/opt/dell/srvadmin/bin
-sh-4.1$ echo $HOME
/home/cpf657/kaushapx
```

(xix) знайти: Ця команда використовується для пошуку файлів і каталогів у файловій системі. Ви можете знайти, вказавши ім'я файлу, ім'я каталогу, дату створення, дату модифікації, власника файлу, дозволи на файл тощо. Ви також можете використовувати символи підстановки у виразі пошуку.

Синтаксис: `знайти [почати пошук з цієї точки] [вираз пошуку] [-опції] [що знайти]`

Ads by optAd360

Команда:

знайти. -ім'я “ файл *”*

```
-sh-4.1$ find . -name "*file*"
./newfile1
./file1
./file2
-sh-4.1$
```

У наведеному вище прикладі ця команда шукала всі файли в поточному каталозі (.), Який містить рядок „файл” (* файл *) в їх назвах файлів (опція -name).

Спробуйте використати `find --help` для вивчення різних опцій, які надає ця команда.

(xx) Вбивство: Це вбудована команда в Linux, яка використовується для завершення процесу вручну. Вам потрібно надати ідентифікатор процесу (PID) команді `kill`, щоб завершити процес.

Синтаксис: вбити PID

(xxi) історія: Ця команда покаже команди, які ви раніше виконували в командному рядку. Щоб переглянути історію всіх команд, вам потрібно просто ввести **історії**.

Наприклад, Якщо я хочу побачити останні 10 виконуваних мною команд, я буду використовувати команду нижче як

історія 10

```
-sh-4.1$ history 10
1033  ls
1034  find . -name *file*
1035  /home/cpf657/kaushapx/test/parent/directory1
1036  find ./ -type f -name "*.txt"
1037  find . -type f -name "*.txt"
1038  find . -name "*file*"
1039  pwd
1040  ls
1041  history -5
1042  history 10
-sh-4.1$
```

ad

У крайньому лівому куті ви можете побачити такі цифри, як 1033, 1034, .. Це номери подій. Це може бути різним для різних комп'ютерів. Ці номери подій можна використовувати для повторення певної команди.

Синтаксис: ! Номер події .

Наприклад,

! 1033

```
-sh-4.1$ !1033
ls
DemoFile.txt  file1  file2  LearnLinuxCommands.txt  newfile1
-sh-4.1$
```

Якщо ви хочете повторити останню виконану команду, просто введіть **!!**

ad

(xxii) голова: Команда `head` показує 10 верхніх рядків файлу. Щоб переглянути конкретну кількість верхніх рядків, використовуйте параметр `-n`.

Команда: `head -n 3 файл2`

```
-sh-4.1$ head -n 3 file2
this is line1
this is line2
this is line3
-sh-4.1$
```

(xxiii) gzip: Ця команда використовується для стиснення файлу.

Синтаксис: `gzip [-options] ім'я файлу`

Наприклад,

ad

gzip -v file1

```
-sh-4.1$ gzip -v file1
file1:  53.1% -- replaced with file1.gz
-sh-4.1$
```

За допомогою `gzip -help` досліджуйте різні варіанти команди `gzip`.

(xxiv) Групи: Ця команда відображає імена груп, учасником яких є користувач.

Синтаксис: групи користувачів

Команда: *групи kaushapx*

```
-sh-4.1$ groups kaushapx
kaushapx : acsgrp elsagrп xсegrп f3717
-sh-4.1$
```

Ads by **optAd360**

(xxv) безкоштовно: Ця команда розповідає нам про використання оперативної та обмінної пам'яті (використаної, вільної, спільної, буфери, кешована) у вашому комп'ютері.

Команда: *безкоштовно -ч*

```
-sh-4.1$ free -h
              total        used        free      shared    buffers     cached
Mem:           62G         13G         48G          41M        516M         10G
-/+ buffers/cache:      3.0G         59G
Swap:          63G           0B         63G
-sh-4.1$
```

(xxvi) менше: Це дуже корисна команда, яку ви можете використовувати для перегляду вмісту файлу в самому терміналі Linux, не відкриваючи файл у редакторі.

Синтаксис: менше ім'я_файлу

Вам потрібно натиснути: `q`, щоб вийти з виводу команди `менше`.

`ad`

Висновок

У цьому підручнику ми детально висвітлили багато корисних та важливих команд Linux разом із прикладами. Сподіваюся, це допоможе вам отримати знання та дослідити різні команди на терміналі Linux відповідно до ваших вимог. Щоб дослідити будь-яку команду, ви завжди можете це зробити **command_name -help**. Це покаже вам подробиці використання цієї конкретної команди в Linux.

Щасливого навчання !!

Рекомендована література

- [Команди Unix: основні та вдосконалені команди Unix з прикладами](#)
- [Unix проти Linux: яка різниця між UNIX та Linux](#)
- [Аргументи командного рядка в сценарії оболонки Unix з прикладом](#)
- [Підручник з труб Unix: Труби в програмуванні Unix](#)
- [Аргументи командного рядка на C ++](#)
- [Команди управління процесами Unix, такі як Ps та Top \(Частина C\)](#)
- [Розширені команди Git та підручник з інтеграції GitHub](#)
- [Команди програм Unix Utilities: Which, Man, Find Su, Sudo \(Частина D\)](#)

MySQL ALTER TABLE - Як додати стовпець до таблиці в MySQL

MySQL Alter Table How Add Column Table Mysql

Дізнайтеся про команду MySQL ALTER Table, щоб додати / скинути стовпець, індекс, обмеження, змінити назву таблиці тощо на прикладах:

MySQL ALTER Команда використовується для модифікації існуючої таблиці шляхом додавання нового стовпця або видалення існуючого стовпця або зміни типу даних стовпця.

Простіше кажучи, команда **ALTER** використовується для модифікації структури існуючої таблиці шляхом додавання / видалення / оновлення стовпців, перейменування таблиць тощо.

Практично неможливо, щоб усі вимоги до даної схеми були доступні заздалегідь, отже, обов'язково повинен бути спосіб оновлення структури таблиці за необхідності.



Що ви дізнаєтесь:

- MySQL ALTER TABLE
- Різні способи використання команди таблиці ALTER
- ДОДАТИ СТОЛЬБУ
- КРАПКА КРАПКИ
- Додати / скинути обмеження
- Індекси додавання / скидання
- Змінити значення стовпця за замовчуванням
- Оновити назву таблиці
- Часті запитання
- Висновок
- Рекомендована література

MySQL ALTER TABLE

СИНТАКСИС:

ALTER TABLE table_name [alter_option1, alter_option2...]

Тут table_name - це назва таблиці, над якою ми хочемо виконати дію зміни. Разом із значком можна вказати один або кілька варіантів alter_options **ЗМІНИТЬ СТОЛ** команди.

Давайте розглянемо різні варіанти використання alter_options, надані MySQL.

Зразок даних та налаштування таблиці

З метою розуміння різних звичаїв **ВІК** команда, ми будемо використовувати 2 зразкові таблиці - Співробітник та Відділ, що мають структуру, як зазначено нижче.

Ми також додаємо кілька фіктивних даних до обох таблиць.

Ми не додали жодних обмежень FOREIGN KEY серед цих таблиць. Ми навчимося додавати їх за допомогою команди ALTER у прикладах, показаних у розділах нижче, під час обговорення.

```
CREATE TABLE IF NOT EXISTS Employee_Department ( id INT, name VARCHAR(100),
PRIMARY KEY (id) ); CREATE TABLE IF NOT EXISTS Employee ( name VARCHAR(100), id
INT, address VARCHAR(100), department_id INT, PRIMARY KEY (id) ); INSERT INTO
`employee_department`(`name`,`id`)
VALUES
('Information
Technology',1),('HR',2),('Finance',3),('Accounting',4),('Housekeeping',5
),('Security',6),('Support',7),('Contract Staff',8),('Sales',9),('Management',10); INSERT INTO
`employee` (`id`,`name`,`address`) VALUES (1,'Alyssa Villarreal','Ap #558-3012 Nulla
Street'),(2,'Vanna Parks','103-986 Cursus Rd.),(3,'Quyn Byers','770 Nulla Avenue'),(4,'Kristen
Hall','Ap #622-9967 Nullam St.),(5,'Chelsea Stone','6017 Elementum St.),(6,'Sacha Sweeney','239-
7402 Etiam St.),(7,'Sophia Charles','799 -745 Tellus. Rd.),(8,'Jennifer Joyner','5803 Ligula.
Avenue'),(9,'Mercedes Head','Ap #375-7999 Facilisis Rd.),(10,'Reagan Holland','P.O. Box 625, 5744
Proin Avenue');
```

Різні способи використання команди таблиці ALTER

MySQL ALTER Команда може використовуватися для кількох речей, таких як додавання / скидання стовпця, додавання / скидання індексу або обмеження та оновлення самої таблиці. Давайте розглянемо різні сценарії за допомогою прикладів.

ДОДАТИ СТОЛЬБУ

З наведеними вище зразками таблиць, давайте додамо новий стовпець з іменем 'department_id' типу INT у таблицю Employee.

```
ALTER TABLE employee ADD COLUMN department_id INT
```

КРАПКА КРАПКИ

Подібно до додавання нового стовпця, **MySQL ALTER** команда може бути використана для видалення існуючого стовпця з таблиці.

Спробуйте видалити доданий вище стовпець department_id із таблиці Employee.

