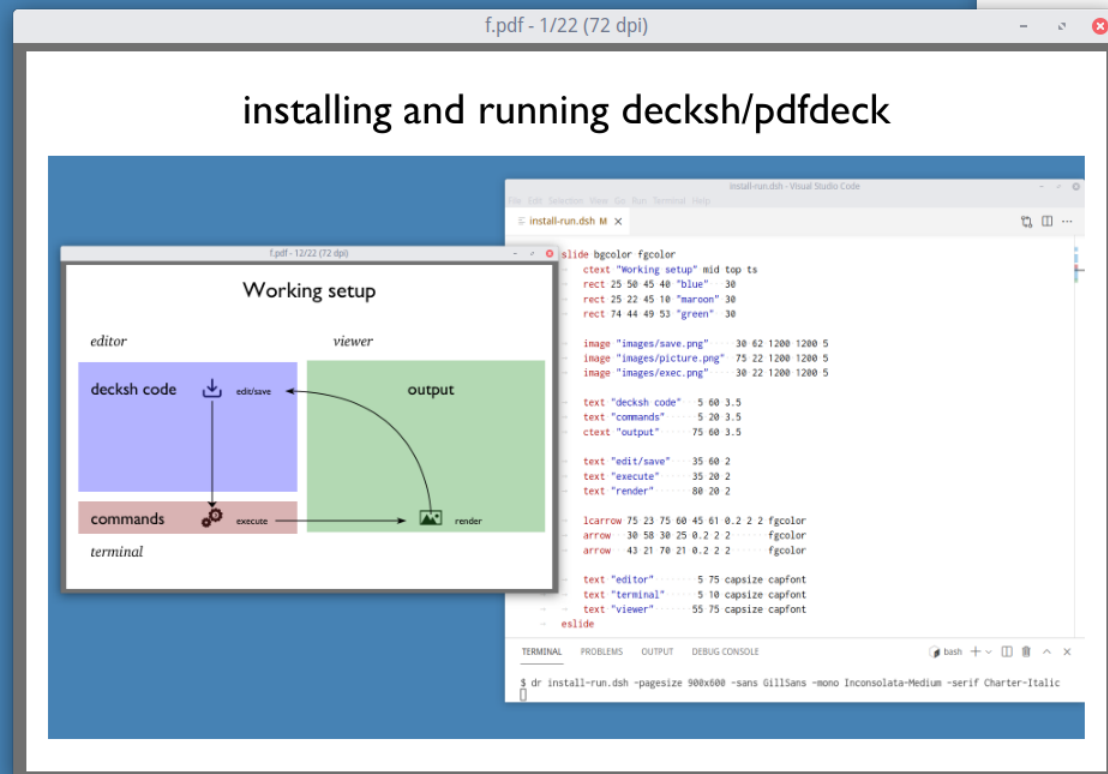


Installing and Running decksh/pdfdeck



```
install-run.dsh - Visual Studio Code
File Edit Selection View Go Run Terminal Help
install-run.dsh M X

→ slide bgcolor fgcolor
  ctext "Working setup" mid top ts
  rect 25 50 45 40 "blue" 30
  rect 25 22 45 10 "maroon" 30
  rect 74 44 49 53 "green" 30

  image "images/save.png" 30 62 1200 1200 5
  image "images/picture.png" 75 22 1200 1200 5
  image "images/exec.png" 30 22 1200 1200 5

  text "decksh code" 5 60 3.5
  text "commands" 5 20 3.5
  ctext "output" 75 60 3.5

  text "edit/save" 35 60 2
  text "execute" 35 20 2
  text "render" 80 20 2

  larrow 75 23 75 60 45 61 0.2 2 2 fgcolor
  arrow 30 58 30 25 0.2 2 2 fgcolor
  arrow 43 21 70 21 0.2 2 2 fgcolor

  text "editor" 5 75 capsize capfont
  text "terminal" 5 10 capsize capfont
  text "viewer" 55 75 capsize capfont
→ eslide

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE
$ dr install-run.dsh -pagesize 900x600 -sans GillSans -mono Inconsolata-Medium -serif Charter-Italic
```

Installing

Installing using go and git

```
$ go install github.com/ajstarks/decksh/cmd/decksh@latest
```

Install the latest version of decksh

```
$ go install github.com/ajstarks/deck/cmd/pdfdeck@latest
```

Install the latest version of pdfdeck

```
$ git clone https://github.com/ajstarks/deckfonts $HOME/deckfonts
```

Install fonts into \$HOME/deckfonts


```
$ decksh -help
```

```
$ pdfdeck -help
```

Do a test run of decksh and pdfdeck

Installing decksh and pdfdeck binaries

<https://github.com/ajstarks/decksh/tree/master/cmd/decksh/binaries>

| | |
|--|------------|
|  darwin-amd64-decksh | 1 hour ago |
|  darwin-arm64-decksh | 1 hour ago |
|  linux-amd64-decksh | 1 hour ago |
|  windows-386-decksh.exe | 1 hour ago |
|  windows-amd64-decksh.exe | 1 hour ago |

Pick your type, download, rename to 'decksh', place where your apps live.



