What is the best way to use a console when developing?

Asked 16 years, 3 months ago   Modified 7 years, 3 months ago

Viewed 335 times

2

For scripting languages, what is the most effective way to utilize a console when developing? Are there ways to be more productive with a console than a "compile and run" only language?

Added clarification: I am thinking more along the lines of Ruby, Python, Boo, etc. Languages that are used for full blown apps, but also have a way to run small snippets of code in a console.

scripting   console

Share

Improve this question

Follow

edited Aug 28, 2008 at 15:28

asked Aug 28, 2008 at 15:13

bmavity
2,508 ● 2 ● 24 ● 23

You might be surprised what languages are used for full-blown applications. – Jon Ericson Sep 9, 2008 at 21:48

5 Answers

Sorted by: Highest score (default) ⇕

> I am thinking more along the lines of Ruby, ...

2

Well for Ruby the `irb` interactive prompt is a great tool for "practicing" something simple. Here are the things I'll mention about the irb to give you an idea of effective use:

- *Automation*. You are allowed a `.irbrc` file that will be automatically executed when launching irb. That means you can load your favorite libraries or do *whatever* you want in full Ruby automatically. To see what I mean check out some of the ones at dotfiles.org.

- *Autocompletion*. That even makes writing code easier. Can't remember that string method to remove newlines? `"".ch<tab>` produces chop and chomp. *NOTE: you have to enable autocompletion for irb yourself*

- *Divide and Conquer*. irb makes the small things really easy. If you're writing a function to manipulate strings, the ability to test the code interactively right in the prompt saves a lot of time! For instance you can just open up irb and start running functions on an

example string and have working and tested code already ready for your library/program.

- *Learning, Experimenting, and Hacking*. Something like this would take a very long time to test in C/C++, even Java. If you tried testing them all at once you might seg-fault and have to start over.

  Here I'm just learning how the `String#[]` function works.

  ```
  joe[~]$ irb
  >> "12341:asdf"[/\d+/]
  # => "12341"
  >> "12341:asdf"[/\d*/]
  # => "12341"
  >> "12341:asdf"[0..5]
  # => "12341:"
  >> "12341:asdf"[0...5]
  # => "12341"
  >> "12341:asdf"[0, ':']
  TypeError: can't convert String into Integer
    from (irb):5:in `[]'
    from (irb):5
  >> "12341:asdf"[0, 5]
  # => "12341"
  ```

- *Testing and Benchmarking*. Now they are nice and easy to perform. [Here](#) is someone's idea to emulate the Unix `time` function for quick benchmarking. Just add it to your `.irbrc` file and its always there!

- *Debugging* - I haven't used this much myself but there is always the ability to debug code [like this](#). Or pull out some code and run it in the irb to see what its actually doing.

I'm sure I'm missing some things but I hit on my favorite points. You really have zero limitation in shells so you're limited only by what you can think of doing. I almost always have a few shells running. Bash, Javascript, and Ruby's irb to name a few. I use them for a lot of things!

Share Improve this answer

Follow

edited Sep 13, 2017 at 6:16

Vineet Jain
1,565 ● 4 ● 21 ● 32

answered Aug 28, 2008 at 16:11

Joseph Pecoraro
2,878 ● 1 ● 20 ● 25

▲

1

▼

🔖

🕓

I think it depends on the console. The usefulness of a CMD console on windows pails in comparison to a Powershell console.

