Tiling Window Managers

Aline Abler

TheAlternative.ch
for a sustainable digital world

April 4, 2016

What will we learn today?

- What is a window manager?
- What makes it tiling?
- Why is tiling cool?
- How does it work?
- How do I put it on my system?
- Which one should I use?

Well, it manages windows

You already have one

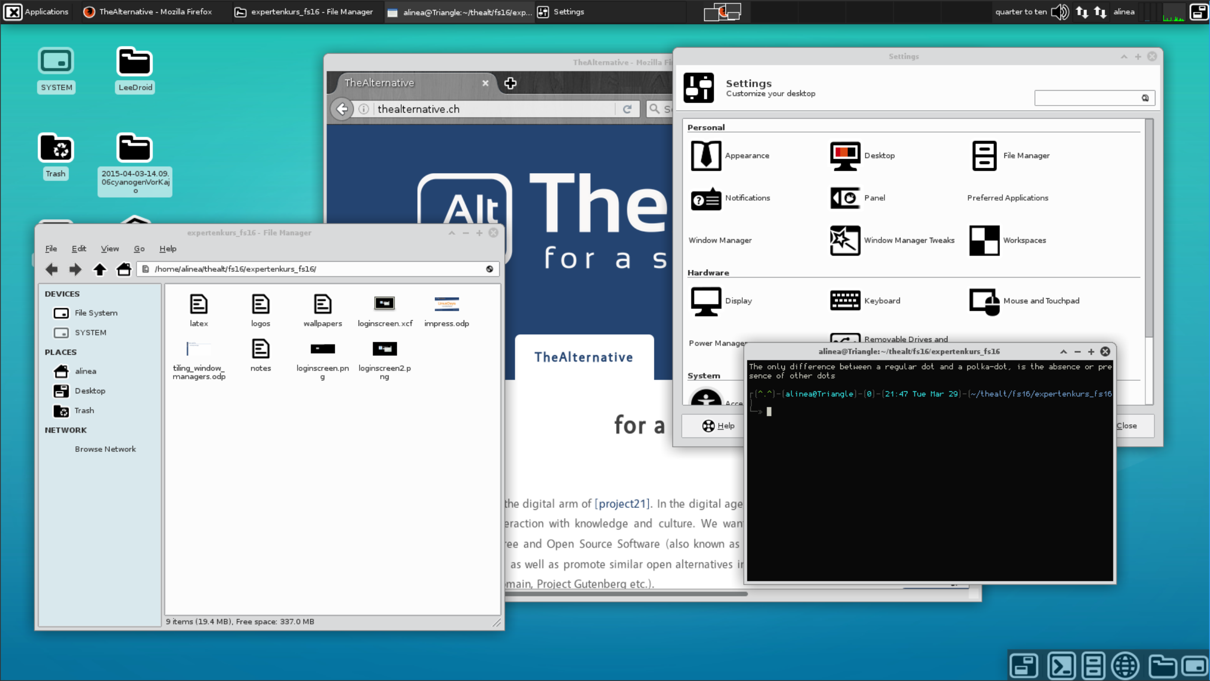
Stacking Window Managers

Each window is freely draggable and resizable

When do we call it tiling?

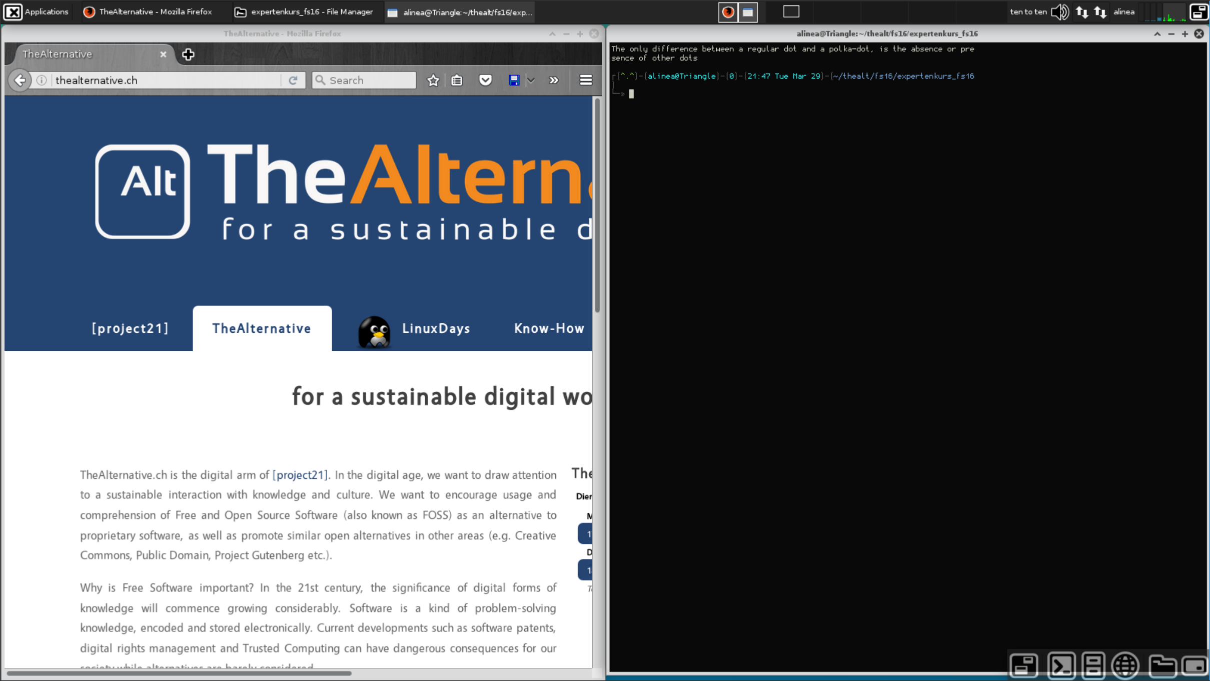
- Windows are arranged for you
- They always take up the entire screen
- You always see all of them

How is that better?



the digital arm of [project21]. In the digital age
eraction with knowledge and culture. We wan
ree and Open Source Software (also known as
as well as promote similar open alternatives i
omain, Project Gutenberg etc.).

```
The only difference between a regular dot and a polka-dot, is the absence or pre
sence of other dots

┌[^.^]─[alinea@Triangle]─[0]─[21:47 Tue Mar 29]─[~/thealt/fs16/expertenkurs_fs16
└>
```