<https://github.com/ajstarks/deck/tree/master/cmd/pdfdeck/binaries>

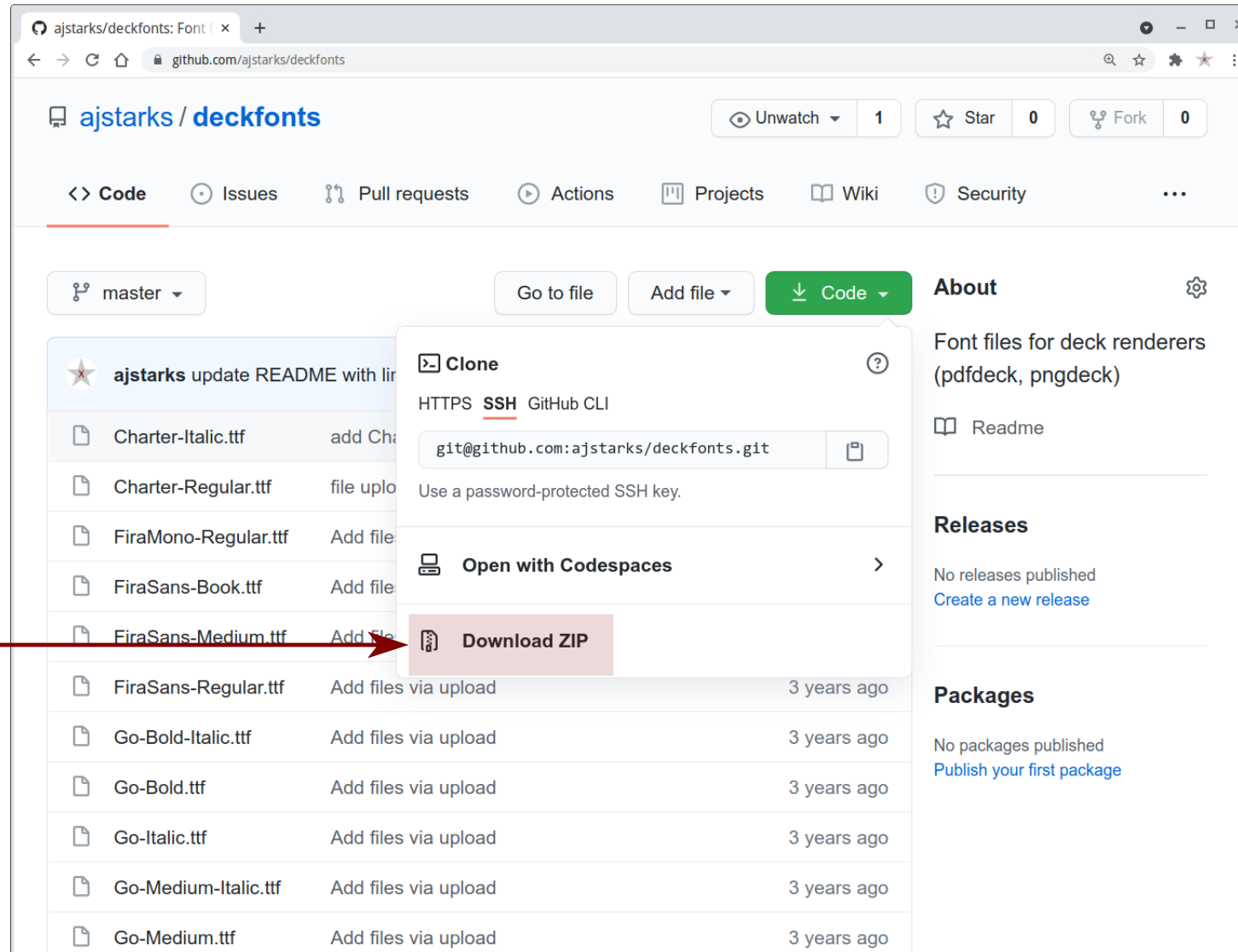
| | |
|---|------------|
|  darwin-amd64-pdfdeck | 1 hour ago |
|  darwin-arm64-pdfdeck | 1 hour ago |
|  linux-amd64-pdfdeck | 1 hour ago |
|  windows-386-pdfdeck.exe | 1 hour ago |
|  windows-amd64-pdfdeck.exe | 1 hour ago |

Pick your type, download, rename to 'pdfdeck', place where your apps live.



Downloading the fonts

Download, unzip
to your home
directory.