```
ALTER TABLE employee DROP COLUMN department_id
```

Зверніть увагу, що, скидаючи Стовпець, вам просто потрібно вказати назву стовпця, а не його тип даних.

Щоб перевірити, чи вдалася команда ALTER, можна запустити команду SHOW COLUMNS. SHOW COLUMNS FROM employee;

Вихід:

Польовий	Тип	Нуль	Ключ	За замовчуванням	Додатково
ім'я	варчар (100)	ТАК		НУЛЬ	
ідентифікатор	інт	ТАК		НУЛЬ	
адресу	варчар (100)	ТАК		НУЛЬ	

Зараз ви можете бачити, що в таблиці „Співробітник” немає стовпця_відділу.

Додати / скинути обмеження

Давайте тепер подивимось, як за допомогою ALTER можна додати обмеження.

Команда ALTER може використовуватися для додавання або видалення обмежень для існуючої таблиці. Давайте подивимось приклад обмеження ІНОЗЕМНИХ КЛЮЧІВ, яке потрібно додати між 2 таблицями.

Ми будемо використовувати команду ALTER для додавання ІНОЗЕМНОГО КЛЮЧОВОГО ОБМЕЖЕННЯ в таблиці Employee для таблиці worker_department.

Додайте новий стовпець з іменем department_id у таблиці Employee.

```
ALTER TABLE employee ADD COLUMN department_id INT
```

Тепер додайте обмеження FOREIGN KEY.

```
ALTER TABLE employee ADD FOREIGN KEY (department_id) REFERENCES employee_department(id)
```

Щоб перевірити це обмеження, ми можемо використовувати команду SHOW, щоб побачити визначення таблиці.

```
SHOW CREATE TABLE employee;
```

Таблиця	Створити таблицю
працівник	СТВОРИТИ ТАБЛИЦЮ 'працівник' ('name' varchar (100) ЗАВДАННЯ НУЛЬ, 'id' int ЗА замовчуванням NULL, 'address' varchar (100) ЗАВДАННЯ НУЛЬ, 'department_id' int DEFAULT NULL, КЛЮЧ 'Ідентифікатор_відділу' ('Ідентифікатор_відділу'), ОБМЕЖЕННЯ 'worker_ibfk_1' ІНОЗЕМНИЙ КЛЮЧ (' department_id') СПИСОК ЛІТЕРАТУРИ 'worker_department' (' id')) ДВИГУН = ШАРСЕТИ ЗА ЗАМОВЛЕННЯМ InnoDB = utf8mb4 COLLATE = utf8mb4_0900_ai_ci

Ви виявите, що в стовпці department_id створено обмеження FOREIGN KEY.

Давайте тепер подивимося, як ми можемо використовувати команду ALTER, щоб видалити обмеження FOREIGN KEY, яке ми щойно додали.

```
ALTER TABLE employee DROP CONSTRAINT employee_ibfk_1
```

Зверніть увагу, що вам потрібно буде використовувати ім'я обмеження, яке було створено.

Отже, у наведеному вище виводі команди SHOW ви можете побачити назву обмеження, яке MySQL створило для обмеження FOREIGN KEY: **worker_ibfk_1**

Запустіть команду SHOW, щоб побачити, чи обмеження видалено, як показано нижче:

```
SHOW CREATE TABLE employee;
```

Таблиця	Створити таблицю
працівник	СТВОРИТИ ТАБЛИЦЮ 'працівник' ('name' varchar (100) ЗАВДАННЯ НУЛЬ, 'id' int ЗА замовчуванням NULL, 'address' varchar (100) ЗАВДАННЯ НУЛЬ, 'department_id' int DEFAULT NULL, КЛЮЧ 'Ідентифікатор_відділу' ('Ідентифікатор_відділу')) ДВИГУН = ШАРСЕТИ ЗА ЗАМОВЛЕННЯМ InnoDB = utf8mb4 COLLATE = utf8mb4_0900_ai_ci

Індекси додавання / скидання

MySQL ALTER Команда table також дозволяє створювати або скидати INDEXES до таблиці.

За допомогою команди ALTER TABLE можна додати чотири різні типи індексів.

1) Додайте ПЕРВИННИЙ КЛЮЧ: Ця команда додає індекс PRIMARY KEY до заданого стовпця (або стовпців)

У наведеному нижче прикладі використовуйте таблицю Employee та додайте індекс PRIMARY KEY до стовпця «Id».

```
ALTER TABLE employee ADD PRIMARY KEY (id);
```

2) **Додати УНІКАЛЬНЕ:** Додає або створює унікальний індекс щодо заданих стовпців. Припустимо, гіпотетично ми просто хочемо унікальні імена в таблиці. Додайте УНІКАЛЬНИЙ індекс у стовпець „ім'я” таблиці Employee, як показано нижче.

ALTER TABLE employee ADD UNIQUE (name);

3) **Додати ІНДЕКС:** Додайте звичайний індекс до будь-якого стовпця. Додайте звичайний індекс у поле 'address' у таблиці Employee.

ALTER TABLE employee ADD INDEX (name);

4) **Додайте індекс FULLTEXT:** Це спеціальний тип покажчика, який використовується для пошуку тексту.

Ми додамо індекс FULLTEXT до значень стовпця 'name' у таблиці Employee.

ALTER TABLE employee ADD FULLTEXT(address);

Для всіх вищезазначених 4 прикладів ми можемо запустити **ПОКАЖІТЬ СТВОРИТИ ТАБЛИЦЮ** команда, щоб отримати детальну інформацію про створені індекси.

SHOW CREATE TABLE employee;

Таблиця	Створити таблицю
працівник	СТВОРИТИ ТАБЛИЦЮ 'працівник' ('name' varchar (100) ЗАВДАННЯ НУЛЬ, 'id' int NOT NULL, 'address' varchar (100) ЗАВДАННЯ НУЛЬ, 'department_id' int DEFAULT NULL, ПОЧАТКОВИЙ КЛЮЧ ('id'), УНІКАЛЬНИЙ КЛЮЧ 'name' ('назва'), КЛЮЧ 'name_2' ('ім'я'), ФУЛЬТЕКСТОВИЙ КЛЮЧ 'адреса' ('адреса')) ДВИГУН = ШАРСЕТІ ЗА ЗАМОВЛЕННЯМ InnoDB = utf8mb4 COLLATE = utf8mb4_0900_ai_ci

У наведеному вище результаті ви помітите, що всі індекси були створені відповідно до значення в стовпці create_table.

Подібно до створення індексів, команду ALTER також можна використовувати для видалення існуючих індексів із таблиці.

Приклад:

ALTER TABLE employee DROP INDEX address;

Наведена команда знизить повний індекс, який ми додали для поля адреси.

Зверніть увагу на назву 'Адреса'. Це має відповідати імені індексу у вихідному файлі **ПОКАЖІТЬ СТВОРИТИ ТАБЛИЦЮ** команди.

Змінити значення стовпця за замовчуванням

ЗМІНИТЬ СТОЛ Команда також може бути використана для встановлення / зміни значення DEFAULT для існуючого стовпця.

Давайте розглянемо приклад, коли ми змінимо значення за замовчуванням стовпця department_id на «1». При цьому будь-які нові вставки до цієї таблиці змінювали б / встановлювали значення для стовпця department_id на 1

ALTER TABLE employee ALTER department_id SET DEFAULT 1

Додайте новий запис до таблиці Employee:

INSERT INTO 'employee' ('id', 'name', 'address') VALUES (11, 'Saket Sharma', 'NB 23 Domlur')

А тепер давайте запитамо у department_id цей запис і подивимось, яке значення встановлюється.

SELECT department_id FROM employee where id=11 //Output department_id 1

Оновити назву таблиці

MySQL ALTER Команда table може також використовуватися для оновлення імені існуючої таблиці. Іноді потрібно оновити структуру схеми, що може включати зміну / оновлення імен таблиць.

Приклад - Ми перейменуємо таблицю `worker` в `Employee\_details`.