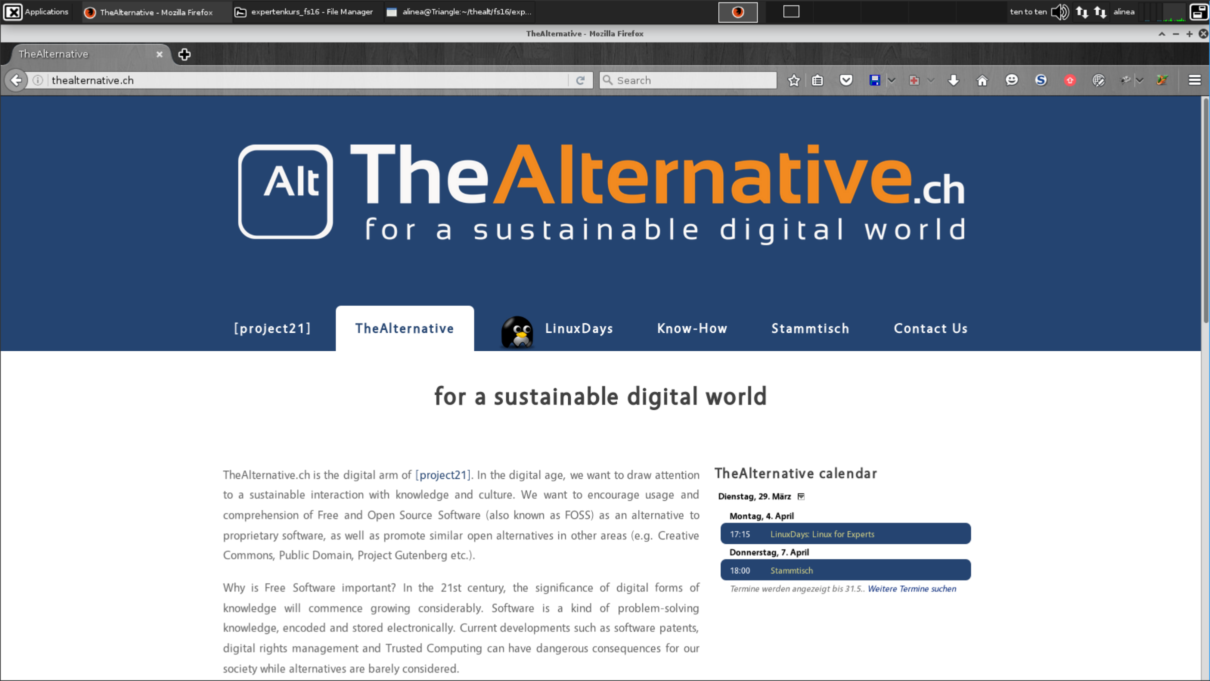

Applications
TheAlternative - Mozilla Firefox
expertenkurs_fs16 - File Manager
alinea@Triangle:~/thealt/fs16/exp...
ten to ten
alinea
TheAlternative - Mozilla Firefox
TheAlternative
thealternative.ch
Search
Alt
TheAltern
for a sustainable d
[project21]
TheAlternative
LinuxDays
Know-How
alinea@Triangle:~/thealt/fs16/expertenkurs_fs16

for a sustainable digital wo

TheAlternative.ch is the digital arm of [project21]. In the digital age, we want to draw attention to a sustainable interaction with knowledge and culture. We want to encourage usage and comprehension of Free and Open Source Software (also known as FOSS) as an alternative to proprietary software, as well as promote similar open alternatives in other areas (e.g. Creative Commons, Public Domain, Project Gutenberg etc.).

Why is Free Software important? In the 21st century, the significance of digital forms of knowledge will commence growing considerably. Software is a kind of problem-solving knowledge, encoded and stored electronically. Current developments such as software patents, digital rights management and Trusted Computing can have dangerous consequences for our

```
The only difference between a regular dot and a polka-dot, is the absence or pre
sence of other dots

┌(^.^)─(alinea@Triangle)─(0)─(21:47 Tue Mar 29)─(~/thealt/fs16/expertenkurs_fs16
└─> █
```


Applications
TheAlternative - Mozilla Firefox
expertenkurs_fs16 - File Manager
alinea@Triangle:~/thealt/fs16/exp...
ten to ten
alinea
TheAlternative - Mozilla Firefox
TheAlternative
thealternative.ch
Search
Alt
TheAlternative.ch
for a sustainable digital world
[project21]
TheAlternative
LinuxDays
Know-How
Stammtisch
Contact Us

for a sustainable digital world

TheAlternative.ch is the digital arm of [project21]. In the digital age, we want to draw attention to a sustainable interaction with knowledge and culture. We want to encourage usage and comprehension of Free and Open Source Software (also known as FOSS) as an alternative to proprietary software, as well as promote similar open alternatives in other areas (e.g. Creative Commons, Public Domain, Project Gutenberg etc.).

Why is Free Software important? In the 21st century, the significance of digital forms of knowledge will commence growing considerably. Software is a kind of problem-solving knowledge, encoded and stored electronically. Current developments such as software patents, digital rights management and Trusted Computing can have dangerous consequences for our society while alternatives are barely considered.