Fonts Catalog

Default Fonts

Times times timesi timesi timesb timesbi

Hamburgevons 0123456789

Helvetica helvetica helveticai helveticab helveticabi

Hamburgevons 0123456789

Courier courier courieri courierb courieri

Hamburgevons 0123456789

Alternative serif, sans, mono

Charter Charter-Regular Charter-Italic

Hamburgevons 0123456789

Fira Sans FiraSans-Book FiraSans-Medium FiraSans-Regular

Hamburgevons 0123456789

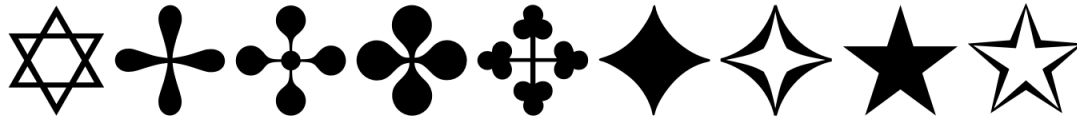
Inconsolata Inconsolata-Regular Inconsolata-Bold Inconsolata-Medium Inconsolata-Condensed

Hamburgevons 0123456789

Symbol fonts

Zapf Dingbats

zapfdingbats



Gophers

gophers



State Face

stateface

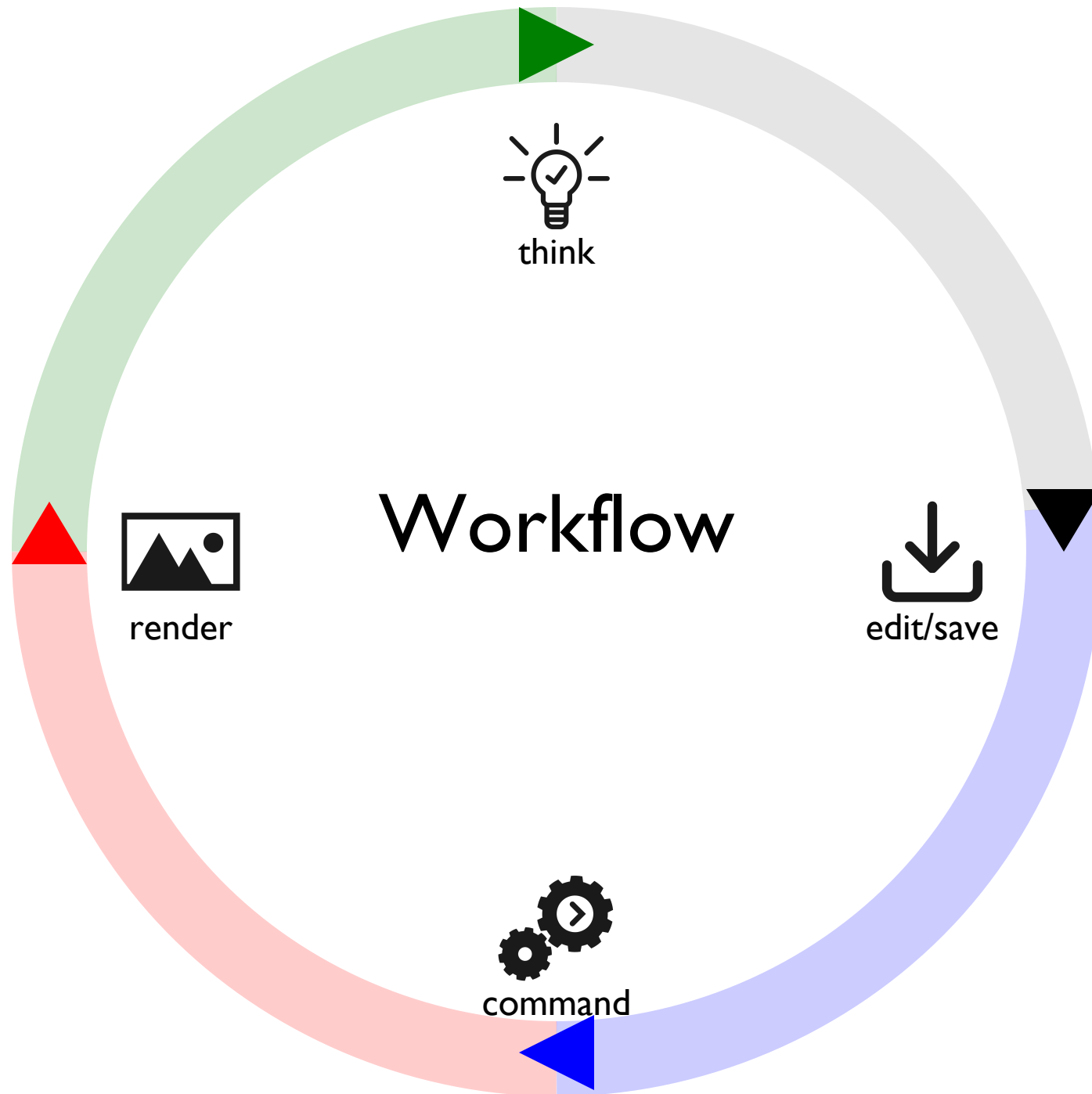


Wee People

weepeople



Running



Working setup

editor

viewer

decksh code