Share Improve this answer

Follow

answered Aug 28, 2008 at 15:18

EBGreen
37.6k ● 12 ● 68 ● 86

▲

1

▼

🔖

You didn't say what OS you're using but on Linux I been using a tabbed window manager (wmii) for a year or so and it has radically changed the way I use applications - consoles or otherwise.
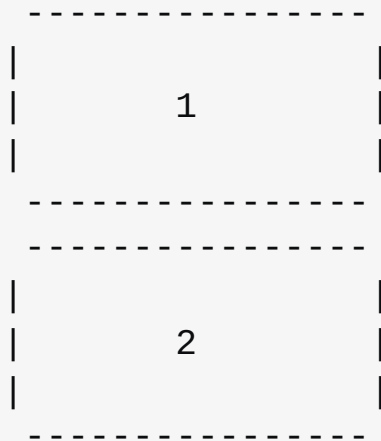
I often have four or more consoles and other apps on a virtual desktop and with wmii I don't have to fiddle with
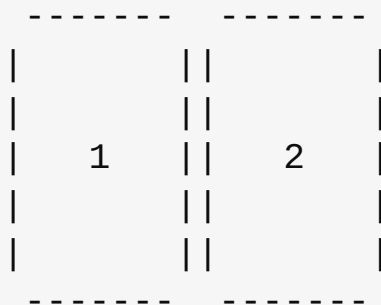
resizing windows to line everything up just so. I can trivially rearrange them into vertical columns, stack them up vertically, have them share equal amounts of vertical or horizontal space, and move them between screens.
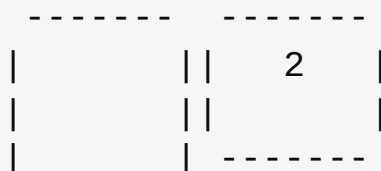
Say you open two consoles on your desktop. You'd get this (with apologies for the cronkey artwork):
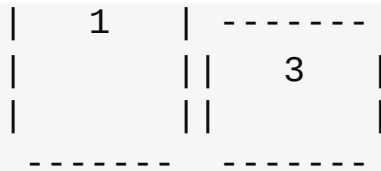
```
 - - - - - - - - - - - - - - -
|                           |
|             1             |
|                           |
 - - - - - - - - - - - - - -
 - - - - - - - - - - - - - -
|                           |
|             2             |
|                           |
 - - - - - - - - - - - - - -
```

Now I want them side-by-side. I enter SHIFT-ALT-L in window 2 to move it rightwards and create two columns:

```
 - - - - - -   - - - - - -
|           | |           |
|           | |           |
|     1     | |     2     |
|           | |           |
|           | |           |
 - - - - - -   - - - - - -
```

Now I could open another console and get

```
 - - - - - -   - - - - - -
|           | |     2     |
|           | |           |
|           | |    - - - - - -
```

```
|   1   | -------
|       ||   3   |
|       ||       |
 -------  -------
```

Then I want to temporarily view console 3 full-height, so I hit ALT-s in it and get:

```
 -------  -------
|       | -------
|       ||       |
|   1   ||   3   |
|       ||       |
|       ||       |
 -------  -------
```
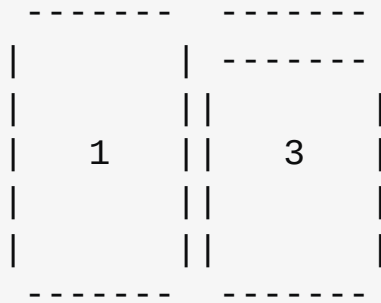
Consoles 2 and 3 are stacked up now.

I could also give windows tags. For example, in console 2 I could say ALT-SHIFT-twww+dev and that console would be visible in the 'www' and 'dev' virtual desktops. (The desktops are created if they don't already exist.) Even better, the console can be in a different visual configuration (e.g., stacked and full-screen) on each of those desktops.

Anyway, I can't do tabbed window managers justice here. I don't know if it's relevant to your environment but if you get the chance to try this way of working you probably won't look back.

Share Improve this answer

Follow

answered Aug 30, 2008 at 22:59

asussex

163 ● 3 ● 5

▲

0

▼

I've added a shortcut to my Control-Shift-C key combination to bring up my Visual Studio 2008 Console. This alone has saved me countless seconds when needing to register a dll or do any other command. I imagine if you leverage this with another command tool and you may have some massive productivity increases.

Share Improve this answer

Follow

answered Aug 28, 2008 at 15:27

Pat
5,282 ● 1 ● 38 ● 53

▲

0

▼

Are you kidding?

In my Linux environment, the console is my lifeblood. I'm proficient in bash scripting, so to me a console is very much like sitting in a REPL for Python or Lisp. You can quite literally do anything.

I actually write tools used by my team in bash, and the console is the perfect place to do that development. I really only need an editor as a backing store for things as I figure them out.

Share Improve this answer

Follow

answered Aug 28, 2008 at 15:30

Ben Collins
20.7k ● 18 ● 128 ● 190