TheAlternative calendar

Dienstag, 29. März

Montag, 4. April

17:15 LinuxDays: Linux for Experts

Donnerstag, 7. April

18:00 Stammtisch

Termine werden angezeigt bis 31.5.. *Weitere Termine suchen*

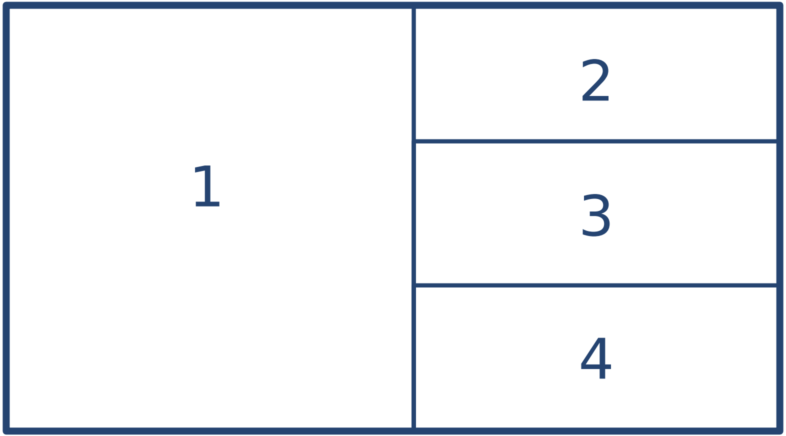
Tiling approaches

List vs. Tree

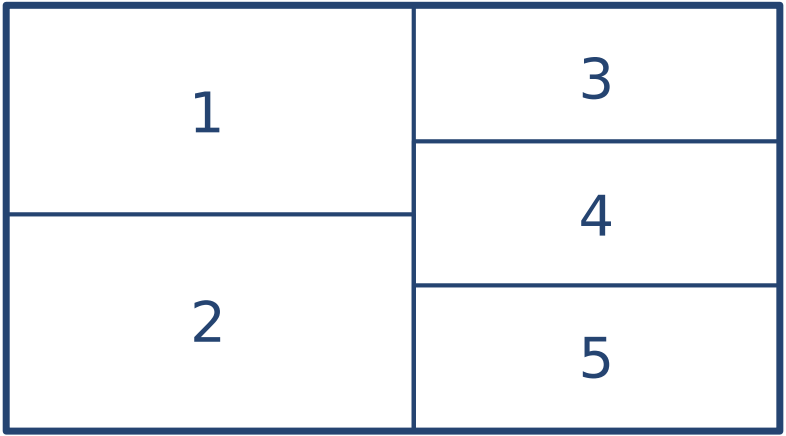
List based tiling

- Windows are internally represented as ordered list
- Arrangement is based on their positions in the list
- Numerous ways to do this