edit/save

output

commands

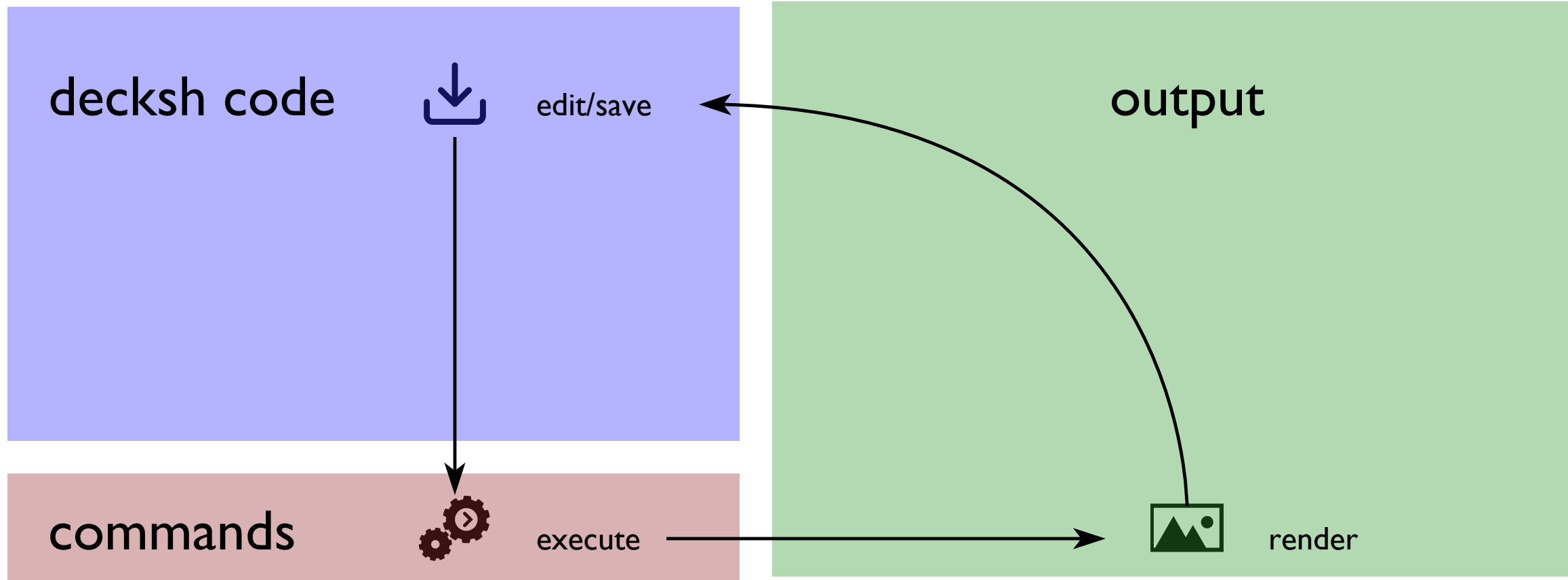


execute



render

terminal

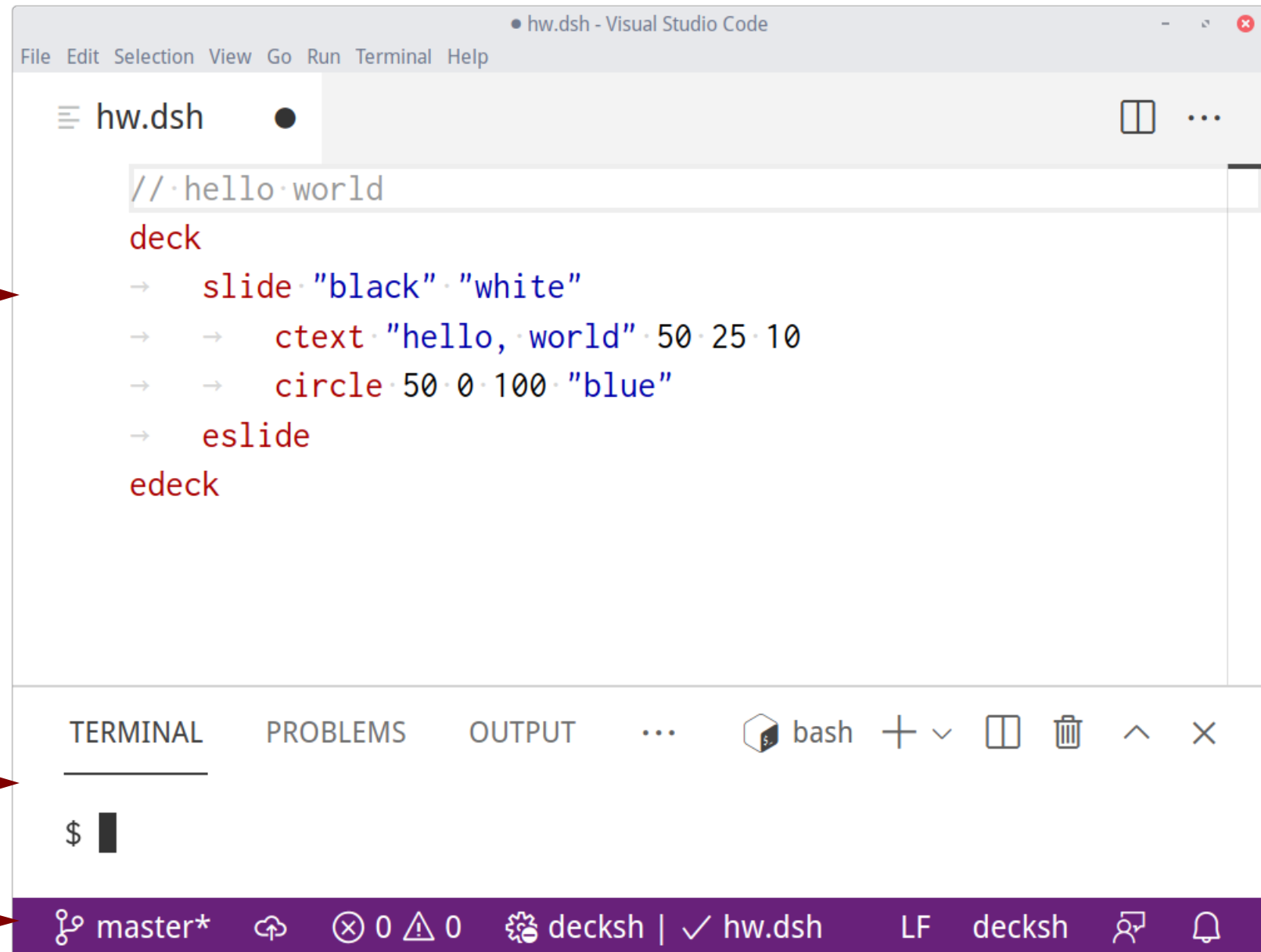


VSCode: Editor for Mac, Windows, and Linux

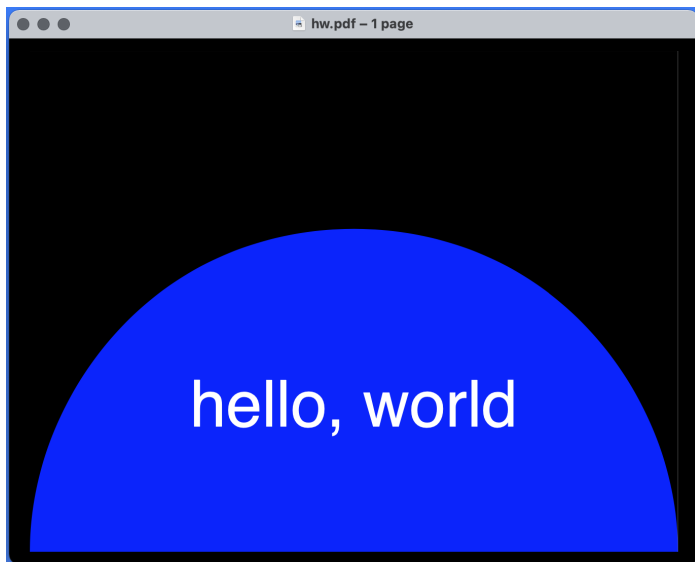
decksh code with
syntax highlighting