```
ALTER TABLE table_name RENAME TO new_table_name;
```

Часті запитання

Q # 1) Як змінити структуру таблиці в MySQL?

Відповідь: MySQL надає команду ALTER для зміни структури існуючої таблиці в MySQL. Ви можете виконувати різні операції за допомогою команди ALTER, наприклад,

- Додавання / скидання стовпця
- Додавання / видалення індексу або обмеження
- Перейменування таблиці

Q # 2) Чи блокує команда ALTER таблицю?

Відповідь: Це може залежати від версій - у старих версіях MySQL блокування таблиці під час виконання оператора ALTER. Загалом, блокування відбуватиметься, тобто будь-яке читання та запис під час ALTER.

Наприклад - Додавання нового стовпця до існуючої таблиці торкнеться всіх рядків таблиці, до яких додається стовець, і коли таблиця має кількість записів у мільйонах, на виконання операції ALTER може знадобитися деякий час. У цьому випадку блокування під час читання / запису буде дорогим, оскільки база даних буде недоступною та недоступною.

Q # 3) Чи можна додати більше одного стовпця в таблицю ALTER?

Відповідь: Так, команду ALTER можна використовувати для додавання декількох стовпців в межах одного оператора.

Давайте розглянемо приклад, коли у нас є таблиця Employee із стовпцями id, ім'ям та адресою, і припустимо, ми хочемо додати ще 2 стовпці з іменами - найвище_визначення (тип varchar) та номер телефону (тип varchar)

```
ALTER TABLE employee ADD highest_education VARCHAR(50), ADD phone_number VARCHAR(20);
```

Давайте подивимося на інформацію в таблиці

```
SHOW COLUMNS FROM employee;
```

Вихід:

Польовий	Тип	Нуль	Ключ	За замовчуванням	Додатково
ім'я	варчар (100)	ТАК	ОБ'ЄДНАНІ	НУЛЬ	
ідентифікатор	інт	НЕ	PRI	НУЛЬ	
адресу	варчар (100)	ТАК		НУЛЬ	
department_id	інт	ТАК		один	
найвища_освіта	варчар (50)	ТАК		НУЛЬ	
телефонний номер	варчар (20)	ТАК		НУЛЬ	

У вихідних даних ви бачите стовпці, додано найвище_освіта та номер телефону.

Висновок

У цьому підручнику ми дізналися про різні способи використання **MySQL ALTER** Команда таблиці. ALTER є важливою командою, оскільки вона може бути використана для різних цілей для ALTER структури існуючої таблиці в MySQL.

Він може використовуватися для таких речей, як додавання / видалення стовпця, додавання / скидання індексів або обмежень, а також може використовуватися навіть для перейменування існуючої таблиці.

Кожен, хто вивчає MySQL, повинен глибоко розуміти команду таблиці ALTER, оскільки вона в основному використовується на етапах обслуговування баз даних, де потрібно змінити існуючу схему таблиці.

Рекомендована література

- [MySQL Створення таблиці Підручник з прикладами](#)
- [MySQL Insert Into Table - Вставте синтаксис заяви та приклади](#)
- [Типи даних MySQL | Що таке різні типи даних у MySQL](#)
- [Основи обмеження ІНОЗЕМНИХ КЛЮЧІВ MySQL на прикладах](#)
- [Підручник з MySQL JOIN: Внутрішній, Зовнішній, Хрестовий, Лівий, Правий та Я](#)
- [MySQL Create Database - Як створити базу даних в MySQL](#)
- [40 найкращих запитань та відповідей на інтерв'ю MySQL \(2021 запитання\)](#)
- [Хеш-таблиця в C ++: Програми для впровадження хеш-таблиць і хеш-карт](#)

Як читати або записувати дані з аркуша Excel у веб-драйвері Selenium

How Read Write Data From Excel Sheet Selenium Web Driver

У цьому посібнику ви навчитеся читати та записувати дані з файлу Excel у Selenium WebDriver за допомогою API FILLO та SQL.

Читання або запис даних є однією з найбільш часто використовуваних операцій, або отримання значень із таблиць бази даних, або отримання значень з аркуша Excel та використання їх для виконання аналітики.

У цій статті ми говоримо про отримання значень із файлу Excel за допомогою операторів SQL та API FILLO.



Що ви дізнаєтесь:

- [Огляд SON API](#)

- Дозволена операція з Fillo
- Код селену
- Висновок
- Рекомендована література

Огляд SON API

FILLO - це Java API, який використовується для отримання даних із файлів Excel. За допомогою API FILLO параметризація стає дуже простою, тобто запуск тестових кейсів у Selenium з іншим набором даних.

До цього ми використовували JXL API для параметризації, пізніше POI вийшов на ринок apache, і в обох цих API нам потрібно написати великий код, щоб обходити рядки та стовпці та отримувати значення, що зберігаються в аркуші Excel.

Але тепер, завдяки цьому новому API FILLO, не потрібно турбуватися про розмір рядків та стовпців, про все буде дбати API внутрішньо. Вам просто потрібно викликати класи, визначені в API, і оголосити свої тестові дані у файлі Excel. Зверніться до офіційного сайту для отримання документації та завантаження файлу jar - [SON Бібліотека Java](#)

Якщо ви використовуєте Maven, тоді використовуйте залежність Maven.

Банка FILLO із залежністю Maven

```
<!-- https://mvnrepository.com/artifact/com.codoid.products/fillo -->
<dependency>
  <groupId>com.codoid.products</groupId>
  <artifactId>fillo</artifactId>
  <version>1.18</version>
</dependency>
```

почати

- Це Excel API для мови Java.
- Він підтримує файли .xls та .xlsx.
- Він підтримує запити SELECT, UPDATE та INSERT.
- Використовувати з реченням WHERE та реченням LIKE або без нього.

Дозволена операція з Fillo

Операція SELECT: Оператор SELECT виконує ту саму функцію, що і при отриманні значень з таблиці та відображенні кінцевому користувачеві, так само, як тут оператор SELECT повертає дані з аркуша Excel.

Синтаксис:

SELECT * From Sheet Name

Операція оновлення: Оператор UPDATE змінює існуючі записи в аркуші Excel.

Синтаксис:

UPDATE sheet1 Set Column Name= 'Value'

Операція INSERT: Оператор INSERT вставляє новий запис у аркуш Excel.

Синтаксис:

INSERT INTO Sheet Name (ColumnName1,ColumnName2) VALUES ('Val1','Val2')

Виконайте однакові операції з операторами WHERE та LIKE:

- "ВИБЕРІТЬ * із імені аркуша, де ID = 1 та name = 'Ісус'"
- "ВИБЕРІТЬ * з імені аркуша, де стовпець1 = значення1 і стовпець2 = значення2 і стовпець3 = значення3"
- "ОНОВИТИ Ім'я аркуша Набір Країна = 'Великобританія', де ID = 10 та ім'я = 'Ісус'"
- "ВИБЕРІТЬ * з імені аркуша, де ім'я на зразок, Jes%"

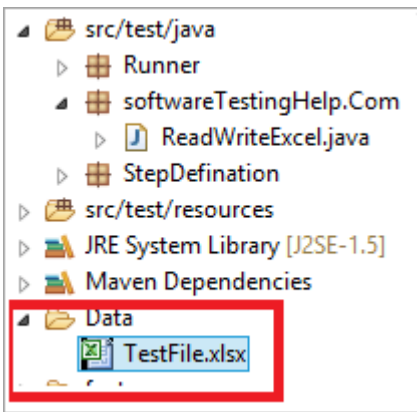
Етапи виконання, яких слід дотримуватися для операції ВИБОР / ВСТАВКА / ОНОВЛЕННЯ:

1) // Створення об'єкта класу Fillo.

Тепер, використовуючи наведені вище кроки, у фреймворці Selenium ми отримаємо значення рядків і стовпців, що зберігаються у файлі Excel, а саме “TestFile.xlsx”, і введемо значення на демонстраційному сайті: <https://wordpress.com/start/about?ref=create-blog-lp>

Архуш Excel із значеннями рядків: 'Нова тестова довідка' та 'Testing_Related_Contents'

Файл Excel, розміщений у папці Project:



Код селену

Повний код селену, який отримує значення з аркуша Excel та входить на тестовому майданчику, поданий нижче:

```
package softwareTestingHelp.Com; import org.openqa.selenium.By; import
org.openqa.selenium.WebDriver; import org.openqa.selenium.chrome.ChromeDriver; import
com.codoid.products.exception.FilloException; import com.codoid.products.fillo.Connection; import
com.codoid.products.fillo.Fillo; import com.codoid.products.fillo.Recordset; public class
ReadWriteExcel { static WebDriver driver; //demo site -https://wordpress.com/start/about?ref=create-
blog-lp //download jar file - https://mvnrepository.com/artifact/com.codoid.products/fillo
publicstaticvoid main(String args[]) throwsInterruptedException, FilloException{ //Calling up the
GoogleChrome driver System.setProperty('webdriver.chrome.driver', 'D:\Srinivas\New
folder\exe\chromedriver.exe'); driver = newChromeDriver(); //Opening the demo site -
wordpress.com driver.get('https://wordpress.com/start/about?ref=create-blog-lp'); //Locating the Test
data excel file String excelPath = '.\Data\TestFile.xlsx'; System.out.println(excelPath); //Create an
Object of Fillo Class Fillo fillo = newFillo(); //Create an Object for Connection class and use
getConnection() //method defined inside Fillo class, to establish connection between excelsheet and
Fillo API's. Connection connection = fillo.getConnection(excelPath); //Select all the values present
in a sheet, which is present inside the excel and store its output in a String variable String
strSelectQuery = 'Select * from Data'; System.out.println(strSelectQuery); //Execute the Select
query and store the result in a Recordset class present in Fillo API. Recordset recordset =null;
recordset = connection.executeQuery(strSelectQuery); //use while loop to iterate through all columns
and rows available in sheet present inside excel file while(recordset.next()){
System.out.println('Column 1 = ' +recordset.getField('SiteTitle')); String siteTitle =
recordset.getField('SiteTitle'); driver.findElement(By.xpath('//input[@name='siteTitle']')).clear();
driver.findElement(By.xpath('//input[@name='siteTitle']')).sendKeys(siteTitle)
;
System.out.println('Column 2 = ' +recordset.getField('SiteTopic')); String siteTopic =
recordset.getField('SiteTopic'); driver.findElement(By.xpath('//input[@name='siteTopic']')).clear();
driver.findElement(By.xpath('//input[@name='siteTopic']')).sendKeys(siteTopic)
;
connection.close(); } //Use update query to update the details in excel file Connection connection1 =
fillo.getConnection(excelPath); System.out.println('Column 1 value before Update clause = '
+recordset.getField('SiteTitle')); String strUpdateQuery = 'Update Data Set SiteTitle =
'SoftwareTestingHelp.com'
'; System.out.println(strUpdateQuery);
connection1.executeUpdate(strUpdateQuery); System.out.println('Column 1 value after Update
clause = ' +recordset.getField('SiteTitle')); //Use Insert query to update the data in excel sheet
Connection connection2 = fillo.getConnection(excelPath); System.out.println('Column 1 and column
2 value before insert clause = ' +recordset.getField('SiteTitle') +recordset.getField('siteTopic')); String
strInsertQuery = 'INSERT INTO Data (SiteTitle,SiteTopic) Values('Bharat','NewDelhi')';
System.out.println(strInsertQuery); connection2.executeUpdate(strInsertQuery);
```