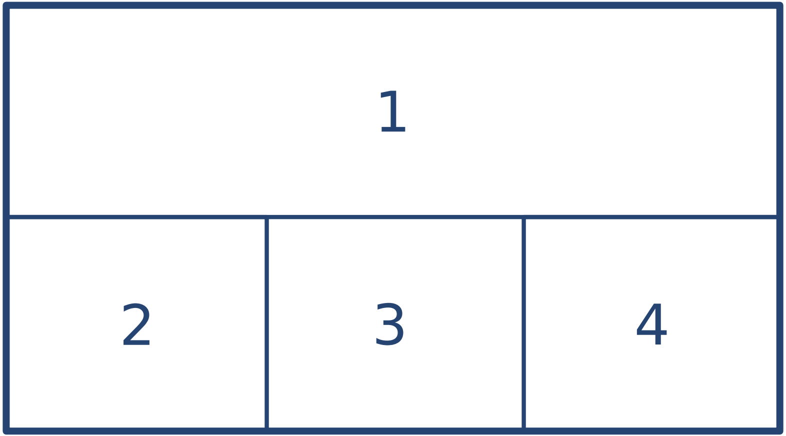
Stack



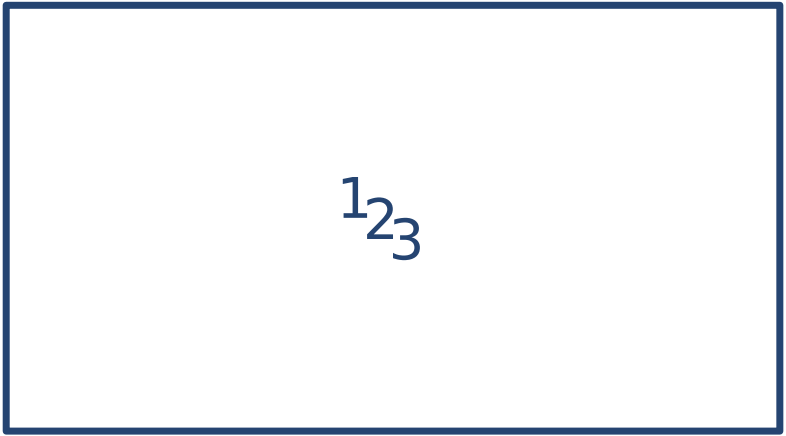
nStack



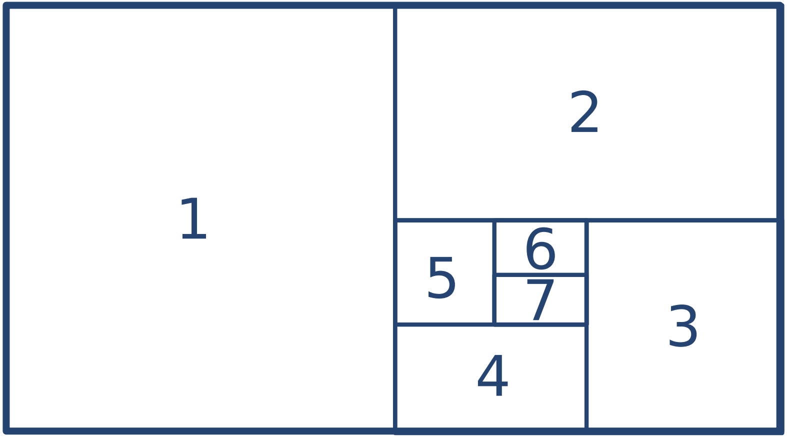
hStack



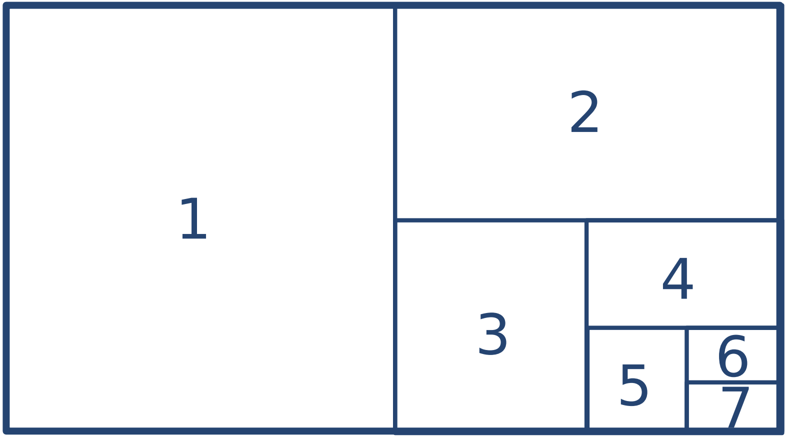
Max / Full / Monocle



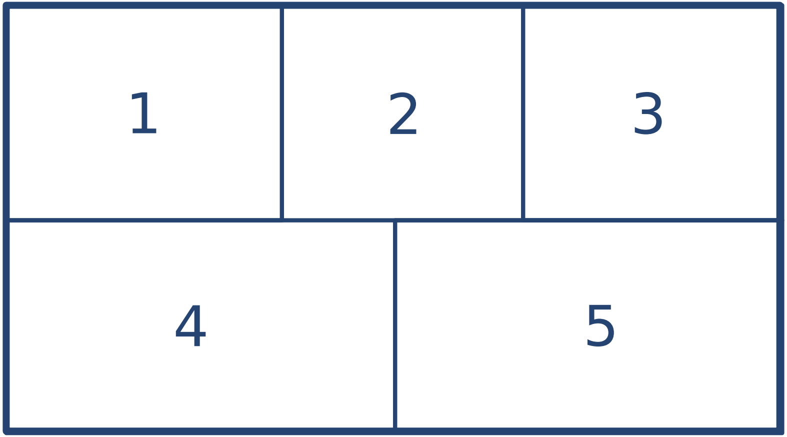
Spiral



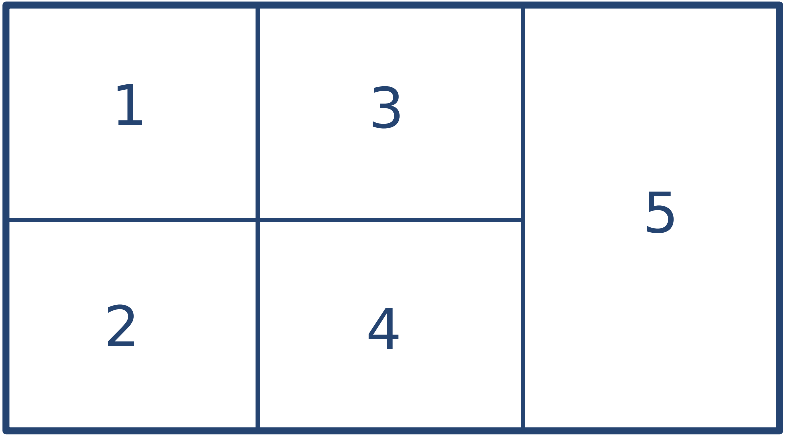
Dwindle



hGrid



vGrid



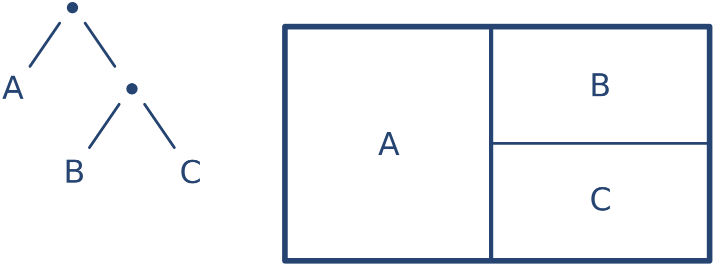
List based tiling

- Easy to change layout
- Rather unflexible

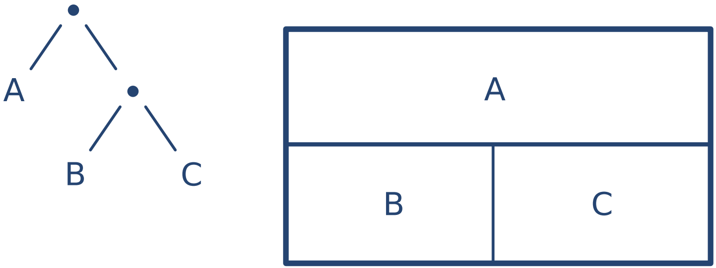
Tree based tiling

- Windows are internally represented as leaves of a tree
- Think of nested containers
- Each internal node can have its own layout

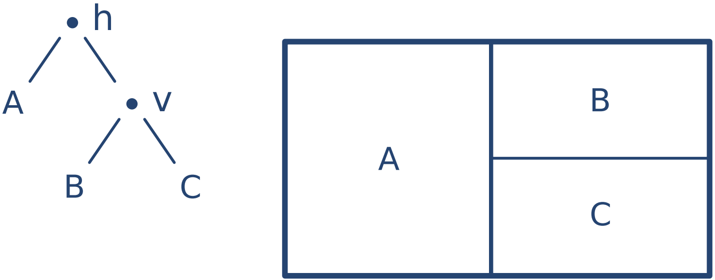
Tree



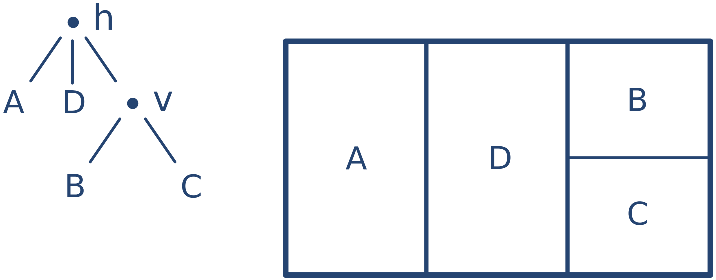
Tree



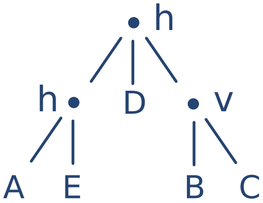
Tree

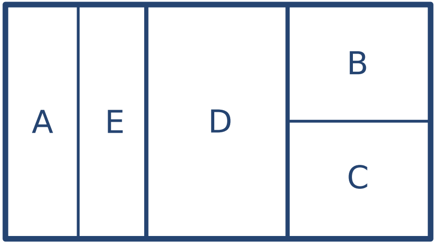


Tree

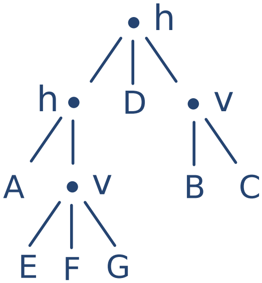


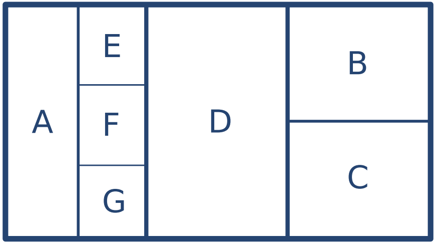
Tree





Tree





Tree based tiling

- A lot more flexible
- More complicated to modify layout

Restrictions

A Tiling Window Manager is not a full-blown Desktop Environment

- No status bar, no workspace overview...
- You will need additional software for this

What you already have

- File manager
- Terminal emulator
- Text editor
- Document viewer, Image viewer, Media player
- Login manager

What you need

- Launcher: dmenu, Kupfer
- Status bar: lemonbar, dzen, Tint2
- System tray: stalonetray
- Notification service: dunst, statnot, twmn
- Lock screen: i3lock, slock
- Something to set your wallpaper: feh, nitrogen

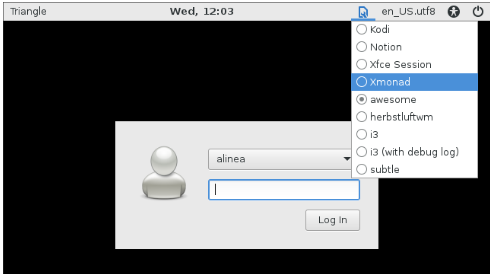
But wait . . .

