Instructions for installation

1. Install the required dependencies for dwm. I can't verify it on my system but according to a user on GitHub this should do it for Fedora so it should be fine for anything Red Hat based?

   ```
   localhost# dnf install libX11-devel libXft-devel libXinerama-devel
   libXrandr-devel xorg-x11-xinit-session
   ```

   Of course it also is possible to just install dwm from the repositories because then all the dependencies should be installed and after doing that you could simply replace the installed directory with the one which I give you and run sudo make install again.

2. Copy the dwm-directory to any path of your system and go into it.
   sudo make install (no clean install needed because it is not a clean install considering that you copy it from me)
   All settings are chosen in the config.h file so modify that before compiling if you want to make any changes. Do not make the changes in the config.def.h file, it won't be applied while compiling dwm.

3. Take from my .xinitrc what you like and put that in your own .xinitrc or in your .Xprofile.

4. If you want a bit more functionality in the panel then copy my scripts for xsetroot, I called it panelfilling, this script is started from the .xinitrc or .Xprofile. I will give the laptop-variant for which I added the battery-charge.
   https://dwm.suckless.org/status_monitor/
   Here you can find all the programs which are listed on the Suckless-website to make a fancy-panel, dwm-blocks is the big hype, one of its claims to fame is that it allows to set different timers for different elements, it also has patches for additional features.

   https://github.com/torrinfail/dwmblocks
   It seems to me that this is an excellent starting point to pimp the statusbar if you'd like that, easy to copy as long as you don't have any issue with pasting the icons.

   Also take a look at some of the patches like this one:
   https://dwm.suckless.org/patches/alternativetags/

5. When you make any change in the config.h or you patch dwm then you always have to recompile it first (*sudo make install)* and only after quitting dwm and restarting dwm the changes will take effect. When compiling fails then you keep the old binary and dwm will remain on your system as it was.

Most important hotkeys

You can read all the hotkeys in the config.h file but here I will list the ones which are essential to get started.

windows+enter	open the chosen terminal
windows+numpad-number	select workspace
windows+shift+numpad-number	move focused window to selected workspace
windows+control+numpad-number	add the selected workspace to the currently active workspace. All the windows on the added workspaces will be shown
ctrl+shift+d	open d-menu, by default it appears on the top of the monitor and unpatched it simply searches for programs in your PATH-directories.
windows+j/k	change focus to the next/previous window
windows+shift+j/k	move all the programs to the next/previouw window
windows+shift+u/i	switch two programs in between two windows, for when you want more control over which program is where without closing and re-opening programs
windows+b	toggle panel
windows+tab	move to the previously selected workspace
windows+h/l	change the ratio between master and stack
windows+shift+h/l	change the ratio between two windows in the stack
windows(+shift)+,/.	to switch between monitor and send programs to another monitor (each monitor has independent workspaces)
windows+t	tiling mode
windows+f	'fullscreen' mode
windows+m	one window visible (not the same as fullscreen)
windows+shift+c	kill application in focused window and remove window
windows+shift+q	kill dwm
windows+d/i	decrease or increase the number of windows for the master-area
windows+t	master/stack layout
windows+m	1 window gets the full screen
windows+lmb/rmb	drag/resize window

Patching dwm

When you install dwm from source you get a rather minimal version without some of the desired features, you have to patch those in. This can be done manually or via a tool like flexipatch.
If you want to search for patches then I recommend to look here first instead of the atrocious (lot of mouse-clicking and scrolling required) website of suckless:
https://gist.github.com/nic0-lab/c435dd4e77b2b3d274d142b126899525

On this install of dwm the following patches were applied: rotatestack, attachaside, pertag, systray, movestack, resizecorner, swallow, cfacts

Manual patching

If you want to use the integrated systemtray of dwm then I highly recommend that you do that patch automatically via the patch program which most likely already is on the user his system. Out of my head this should work: *patch < diff-file*
For the other patches you could try your luck with the patch program but because the patch has to be applied on different code than the source file it gets more difficult to automatically patch over time and you are likely to have to do some debugging. Also often you have to use patches which of which the diff-files were generated for older versions of dwm.

Alternatively you can just open the diff-file in your favorite text-editor (syntax-highlighting highly recommended!). The structure of a diffile is pretty self-explanatory, it tells you with a header which files the changes are shown for, it gives you thenumber (unpatched version of dwm!) and some surrounding code, if there is a - in front of a line then you remove that line, if there is a + in front of a line then you add the line. It helps if you have basic knowledge of C-programming (structure of a file, functions and scope) but it definitely is not required. Most patches shouldn't take more than 5-10 minutes per patch if you do it this way and rarely a patch does not work if it was not updated to the newer versions of dwm, but sometimes you need to be a bit creative due to little changes in the code at the spots where you patch it. To remove a patch you simply do it the reversed way: add lines with a - in front of it and remove lines with a + in front of it.
Always backup the directory first before patching manually.

Flexipatch

There also is a lazier way to patch dwm, most patches are an option if you use this. You simply select the patches which you want to use in a config-file by selecting those (0 => 1 in the preprocessor directives) and then you compile it (*sudo make clean install)* and run another script to clean up all the unused code for the unused patches.
https://github.com/bakkeby/dwm-flexipatch

What makes dwm unique as a tiling window manager?

It comes with an integrated panel which makes it easy to hide/unhide it in a clean (windows get redrawn, the panel is not put over the top of the windows) and bugfree way, dwm comes with an integrated and well-working (for example not the case with Sway) systemtray if you apply that

patch, dwm is incredibly fast (roughly 300 kB of C-code, with patches) and dwm has as they call it the tag-feature: dwm uses an xor-bitwise operation to select tags, for example 100100100 to select the three tags which have a value of 1. This way you can easily select multiple workspaces to be shown at the same time, you can open a program on multiple workspaces and you can temporarily show the windows of a workspace on the selected workspace and see that program on both workspaces if you select any of those two. Why is this useful? Imagine that you have a PDF-reader, browser and text-document open, you don't want a quarter monitor for two of those programs all the time so you can quickly switch between any combination of 2 or all 3 at the same time depending on what you need at that particular moment. The modularity of dwm could be considered both an advantage (faster, no code which you don't need) and disadvantage (extra work to set it up).