```
System.out.println('Column 1 and column 2 value after insert clause = ' +recordset.getField('SiteTitle')  
+recordset.getField('siteTopic')); } }
```

Знімок запущеного коду

ReadWriteExcel.java

```
1 package softwareTestingHelp.Com;
2
3 import org.openqa.selenium.By;
4
5 import org.openqa.selenium.WebDriver;
6
7 import org.openqa.selenium.chrome.ChromeDriver;
8
9 import com.codoid.products.exception.FilloException;
10
11 import com.codoid.products.fillo.Connection;
12
13 import com.codoid.products.fillo.Fillo;
14
15 import com.codoid.products.fillo.Recordset;
16
17 public class ReadWriteExcel {
18
19     static WebDriver driver;
20
21     //demo site -https://wordpress.com/start/about?ref=create-blog-lp
22     //download jar file - https://mvnrepository.com/artifact/com.codoid.products/fillo
23
24     public static void main(String args[]) throws InterruptedException, FilloException{
25
26         //Calling up the GoogleChrome driver
27         System.setProperty("webdriver.chrome.driver", "D:\\Srinivas\\New folder\\exe\\chromedriver.exe");
28         driver = new ChromeDriver();
29
30         //Opening the demo site - wordpress.com
31         driver.get("https://wordpress.com/start/about?ref=create-blog-lp");
32
33         //Locating the Test data excel file
34         String excelPath = ".\\Data\\TestFile.xlsx";
35         System.out.println(excelPath);
36
37         //Create an Object of Fillo Class
38         Fillo fillo = new Fillo();
39
40         //Create an Object for Connection class and use getConnection()
41         //method defined inside Fillo class, to establish connection between excel sheet and Fillo API's.
42         Connection connection = fillo.getConnection(excelPath);
43
44         //Example of SELECT Statement
45         //Select all the values present in a sheet, which is present inside the excel and store its output in
46         //String variable
47         String strSelectQuery = "Select * from Data";
48         System.out.println(strSelectQuery);
49
50         //Execute the Select query and store the result in a Recordset class present in Fillo API.
51         Recordset recordset = null;
52         recordset = connection.executeQuery(strSelectQuery);
53
54         //use while loop to iterate through all columns and rows available in sheet present inside excel file
55         while(recordset.next()){
56
57             System.out.println("Column 1 = " +recordset.getField("SiteTitle"));
58             String siteTitle = recordset.getField("SiteTitle");
59
60             driver.findElement(By.xpath("//input[@name='siteTitle']")).clear();
61             driver.findElement(By.xpath("//input[@name='siteTitle']")).sendKeys(siteTitle);
62
63             System.out.println("Column 2 = " +recordset.getField("SiteTopic"));
64             String siteTopic = recordset.getField("SiteTopic");
65
66             driver.findElement(By.xpath("//input[@name='siteTopic']")).clear();
67             driver.findElement(By.xpath("//input[@name='siteTopic']")).sendKeys(siteTopic);
68             connection.close();
69
70         }
71
72         //Use update query to update the details in excel file
73         Connection connection1 = fillo.getConnection(excelPath);
74
75         System.out.println("Column 1 value before Update clause = " +recordset.getField("SiteTitle"));
76         String strUpdateQuery = "Update Data Set SiteTitle = 'SoftwareTestingHelp.com' ";
77
78         System.out.println(strUpdateQuery);
```

Вивід коду, який відображає результат оператора SQL:

```
Starting ChromeDriver 2.38.552522 (437e6fbedfa8762dec75e2c5b3ddb84763dc9dcb) on port 33196
Only local connections are allowed.
Jun 12, 2018 11:02:48 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS
.\Data\TestFile.xlsx
Select * from Data
Column 1 = Software
Column 2 = Testing Related Contents
Result of Select statement
Column 1 value before Update clause = Software
Update Data Set SiteTitle = 'SoftwareTestingHelp.com'
1 row(s) affected
Column 1 value after Update clause = Software
Result of Update Statement
Column 1 and column 2 value before insert clause = SoftwareTesting_Related_Contents
INSERT INTO Data (SiteTitle,SiteTopic) Values('Bharat','NewDelhi')
2 columns(s) affected
Result of Insert Statement
Column 1 and column 2 value after insert clause = SoftwareTesting_Related_Contents
```

Файл Excel після виконання операції оновлення та вставки:

A	B
SiteTitle	SiteTopic
SoftwareTestingHelp.com	Testing_Related_Contents
Bharat	NewDelhi

Демонстраційний веб-сайт, на якому ми отримали дані з Excel і вставили на сайт:

WordPress.com Log In

Step 1 of 4

Let's create a site.

Please answer these questions so we can help you make the site you need.

What would you like to name your site?

New Test Help

What will your site be about?

Testing_Related_Contents

What's the primary goal you have for your site?

Висновок

- Fillo - це дуже корисний Java-API для вилучення даних з аркуша Excel, і він підтримує як .xls, так і .xlsx файли Excel.
- Він підтримує оператори SELECT, UPDATE та INSERT.
- Будьте обережні під час редагування файлу Excel. Якщо будь-які рядки або стовпці не потрібні, видаліть їх.
- Якщо ви просто стерте значення з рядків і стовпців, не видаляючи весь рядок, тоді API припустить, що поля мають значення, і спробує отримати значення з рядків і стовпців, а натомість ми отримаємо порожні значення.
- Нарешті, не забудьте розірвати з'єднання, коли закінчите з вибором значень з Excel.

Щасливого читання !!

Рекомендована література

- Вступ до Selenium WebDriver - Підручник з селену №8
- 25 найкращих команд селену WebDriver, які ви повинні знати
- 10 найкращих засобів тестування API у 2021 році (SOAP та REST API)

- Підручник JAVA для початківців: 100+ практичних навчальних посібників Java
- Найпопулярніші 90 запитань та відповідей на інтерв'ю SQL (ОСТАННІ)
- 30+ найкращих підручників із селену: вивчіть селен на реальних прикладах
- Створення Framework селену та доступ до тестових даних з Excel - Підручник селену №21
- Data Driven Framework у Selenium WebDriver з використанням Apache POI

Топ 10 популярних тестових платформ та інструментів PHP

Top 10 Popular Php Testing Frameworks

Огляд тестових платформ та інструментів PHP:

У сучасному світі PHP є однією з найшвидших та найбільш універсально використовуваних мов сценаріїв на стороні сервера.

PHP в основному використовується для великих веб-сайтів, таких як Facebook, WordPress тощо. Він має точну та точну сертифікацію, великі асоціації та підтримувані фреймворки. Мова програмування PHP набагато простіше для розробника створювати веб-сайти порівняно з іншими мовами, такими як RUBY, SCALA, JAVA тощо.

Для надання коду якості необхідно провести тестування. Оскільки PHP використовується для великомасштабних веб-сайтів, для тестування коду PHP на ринку доступні певні інструменти тестування, що, в свою чергу, робить роботу тестера простішою та точнішою.



Автоматизація, яка сьогодні є вимогливою у всьому світі, значно економить людський час та енергію.

Інструменти PHP використовуються для тестування модуля, а також наскрізного тестування за допомогою BDD (поведінка - керована розробка) і TDD (тестова - розробка).

У цій статті ми перерахували 10 найкращих засобів тестування та оптимізації PHP та фреймворків PHP, що полегшує роботу тестувальника та розробника для тестування коду в PHP та повідомлення про помилки.

Перш ніж приступати до списку інструментів, важливо знати різницю між Frameworks та Tools. По суті, між фреймворками та інструментами немає величезної різниці. Вони обидва взаємопов'язані між собою. Для розробки нового програмного забезпечення або мови ви використовуєте інструменти, а інструменти можуть бути розроблені за допомогою власних фреймворків.

=> Зв'яжіться з нами запропонувати список тут.

Що ви дізнаєтесь:

- Найкращі інструменти та фреймворки для тестування PHP

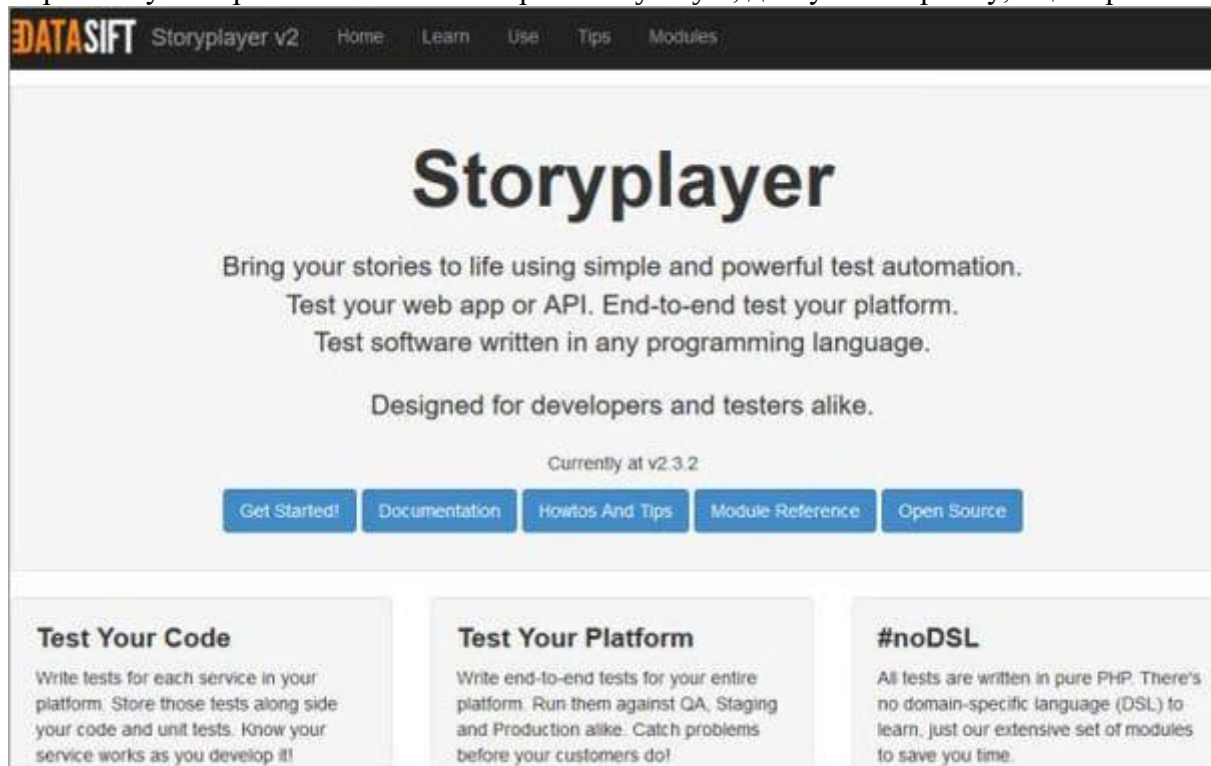
- [# 1\) StoryPlayer](#)
- [# 2\) СеленHQ](#)
- [# 3\) PHPUnit](#)
- [# 4\) Кодецепція](#)
- [# 5\) Простий тест](#)
- [# 6\) Бехат](#)
- [No7\) Огірок](#)
- [# 8\) Атом](#)
- [# 9\) Калан](#)
- [# 10\) Xdebug](#)
- [Висновок](#)
- [Рекомендована література](#)

Найкращі інструменти та фреймворки для тестування PHP

Давайте розпочнемо!!!

1) StoryPlayer

Це інструмент, який використовується для автоматизації як функціональних, так і нефункціональних вимог коду, написаного на PHP. Він побудований Data Sift. Він використовується для тестування як Back-end, так і Front-end ваших веб-сайтів. Він модульний і простий у використанні. Остання версія StoryPlayer, доступна на ринку, - це версія 1.5.



Особливості інструменту:

- Використовується для тестування веб-програми або API.
- Використовується для компонента, а також для наскрізного тестування.
- Тестування наскрізних платформ.
- Призначений як для розробників, так і для тестувальників.
- Можна також використовувати для тестування коду, написаного іншими мовами, крім PHP.
- Працює з веб-браузерами та комбінаціями платформ.
- Можна розширити за допомогою власних плагінів.

Плюси:

- Відкрите джерело.
- Простий і легкий тест і налагодження коду.
- Використовується для крос-браузерного тестування.
- Встановити через Composer.

Мінуси:

- Без встановлення композитора це не спрацює.
- Якщо ви не знаєте PHP, знадобиться час, щоб зрозуміти тестові приклади, розроблені в PHP.

Вартість:

Оскільки це інструмент з відкритим кодом, він доступний на ринку безкоштовно.

Клацніть [тут](#) посилання для завантаження.

2) СеленHQ

Селен - це безкоштовний інструмент автоматизації веб-додатків з відкритим кодом. Його також називають інструментом автоматизації веб-додатків Функціонального тестування. Він підтримує кілька мов, таких як Java, C # (.net), Ruby, Python та PHP тощо.

Селен має різні смаки, і вони:

- # 1) Ядро селену
- # 2) Селен RC (Селен1)
- # 3) Селен IDE
- Selendroid - лише для android.
- Appium - як Android, так і Apple.

SeleniumHQ