Some Window Managers already include some of these features.

feature-rich vs. minimal

Installation and Setup

Use your favorite package manager



What distinguishes Window Managers?

- Tiling algorithm: list vs. tree
- Features: feature-rich vs. minimal
- Multihead: What happens when I connect a second monitor?
- Workspaces: Created dynamically or statically? Tags?
- Configuration: What language?
- Other: scriptability, restoring layout

Window Manager Demo

- bspwm
- dwm
- xmonad
- i3
- awesome
- herbstluftwm

bspwm

- Tiling algorithm: Binary tree
- Features: so minimal it doesn't even handle keyboard shortcuts by itself
- Multihead: Each monitor has a separate set of workspaces
- Workspaces: Created statically
- Configuration: in Bash, easy to use
- Other: scriptable through bspc, pretty

dwm

- Tiling algorithm: List
- Features: minimal
- Multihead: Each tag is used on each monitor
- Workspaces: Fixed amount of tags, windows can be assigned multiple tags
- Configuration: in C, need to recompile
- Other: abundance of patches to suit any need; takes some time and skill to set up right

xmonad

- Tiling algorithm: List
- Features: minimal, but ready-to-use extensions available
- Multihead: Workspaces independent of monitors
- Workspaces: created dynamically
- Configuration: in Haskell, which can be confusing but is very powerful

i3

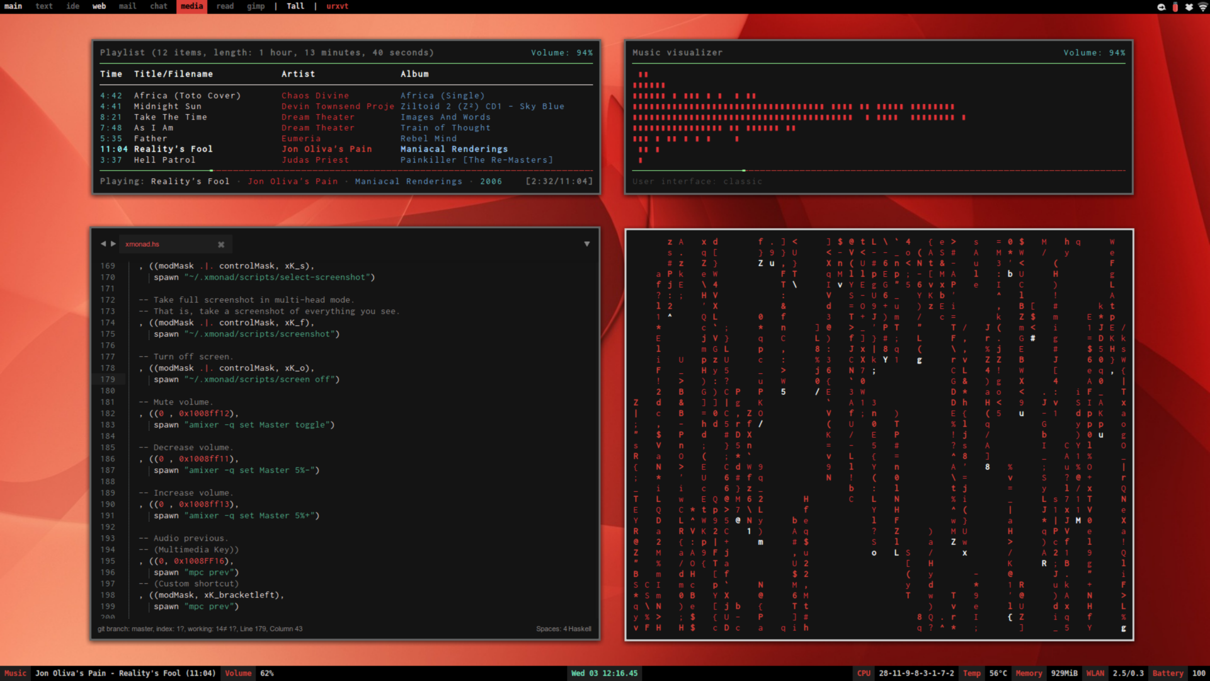
- Tiling algorithm: Tree with additional 'tab' and 'stack' options
- Features: feature-rich
- Multihead: Workspaces belong to monitors but are still independent
- Workspaces: Created dynamically
- Configuration: in custom syntax which is easy enough and well documented
- Other: scriptable through i3-msg, can save and restore layouts

awesome

- Tiling algorithm: List
- Features: extremely feature-rich
- Multihead: Each tag is used on each monitor
- Workspaces: Fixed amount of tags, windows can be assigned multiple tags
- Configuration: in Lua, programming experience is handy

herbstluftwm

- Tiling algorithm: Tree-based frames which contain list-based layouts
- Features: minimal
- Multihead: Workspaces independent of monitors
- Workspaces: Created statically
- Configuration: in Bash, easy to use
- Other: scriptable through herbstclient