Integrated terminal

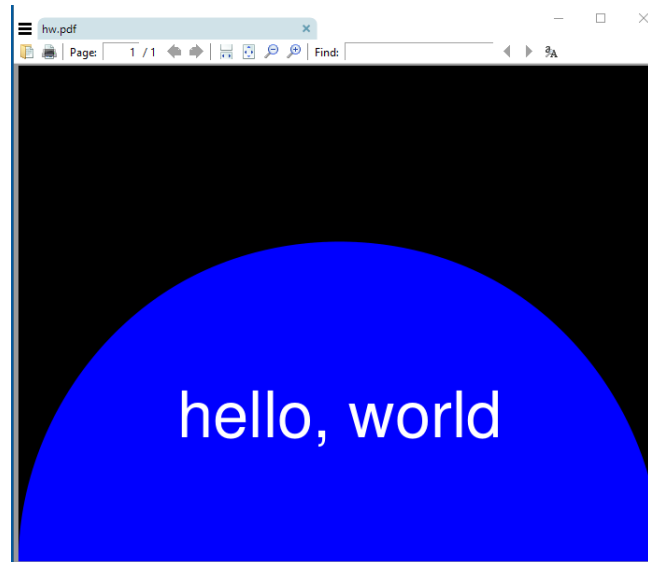
Status showing decksh mode



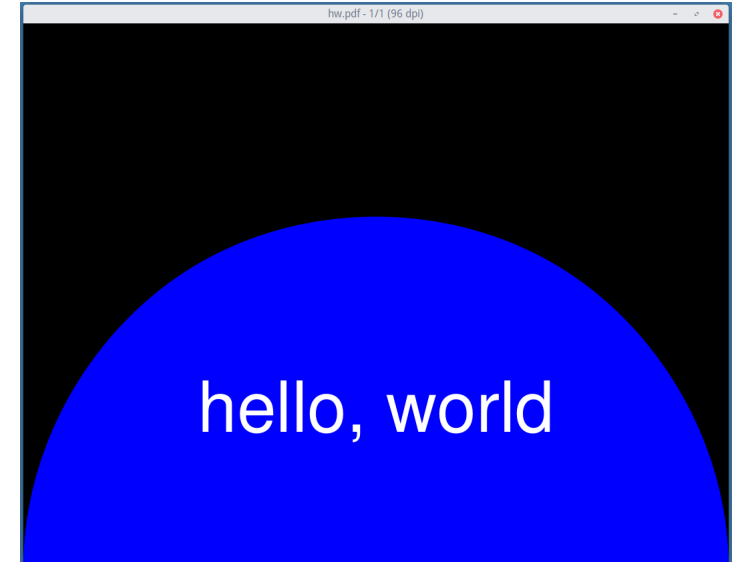
PDF Readers



Mac: Preview



Windows: Sumatra PDF

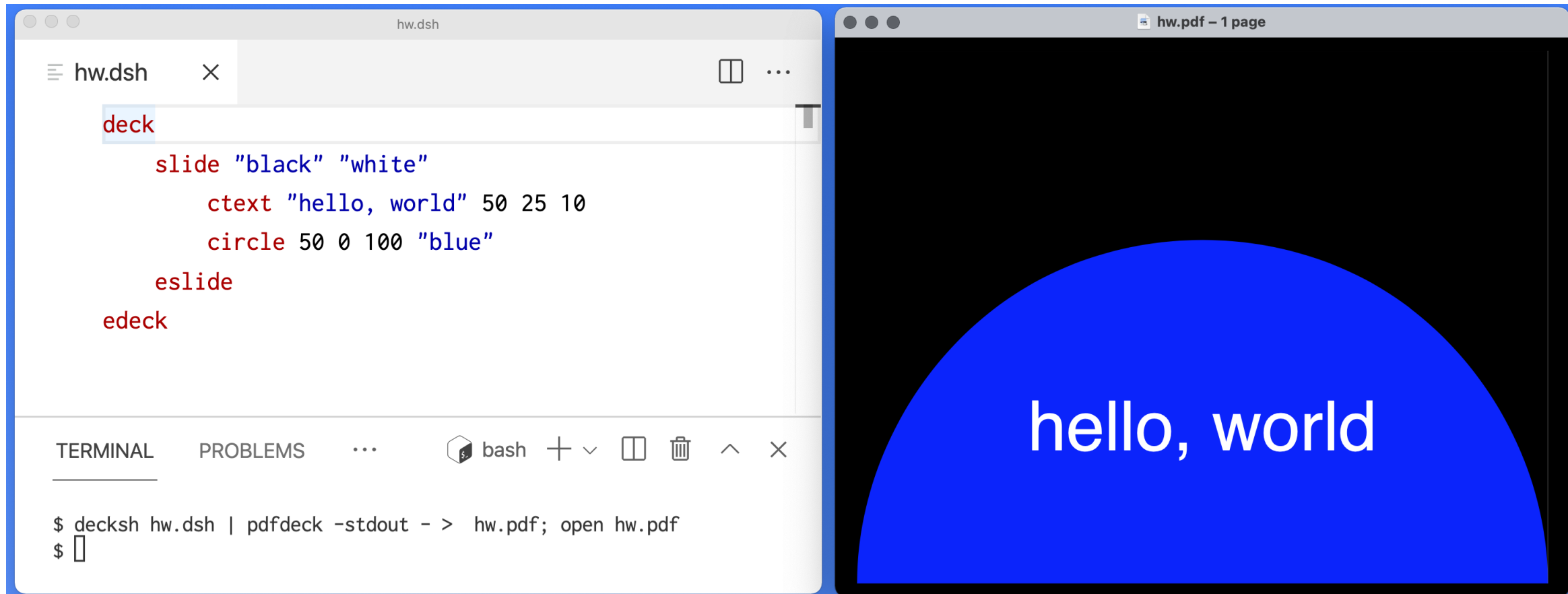


Linux: mupdf

Mac OS

editor: VSCode