Browser Automation

edit this page search selenium: Go

Projects Download Documentation Support About

What is Selenium?

Selenium automates browsers. That's it! What you do with that power is entirely up to you. Primarily, it is for automating web applications for testing purposes, but is certainly not limited to just that. Boring web-based administration tasks can (and should!) also be automated as well.

Selenium has the support of some of the largest browser vendors who have taken (or are taking) steps to make Selenium a native part of their browser. It is also the core technology in countless other browser automation tools, APIs and frameworks.

Which part of Selenium is appropriate for me?

Selenium WebDriver

If you want to

- create robust, browser-based regression automation suites and tests
- scale and distribute scripts across many environments

Selenium IDE

If you want to

- create quick bug reproduction scripts
- create scripts to aid in automation-aided exploratory

Selenium is a suite of tools to automate web browsers across many platforms.

Selenium...

- runs in [many browsers](#) and [operating systems](#)
- can be controlled by many [programming languages](#) and [testing frameworks](#).

[Download Selenium](#)

Особливості інструменту:

- Підтримується усіма операційними системами, такими як Windows, Mac, Linux тощо.
- Тести на регресію можуть бути автоматизовані селеном.
- Найбільш широко використовуваний автоматизований інструмент, що використовується для автоматизації коду на PHP або Java або інших мовах програмування.
- Він має власний API веб-драйвера.

Плюси:

- Вартість низька.
- Перехресне тестування браузера з його сумісністю.
- Підтримує різні рамки тестування.
- Легко інтегрувати з іншою мовою програмування.
- Використовується з TDD (Test Driven Development).
- Корисно для всебічного тестування.

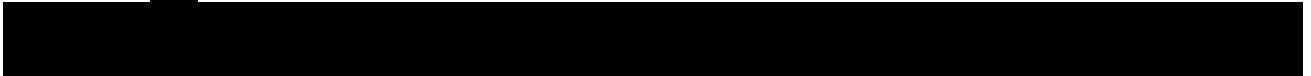
Мінуси:

- Складність у пошуку локаторів.
- Підтримуються лише обмежені браузерери.
- Ви не можете писати ручні сценарії.
- Продуктивність повільна.

Вартість:

Оскільки це інструмент з відкритим кодом, він доступний на ринку безкоштовно.

Клацніть тут посилання для завантаження.



3) PHPUnit

Це інструмент тестування з відкритим кодом, що використовується для PHP-коду.

Це найбільш широко використовувана структура для модульного тестування. Код тестується одиниця за одиницею, щоб з'ясувати помилки та дефекти коду. Процес трохи повільніший, і це займає багато часу, але шанси отримати помилку в кінці менші завдяки модульному тестуванню.



Особливості інструменту :

- Використовується з cmd, тобто командним рядком.
- Ви можете продовжити тестові кейси відповідно до вимог.
- Він використовує твердження коду для перевірки поведінки одиниці.
- Прості та легкі основи модульного тестування.
- Це архітектура xUnit для фреймворку Unit testing.

Плюси:

- Обмежтеся кожною частиною програми для модульного тестування.
- Найбільш широко використовується в організації для модульного тестування.
- Перевірте різні типи контролерів без розширення.
- Створюються різні та декілька типів звітів про випробування.
- Результати тестів налаштовуються.

Мінуси:

- Він не може працювати безпосередньо з веб-браузером.
- Він менш функціональний.
- Для тестування API це не підходить для використання.

Вартість:

Оскільки це інструмент з відкритим кодом, він доступний на ринку безкоштовно.

Клацніть [тут](#) посилання для завантаження.

4) Кодецепція

Codeception - це автоматизована система тестування, яка може бути використана для прийняттого тестування веб-додатків, але також може бути використана для модульного тестування, а також функціонального тестування веб-програм. Швидкість виконання тесту в Codeception є високою, отже, це одна з найбільш бажаних платформ для тестування в PHP.



Особливості інструменту:

- Він простий у використанні і компактний.
- Він підтримує більше одного набору для тестування, тобто приймального, одиничного та функціонального.
- Він працює краще, якщо його інтегрувати з селеном.
- Це робить код легким для читання, запису та налагодження.
- Його також називають BDD (Behaviour Driven Development).

Плюси:

- Багато функцій доступні для тестування низьких рівнів, таких як модульне тестування, або високих рівнів, таких як тестування API або BDD.
- Його можна використовувати з іншими фреймворками також для тестування.
- Він повністю заснований на PHP, тому тестові кейси також написані на ньому, і програмісту для цього не потрібно вивчати різні мови.

Мінуси:

- Налаштування не є простим і простішим.
- Він не має великих ресурсів та документації.
- Він повністю заснований на PHP, тому, якщо розробники не знають PHP, вони не можуть написати тестові кейси для програмного забезпечення.

Вартість:

Оскільки це інструмент з відкритим кодом, він доступний на ринку безкоштовно.

Клацніть [тут](#) для посилання для завантаження.

5) Простий тест

Він використовується як для модуля, так і для веб-тестування. Він підтримує проксі-сервери, SSI, форми, фрейми тощо. Якщо ви використовуєте JUnit, то вам буде простіше використовувати його. Це частина ядра Drupal.

Особливості інструменту :

- Використовується для модульного тестування.
- Підтримує найпростіший HTML-дисплей.
- Тестові кейси - це автоматичне завантаження.
- Обробляє файли cookie під час отримання різних сторінок.
- Відображаються тестові кейси без веб-браузера.
- Підтримує проксі, SSI, форми, фрейми тощо.

Плюси:

- Він перевіряє кожну частину.
- Легко зрозуміти та знайти помилки, якщо виконано модульне тестування.
- Виконання буде швидким.

Мінуси:

- Знання складне.
- Не можна сказати, що кожна частина системи буде працювати правильно.

Вартість:

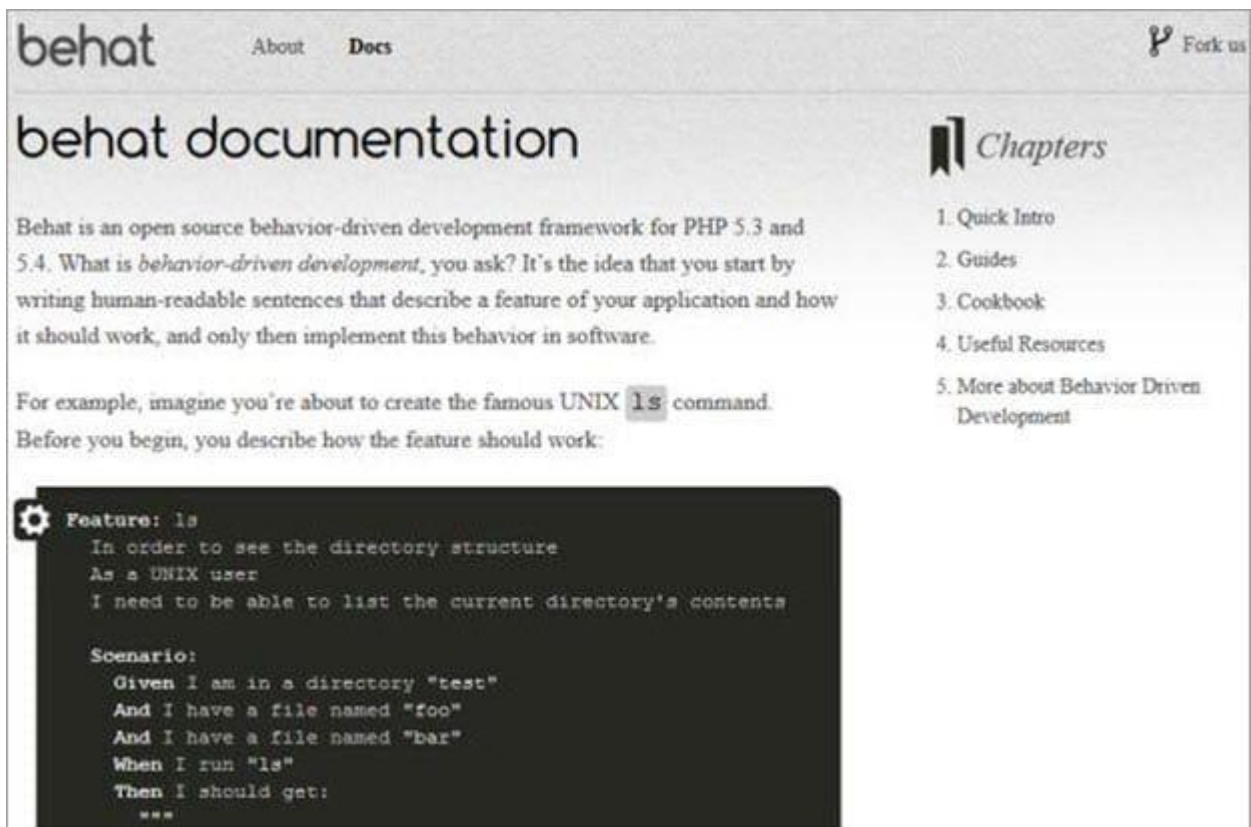
Оскільки це інструмент з відкритим кодом, він доступний на ринку безкоштовно.

Клацніть [тут](#) посилання для завантаження.

6) Бехат

Це фреймворк для тестування PHP-коду. Він використовується для тестування поведінкового розвитку.

Це рамка тестування з відкритим кодом. Він дуже розширюваний, отже, існує безліч розширень. Це дуже легко зрозуміти, оскільки це починається з написання зручних для читання речень.



Особливості інструменту:

- Підтримує BDD для тестування.
- Він представлений мовою, що називається Корнішон, що є зручною для читання.
- Допомогає видалити логічні деталі з тесту поведінки.

Плюси:

- Це BDD і зручно для читання, тому якщо людина, яка не знає мови програмування, також може легко писати функції.
- Обслуговування тестових кейсів простіше і зрозуміліше.

Мінуси:

- Для тестування API йому потрібно декілька інших інструментів для підтримки або інтеграції з ним.
- Програміст повинен розуміти мову корнішона.

Вартість:

Оскільки це інструмент з відкритим кодом, він доступний на ринку безкоштовно для тестувальників та розробників.

Клацніть [тут](#) для посилання для завантаження.

№7) Огірок

Це фреймворк, що керується поведінкою (BDD), який використовується для тестування програмного забезпечення.

Оскільки це BDD, він дуже простий, зрозумілий і зрозумілий для людей. Тестові приклади так легко зрозуміти з огірком. Це ефективний інструмент для тестування. Він написаний мовою програмування Ruby.



Особливості інструменту:

- Плагін для огірка допомагає працювати швидше.
- Це безкоштовний і відкритий код.
- Це механізм розвитку, що керується поведінкою, який зрозумілий для людей.
- Він підтримує різні мови програмування.
- Він підтримує веб- і клієнт-серверні програми.

Плюси:

- Тестові кейси легше зрозуміти, оскільки вони підтримують структуру BDD для тестування.
- Він зосереджений на досвіді кінцевих користувачів.
- Виконання швидко і легко.

Мінуси:

- Не вдається вставити зображення, гіперпосилання тощо.
- Складний завдяки мові програмування Ruby.
- Функція зіставлення об'єкта домену недоступна.

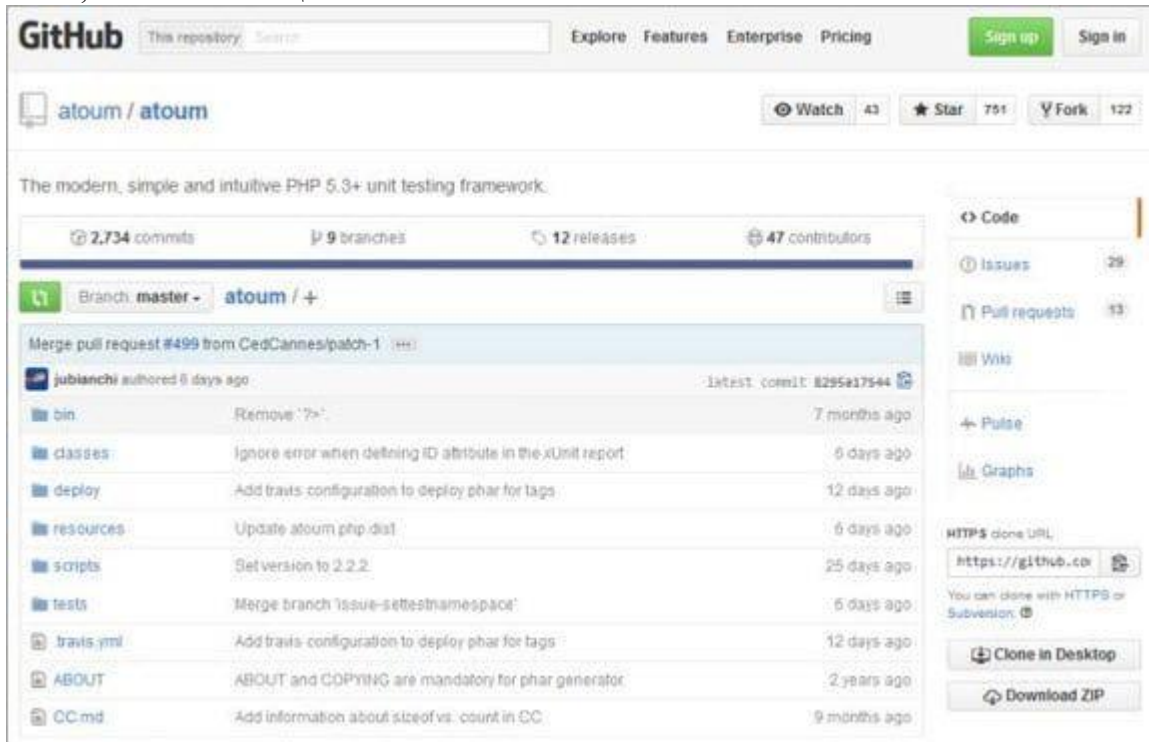
Вартість:

Оскільки це інструмент з відкритим кодом, він доступний на ринку безкоштовно.

Клацніть [тут](#) для посилання для завантаження

8) Атом

Він використовується для модульного тестування. Це простий, легкий та інтуїтивно зрозумілий інструмент для тестування PHP. Рівень безпеки для виконання тестових справ високий в Атомі. Він підтримує фальшиві рідні функції PHP. Він видає різні звіти, такі як TAP, xUnit, конюшина тощо.



Особливості інструменту:

- Він підтримує постачальників даних та автозапуск.
- Тестові приклади виконуються паралельно.
- Легкі, прості та гнучкі конструкції.
- Підтримує змістовні твердження.

Плюси:

- Безпека висока.
- Допомогає писати знуцання.
- Він інтегрований з усіма системами PHP.
- У ньому є безліч розширень, доступних для використання на різних мовах.

Мінуси:

- Знуцання складне.
- Тестує кожну частину програмного забезпечення, але не запевняючи, що кожна частина працює коректно.

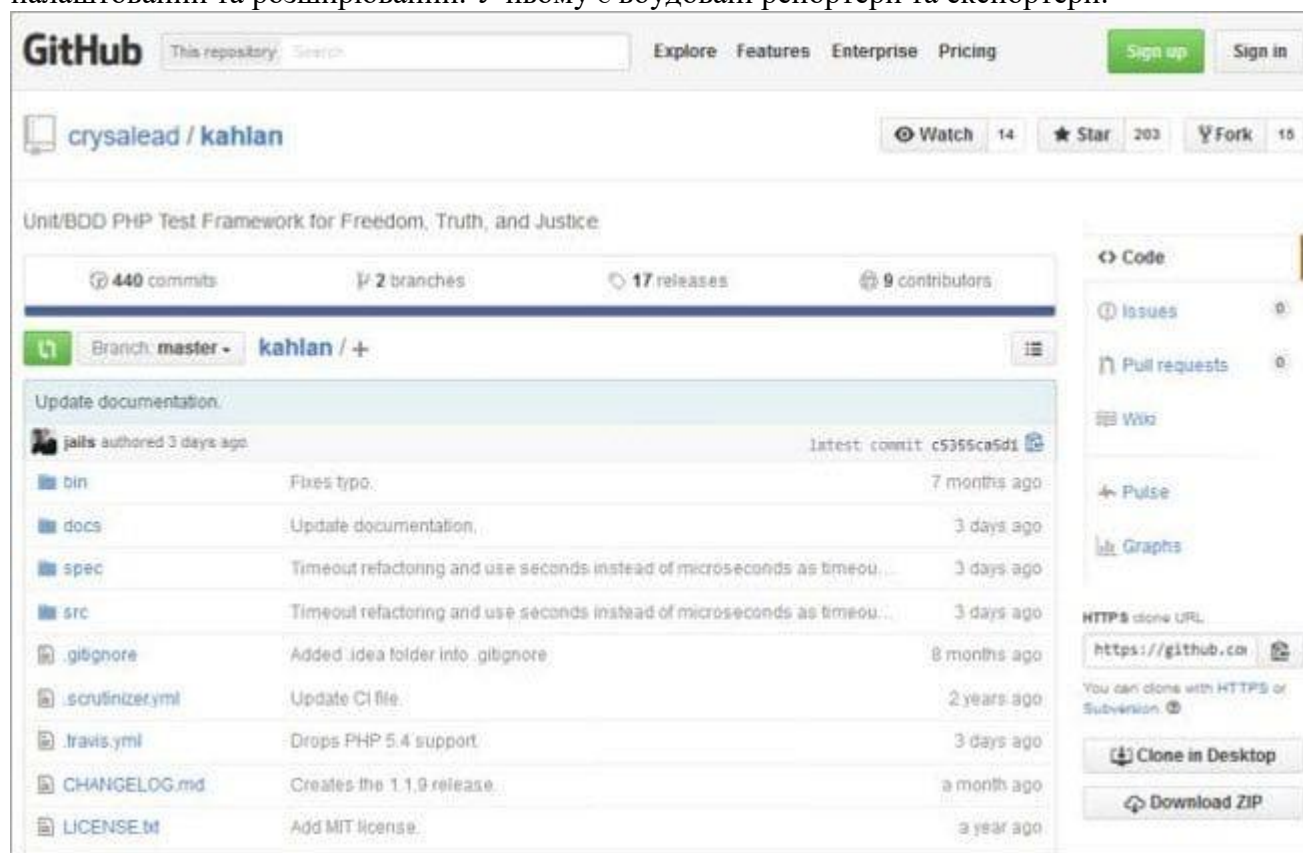
Вартість:

Оскільки це інструмент з відкритим кодом, він доступний на ринку безкоштовно для практичної практики.

Клацніть [тут](#) посилання для завантаження.

9) Калан

Він використовується як для наскрізного функціонального, так і для модульного тестування PHP. Він використовує синтаксис `descri-it`. Це дозволяє динамічно знущатися. Робочий процес налаштований та розширюваний. У ньому є вбудовані репортери та експортери.



Особливості інструменту:

- Підтримує структуру тестування BDD.
- Він підтримує мавповий патч до вашого коду.
- Присутні перевірки, що називаються методами.
- Присутні динамічні знущання, тобто заглушки.
- Для цього потрібні Xdebug та phpdbg.

Плюси:

- Виконано наскрізне функціональне тестування.
- Коди налаштовані та зрозумілі.
- Зрозуміла англійською мовою полегшує розуміння.
- Швидкість виконання висока.
- Це допомагає в семантичному керуванні версіями підтримувати CHANGELOG.

Мінуси:

- Якщо нова, тоді потрібен час, щоб зрозуміти інструмент для тестування.
- Для покриття коду потрібні Xdebug та Phpdbg.
- Він підтримує лише PHP 5.5+.

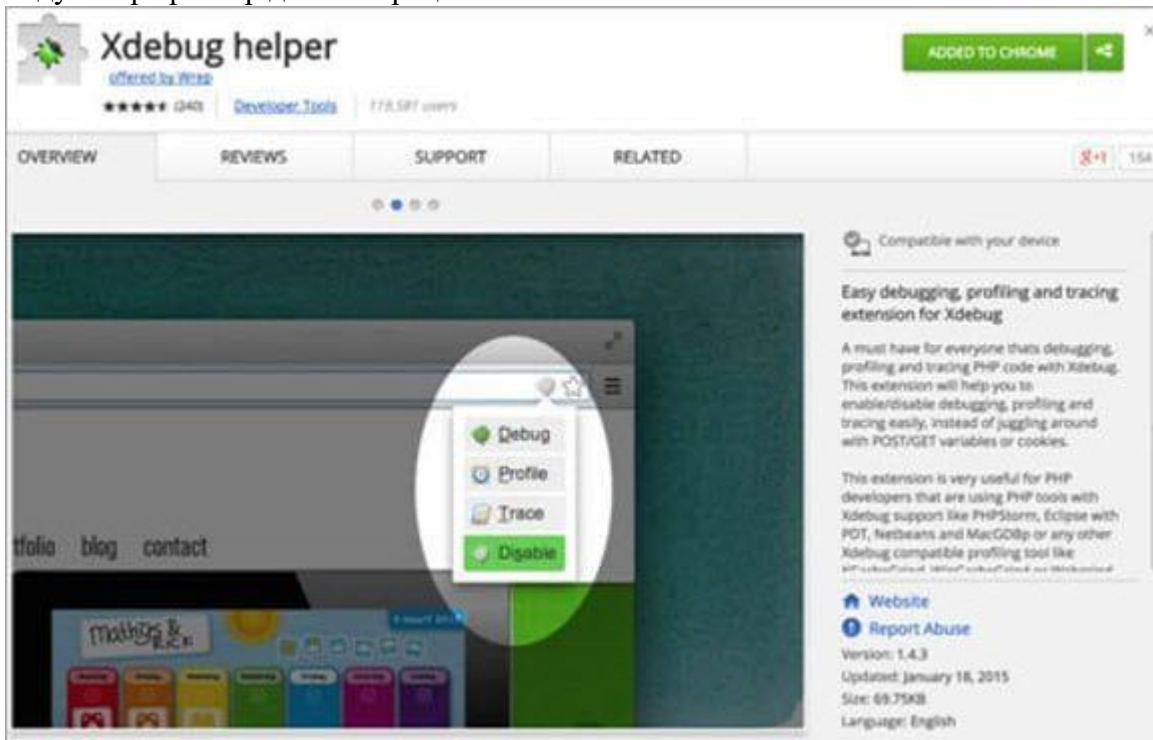
Вартість:

Оскільки це інструмент з відкритим кодом, він доступний на ринку безкоштовно.

Клацніть [тут](#) для посилання для завантаження.

10) Xdebug

Це відкритий код тестування для PHP, який використовується для розробки та налагодження кодів. Для налагодження використовується протокол DBGp. Він підтримує аналіз покриття коду та профайлер для інтеграції з PHPUnit.



Особливості інструменту:

- Коли виникає стан помилки, він відображає сліди стека.
- Використовується для профілювання використання пам'яті.
- Аналіз покриття коду підтримки.
- Використовується для віддаленого налагодження коду.
- Журнали функціонують у іншому форматі файлу.
- Він має вбудовану статистику збору сміття.

Плюси:

- Доступна віддалена налагодження для легкої налагодження коду.
- Установка проста і проста.
- Оптимізація коду краща завдяки профілі.

Мінуси:

- Використовується лише для наскрізного тестування.
- Якщо ви використовуєте розширення Zend, то ви не можете використовувати це з Xdebug.

Вартість:

Оскільки це інструмент з відкритим кодом, він доступний на ринку безкоштовно.

Клацніть [тут](#) для посилання для завантаження.

Окрім вищеописаних інструментів, є також кілька інших інструментів, про які йдеться нижче.

Ще кілька до списку:

11) Перідот

Він використовується для тестування BDD / TDD. Виконання швидко в Peridot.

У ньому є різні плагіни та репортери. Він використовується синтаксисом descri-it, оскільки використовує систему тестування BDD, яка допомагає легше зрозуміти людське речення. Він заснований на SpesBDD. У нього дуже хороші плагіни, завдяки чому його швидкість висока. Він базується на архітектурі на основі подій.

Найкращі та найкращі плагіни - це плагіни Concurrency, за допомогою яких повільні тести також швидко працюють, щоб отримати бажаний результат.

Клацніть [тут](#) для посилання для завантаження.

12) PHP_CodeSniffer

PHP_CodeSniffer використовує два PHP-сценарії.

Перший - це скрипт phpcs, який визначає файли PHP, JavaScript та CSS для виявлення порушень коду, а другий - це скрипт phpcbf, який автоматично виправляє порушення коду.

Клацніть [тут](#) для посилання для завантаження.

13) PHP-детектор безладу

PHPMD - це виділений проект PHP Depend. Це PHP-замінник інструменту Java під назвою PMD.

Він може виявляти помилки, оптимізувати коди, а також знаходити ресурси та параметри, які не є корисними з коду. Він перевіряє складність коду та інтегрується з PHPStorm.

Клацніть [тут](#) для посилання для завантаження.

14) PHPSpec

PHPSpec використовується для поведінкової розробки (BDD) коду. Він використовується для модульного тестування.

SpecBDD використовується на рівні специфікації PHPSpec. Він розробляє код і допомагає знати про процес SpecBDD за допомогою командного рядка.

Клацніть [тут](#) для посилання для завантаження.

15) FirePHP