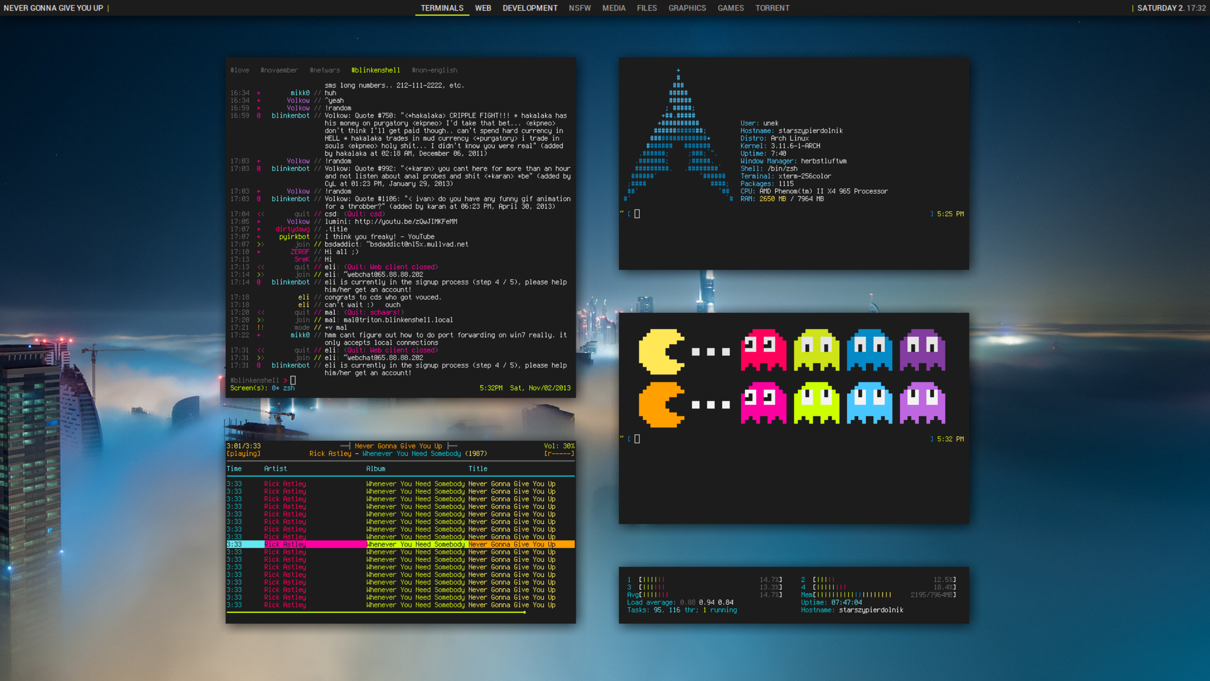
```
Playlist (12 items, length: 1 hour, 13 minutes, 40 seconds)                    Volume: 94%

Time  Title/Filename                  Artist                      Album

4:42  Africa (Toto Cover)             Chaos Divine                Africa (Single)
4:41  Midnight Sun                    Devin Townsend Proje Ziltoid 2 (Z²) CD1 - Sky Blue
8:21  Take The Time                   Dream Theater               Images And Words
7:48  As I Am                         Dream Theater               Train of Thought
5:35  Father                          Eumeria                     Rebel Mind
11:04 Reality's Fool                  Jon Oliva's Pain            Maniacal Renderings
3:37  Hell Patrol                     Judas Priest                Painkiller [The Re-Masters]

Playing: Reality's Fool · Jon Oliva's Pain · Maniacal Renderings · 2006    [2:32/11:04]
```

```
Music visualizer                                                                Volume: 94%

User interface: classic
```

```haskell
  , ((modMask .|. controlMask, xK_s),
    spawn "~/.xmonad/scripts/select-screenshot")

  -- Take full screenshot in multi-head mode.
  -- That is, take a screenshot of everything you see.
  , ((modMask .|. controlMask, xK_f),
    spawn "~/.xmonad/scripts/screenshot")

  -- Turn off screen.
  , ((modMask .|. controlMask, xK_o),
    spawn "~/.xmonad/scripts/screen off")

  -- Mute volume.
  , ((0 , 0x1008ff12),
    spawn "amixer -q set Master toggle")

  -- Decrease volume.
  , ((0 , 0x1008ff11),
    spawn "amixer -q set Master 5%-")

  -- Increase volume.
  , ((0 , 0x1008ff13),
    spawn "amixer -q set Master 5%+")

  -- Audio previous.
  -- (Multimedia Key))
  , ((0, 0x1008FF16),
    spawn "mpc prev")
  -- (Custom shortcut)
  , ((modMask, xK_bracketleft),
    spawn "mpc prev")
```

```
git branch: master, index: 1?, working: 14# 1?, Line 179, Column 43          Spaces: 4 Haskell
```

```
Music Jon Oliva's Pain - Reality's Fool (11:04) Volume 62%      Wed 03 12:16.45      CPU 28-11-9-8-3-1-7-2 Temp 56°C Memory 929MiB WLAN 2.5/0.3 Battery 100
```



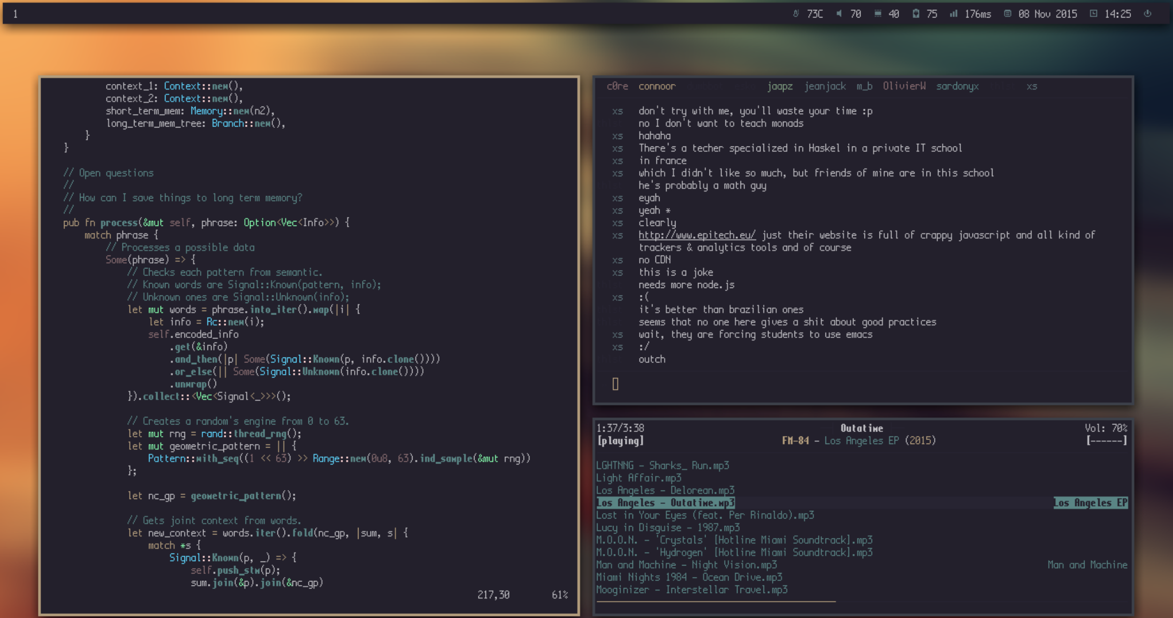
```
#love   #novaember   #netwars   #blinkenshell   #non-english

                             sms long numbers.. 212-111-2222, etc.
16:34 +        mikk0 // huh
16:34 +       Volkow // ^yeah
16:59 +       Volkow // !random
16:59 @   blinkenbot // Volkow: Quote #750: "<+hakalaka> CRIPPLE FIGHT!!! * hakalaka has
                        his money on purgatory <ekpneo> I'd take that bet... <ekpneo>
                        don't think I'll get paid though.. can't spend hard currency in
                        HELL * hakalaka trades in mud currency <+purgatory> i trade in
                        souls <ekpneo> holy shit... I didn't know you were real" (added
                        by hakalaka at 02:18 AM, December 06, 2011)
17:03 +       Volkow // !random
17:03 @   blinkenbot // Volkow: Quote #992: "<+karan> you cant here for more than an hour
                        and not listen about anal probes and shit <+karan> *be" (added by
                        CyL at 01:23 PM, January 29, 2013)
17:03 +       Volkow // !random
17:03 @   blinkenbot // Volkow: Quote #1106: "< ivan> do you have any funny gif animation
                        for a throbber?" (added by karan at 06:23 PM, April 30, 2013)
17:04 <<        quit // csd: (Quit: csd)
17:05 +       Volkow // lumini: http://youtu.be/zQwJIMKFeMM
17:07 +    dirtydawg // .title
17:07 +     pyirkbot // I think you freaky! - YouTube
17:07 >>        join // bsdaddict: ~bsdaddict@nl5x.mullvad.net
17:10 +        ZEROF // Hi all ;)
17:13           SreK // Hi
17:13 <<        quit // eli: (Quit: Web client closed)
17:14 >>        join // eli: ~webchat@65.88.88.202
17:14 @   blinkenbot // eli is currently in the signup process (step 4 / 5), please help
                        him/her get an account!
17:18            eli // congrats to cds who got vouced.
17:18            eli // can't wait :)   ouch
17:20 <<        quit // mal: (Quit: schaars!)
17:20 >>        join // mal: mal@triton.blinkenshell.local
17:21 !!        mode // +v mal
17:22 +        mikk0 // hmm cant figure out how to do port forwarding on win7 really. it
                        only accepts local connections
17:31 <<        quit // eli: (Quit: Web client closed)
17:31 >>        join // eli: ~webchat@65.88.88.202
17:31 @   blinkenbot // eli is currently in the signup process (step 4 / 5), please help
                        him/her get an account!
#blinkenshell > 
Screen(s): 0* zsh                                              5:32PM  Sat, Nov/02/2013
```

```
              +
              #
             ###
            #####
            ######
           ; #####;
          +##.#####
         +##########
        #############;
       ###############+
      #######   #######
    .######;     ;###;`".
   .#######;     ;#####.
   #########.   .########`
  ######'           '######
 ;####                 ####;
 ##'                     '##
#'                         `#

User: unek
Hostname: starszypierdolnik
Distro: Arch Linux
Kernel: 3.11.6-1-ARCH
Uptime: 7:40
Window Manager: herbstluftwm
Shell: /bin/zsh
Terminal: xterm-256color
Packages: 1115
CPU: AMD Phenom(tm) II X4 965 Processor
RAM: 2650 MB / 7964 MB

~ [                                                            ] 5:25 PM
```

```
3:01/3:33                  ─┤ Never Gonna Give You Up ├─                   Vol: 30%
[playing]           Rick Astley - Whenever You Need Somebody (1987)        [r-----]
────────────────────────────────────────────────────────────────────────────────────
Time     Artist                    Album                    Title
────────────────────────────────────────────────────────────────────────────────────
3:33     Rick Astley               Whenever You Need Somebody Never Gonna Give You Up
3:33     Rick Astley               Whenever You Need Somebody Never Gonna Give You Up
3:33     Rick Astley               Whenever You Need Somebody Never Gonna Give You Up
3:33     Rick Astley               Whenever You Need Somebody Never Gonna Give You Up
3:33     Rick Astley               Whenever You Need Somebody Never Gonna Give You Up
3:33     Rick Astley               Whenever You Need Somebody Never Gonna Give You Up
3:33     Rick Astley               Whenever You Need Somebody Never Gonna Give You Up
3:33     Rick Astley               Whenever You Need Somebody Never Gonna Give You Up
3:33     Rick Astley               Whenever You Need Somebody Never Gonna Give You Up
3:33     Rick Astley               Whenever You Need Somebody Never Gonna Give You Up
3:33     Rick Astley               Whenever You Need Somebody Never Gonna Give You Up
3:33     Rick Astley               Whenever You Need Somebody Never Gonna Give You Up
3:33     Rick Astley               Whenever You Need Somebody Never Gonna Give You Up
3:33     Rick Astley               Whenever You Need Somebody Never Gonna Give You Up
3:33     Rick Astley               Whenever You Need Somebody Never Gonna Give You Up
3:33     Rick Astley               Whenever You Need Somebody Never Gonna Give You Up
3:33     Rick Astley               Whenever You Need Somebody Never Gonna Give You Up
```

```
~ [                                                            ] 5:32 PM
```

```
1  [||||||                  14.7%]   2  [|||||                   12.5%]
3  [||||||                  13.3%]   4  [||||||||                18.4%]
Avg[|||||||                 14.7%]   Mem[|||||||||||||||||||| 2195/7964MB]
Load average: 0.88 0.94 0.84         Uptime: 07:47:04
Tasks: 95, 116 thr; 1 running        Hostname: starszypierdolnik
```