FirePHP використовується для розробки консольних текстів у браузері за допомогою коду PHP. FirePHP надсилає тексти того ж типу, що і заголовки HTTP. Існують розширення, які виступають проксі-сервером для JavaScript та перетворюють заголовки HTTP у тексти журналів.

Клацніть [тут](#) для посилання для завантаження.

Висновок

У цій статті ми обговорили найкращі основи та інструменти тестування PHP, які доступні на ринку, а також їх ціни, плюси, мінуси та особливості. Ці інструменти в основному засновані на розробці, що керується поведінкою, або тестовій розробці, тобто для модульного тестування та наскрізного тестування.

Існує ще декілька програм тестування PHP та інструментів, які доступні на ринку та є відкритими, але вищезазначені інструменти є найбільш популярними серед усіх інструментів.

=> **Зв'яжіться з нами запропонувати список тут.**

Рекомендована література

- [Найкращі засоби тестування програмного забезпечення 2021 р. \[Інструменти автоматизації тестування якості\]](#)
- [10 найпопулярніших засобів тестування регресії в 2021 році](#)
- [25 найкращих платформ для тестування Java та інструментів для автоматичного тестування \(частина 3\)](#)
- [Тестування навантаження за допомогою LoadUI - безкоштовний інструмент тестування навантаження з відкритим кодом](#)
- [11 найкращих засобів автоматизації для тестування програм для Android \(Інструменти для тестування додатків Android\)](#)
- [Інструменти та послуги для тестування продуктивності веб-сайтів](#)
- [Топ 15+ найпопулярніших інструментів тестування веб-служб у 2021 році](#)
- Завантажити тестувальник електронних книг

Команда Знайти в Unix: Знайти файли за допомогою файлу пошуку Unix (Приклади)
Find Command Unix

Вступ до команди пошуку в Unix: пошук файлів і каталогів за допомогою команди пошуку файлів Unix

Команда Unix find - це потужна утиліта для пошуку файлів або каталогів.

Пошук може базуватися на різних критеріях, а відповідні файли можна запускати за допомогою визначених дій. Ця команда рекурсивно спускає ієрархію файлу для кожного вказаного імені шляху.

=> [Клацніть тут, щоб переглянути повну серію підручників Unix](#)



Що ви дізнаєтесь:

- [Знайдіть команду в Unix](#)
- [Приклади](#)
- [Висновок](#)
- [Рекомендована література](#)

Знайдіть команду в Unix

Синтаксис:

`find [options] [paths] [expression]`

Параметри цієї команди використовуються для визначення способу обробки символічних посилань. Далі слідує набір шляхів для пошуку. Якщо шляхи не вказані, використовується поточний каталог. Потім даний вираз виконується на кожному з файлів, знайдених у шляхах. Вираз складається з ряду опцій, тестів та дій, кожен з яких повертає логічну форму. Вираз обчислюється зліва направо для кожного файлу у шляху, доки не буде визначений результат, тобто результат буде відомим як true або false.

- Вирази опцій використовуються для обмеження операції пошуку і завжди повертають true.
 -
 - -depth: обробляти вміст каталогу перед обробкою самого каталогу.
 - -maxdepth: максимальні рівні нижче від наданих шляхів, щоб спуститися на відповідність.
 - -mindepth: мінімальні рівні поза наданими шляхами, що спускаються до зіставлення.
- Тестові вирази використовуються для оцінки конкретних властивостей файлів і відповідного повернення true або false. (Де б не використовувався підрахунок 'n': без будь-якого префікса збіг відповідає точному значенню n; з префіксом '+' збіг для значень, більших за n; а з префіксом '-', збіг для значень, менших за n.)
 -

- -atime n: Повертає true, якщо файл було відкрито n днів тому.
- -ctime n: Повертає true, якщо статус файлу було змінено n днів тому.
- -mtime n: Повертає true, якщо вміст файлу було змінено n днів тому.
- -name pattern: Повертає true, якщо ім'я файлу відповідає наданому шаблону оболонки.
- -iname pattern: Повертає true, якщо ім'я файлу відповідає наданому шаблону оболонки. Тут відповідність не враховує регістр.
- -path pattern: Повертає true, якщо ім'я файлу із шляхом відповідає шаблону оболонки.
- -regex pattern: Повертає true, якщо ім'я файлу із шляхом відповідає регулярному виразу.
- -size n: Повертає true, якщо розмір файлу - n блоків.
- -perm - mode: Повертає true, якщо для файлу встановлені всі біти дозволу для режиму.
- -type c: Повертає true, якщо файл має тип c (наприклад, 'b' для файлу блочного пристрою, 'd' для каталогу тощо).
- -username: Повертає true, якщо файл належить імені користувача „name“.
- Вирази дії використовуються для визначення дій, які мають побічні ефекти і можуть повернути true або false. Якщо не вказані дії, дія „-print” виконується для всіх відповідних файлів.
-
- -delete: Видалить відповідний файл і в разі успіху повернеть true.
- -exec команда: Виконайте задану команду для кожного відповідного файлу та повернеть true, якщо значення повернення 0.
- -ok команда: Подібно виразу ‘exec’, але спочатку підтверджує з користувачем.
- -ls: Перелічить відповідний файл у форматі ‘ls -dils’.
- -print: Надрукуйте ім'я відповідного файлу.
- -prune: Якщо файл є каталогом, не спускайтеся в нього, а повертайте true.
- Вираз обчислюється зліва направо і складається разом з наступними операторами.
-
- (вираз) : Використовується для встановлення пріоритету.
- ! expr: Використовується для заперечення виразу.
- вираз1 -до expr2: Результатом є 'і' двох виразів. Expr2 обчислюється лише, якщо expr1 є true.
- expr1 expr2: У цьому випадку оператор ‘і’ є неявним.
- вираз1 -або expr2: Результат - це «або» з двох виразів. Вираз expr2 обчислюється лише, якщо expr1 є помилковим.

Приклади

Перелічить усі файли, знайдені в поточному каталозі, та його ієрархію

```
$ find.
```

Перелічить усі файли, знайдені в поточній ієрархії, та всю ієрархію нижче / home / xyz

```
$ find. /home/XYZ
```

Шукайте файл за назвою abc у поточному каталозі та його ієрархії

```
$ find ./ -name abc
```

Шукайте каталог за назвою xyz у поточному каталозі та його ієрархії

```
$ find ./ -type d -name xyz
```

Знайдіть файл під назвою abc.txt під поточним каталогом та запропонуйте користувачеві видалити кожну відповідність.

Зверніть увагу, що рядок “{ }” під час запуску замінюється фактичною назвою файлу, а “;” рядок використовується для завершення виконуваної команди.

```
$ find ./ -name abc.txt -exec rm -i { } ;
```

Шукайте файли, які були змінені за останні 7 днів у поточному каталозі

```
$ find ./ -mtime -7
```

Шукайте файли, які мають усі дозволи, встановлені в поточній ієрархії

```
$ find ./ -perm 777
```