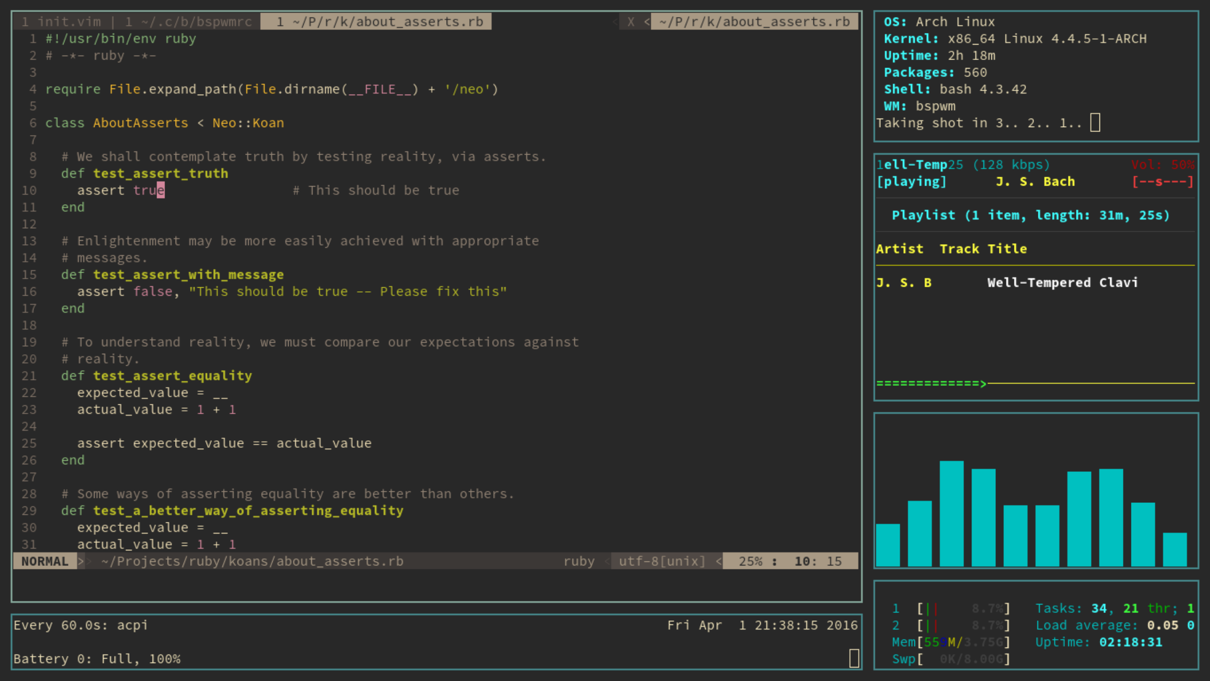


```
1                                   73C   70   40   75   176ms   08 Nov 2015   14:25
```

```
        context_1: Context::new(),
        context_2: Context::new(),
        short_term_mem: Memory::new(n2),
        long_term_mem_tree: Branch::new(),
    }
}

// Open questions
//
// How can I save things to long term memory?
//
pub fn process(&mut self, phrase: Option<Vec<Info>>) {
    match phrase {
        // Processes a possible data
        Some(phrase) => {
            // Checks each pattern from semantic.
            // Known words are Signal::Known(pattern, info);
            // Unknown ones are Signal::Unknown(info);
            let mut words = phrase.into_iter().map(|i| {
                let info = Rc::new(i);
                self.encoded_info
                    .get(&info)
                    .and_then(|p| Some(Signal::Known(p, info.clone())))
                    .or_else(|| Some(Signal::Unknown(info.clone())))
                    .unwrap()
            }).collect::<Vec<Signal<_>>>();

            // Creates a random's engine from 0 to 63.
            let mut rng = rand::thread_rng();
            let mut geometric_pattern = || {
                Pattern::with_seq((1 << 63) >> Range::new(0u8, 63).ind_sample(&mut rng))
            };

            let nc_gp = geometric_pattern();

            // Gets joint context from words.
            let new_context = words.iter().fold(nc_gp, |sum, s| {
                match *s {
                    Signal::Known(p, _) => {
                        self.push_stm(p);
                        sum.join(&p).join(&nc_gp)
                                                                    217,30        61%
```

```
c0re  connoor        jaapz  jeanjack  m_b  OlivierW  sardonyx        xs

xs   don't try with me, you'll waste your time :p
     no I don't want to teach monads
xs   hahaha
xs   There's a techer specialized in Haskel in a private IT school
xs   in france
xs   which I didn't like so much, but friends of mine are in this school
     he's probably a math guy
xs   eyah
xs   yeah *
xs   clearly
xs   http://www.epitech.eu/ just their website is full of crappy javascript and all kind of
     trackers & analytics tools and of course
xs   no CDN
xs   this is a joke
     needs more node.js
xs   :(
     it's better than brazilian ones
     seems that no one here gives a shit about good practices
xs   wait, they are forcing students to use emacs
xs   :/
     outch
```

```
1:37/3:38                        Outatime                                Vol: 70%
[playing]              FM-84 - Los Angeles EP (2015)                     [------]

LGHTNNG - Sharks_ Run.mp3
Light Affair.mp3
Los Angeles - Delorean.mp3
Los Angeles - Outatime.mp3                                          Los Angeles EP
Lost in Your Eyes (feat. Per Rinaldo).mp3
Lucy in Disguise - 1987.mp3
M.O.O.N. - 'Crystals' [Hotline Miami Soundtrack].mp3
M.O.O.N. - 'Hydrogen' [Hotline Miami Soundtrack].mp3
Man and Machine - Night Vision.mp3                                  Man and Machine
Miami Nights 1984 - Ocean Drive.mp3
Mooginizer - Interstellar Travel.mp3
```



```
1 init.vim | 1 ~/.c/b/bspwmrc   1 ~/P/r/k/about_asserts.rb          X < ~/P/r/k/about_asserts.rb
#!/usr/bin/env ruby
# -*- ruby -*-

require File.expand_path(File.dirname(__FILE__) + '/neo')

class AboutAsserts < Neo::Koan

  # We shall contemplate truth by testing reality, via asserts.
  def test_assert_truth
    assert true                # This should be true
  end

  # Enlightenment may be more easily achieved with appropriate
  # messages.
  def test_assert_with_message
    assert false, "This should be true -- Please fix this"
  end

  # To understand reality, we must compare our expectations against
  # reality.
  def test_assert_equality
    expected_value = __
    actual_value = 1 + 1

    assert expected_value == actual_value
  end

  # Some ways of asserting equality are better than others.
  def test_a_better_way_of_asserting_equality
    expected_value = __
    actual_value = 1 + 1
NORMAL > ~/Projects/ruby/koans/about_asserts.rb          ruby < utf-8[unix] < 25% :  10: 15
```

```
OS: Arch Linux
Kernel: x86_64 Linux 4.4.5-1-ARCH
Uptime: 2h 18m
Packages: 560
Shell: bash 4.3.42
WM: bspwm
Taking shot in 3.. 2.. 1..
```

```
1ell-Temp25 (128 kbps)                Vol: 50%
[playing]      J. S. Bach            [--s---]

  Playlist (1 item, length: 31m, 25s)

Artist  Track Title

J. S. B       Well-Tempered Clavi

=============>
```

```
Every 60.0s: acpi                                   Fri Apr  1 21:38:15 2016

Battery 0: Full, 100%
```

```
 1  [||        8.7%]   Tasks: 34, 21 thr; 1
 2  [||        8.7%]   Load average: 0.05 0
 Mem[559M/3.75G]       Uptime: 02:18:31
 Swp[   0K/8.00G]
```

What now?

- Q&A round
  in a few seconds
- Join the next "Stammtisch":
  18:00 Thu, April 7 - learning spaces above the Green Floor, ETH CHN

Course material

- These slides
- Configuration files I've used for the demo
- Can all be found online:
  http://thealternative.ch/index.php?view=knowhow

- License: CC BY-SA 3.0
- Theme by Christian Horea, CC BY