viewer: Preview

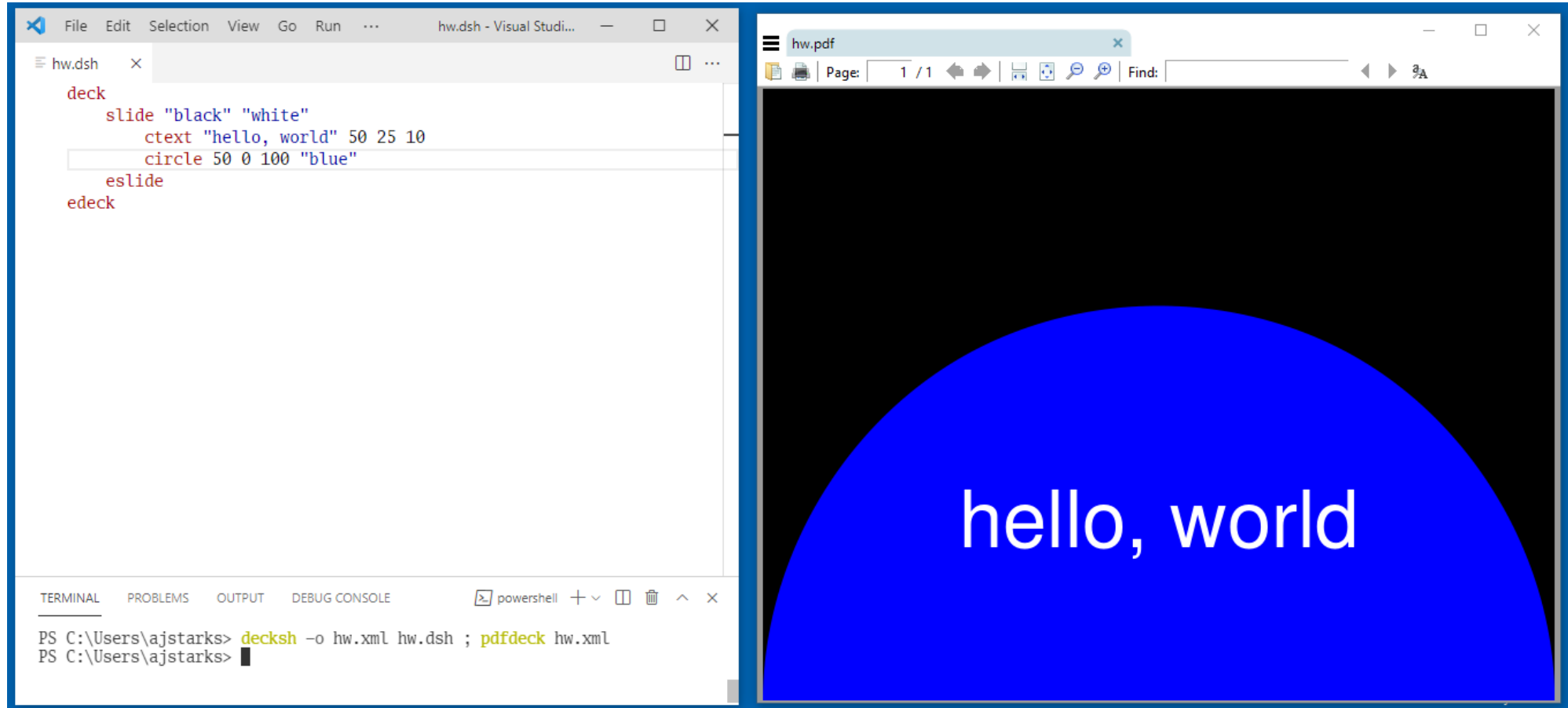


terminal: VSCode

Windows

editor: VSCode

viewer: Sumatra PDF

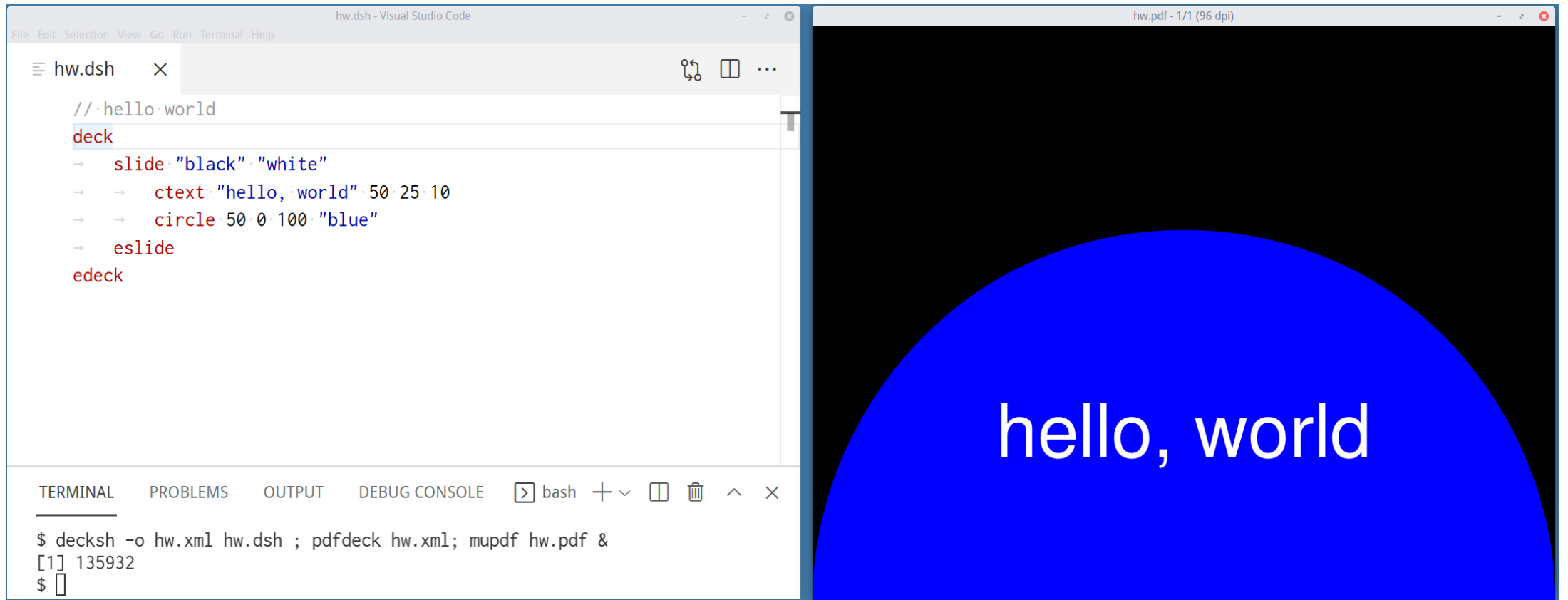


terminal: VSCode

Linux

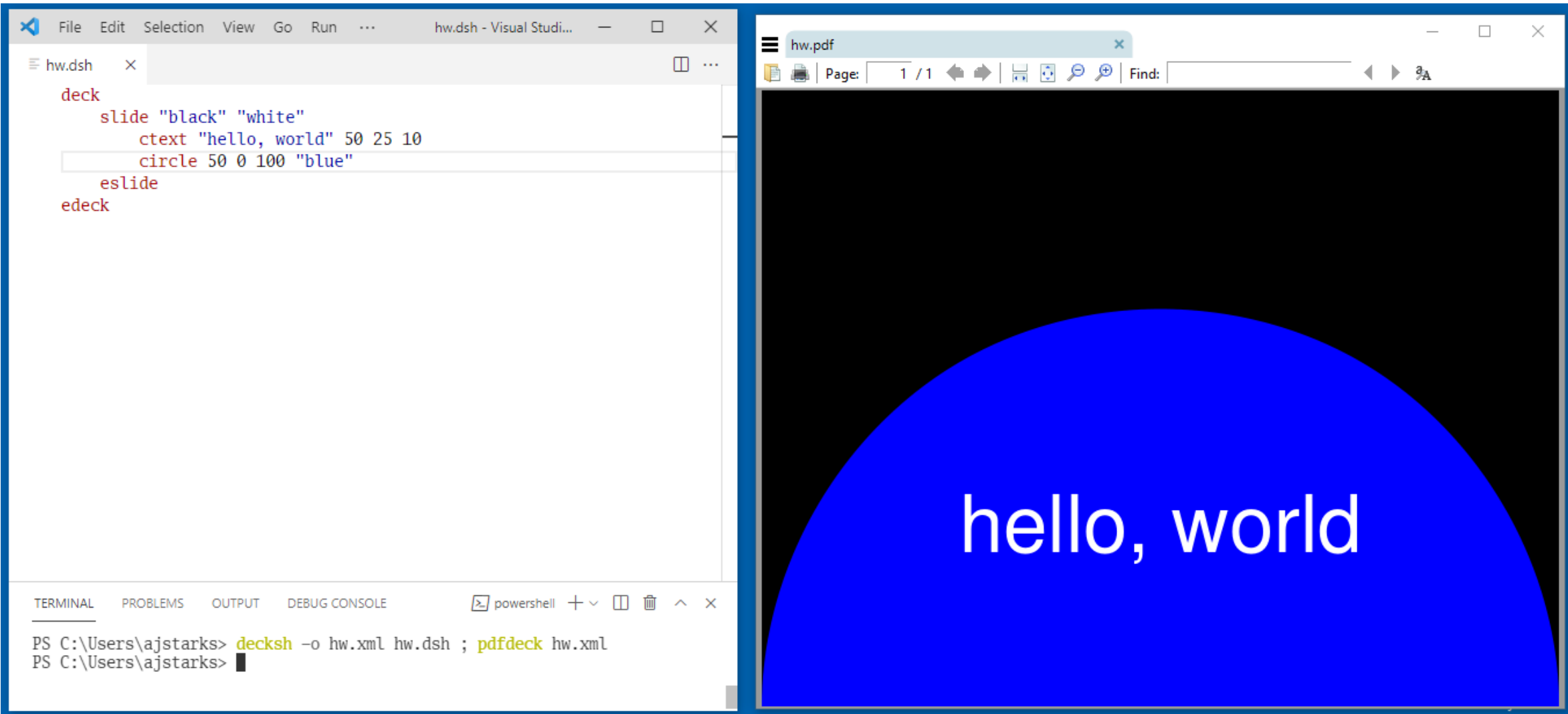
editor: VSCode

viewer: mupdf

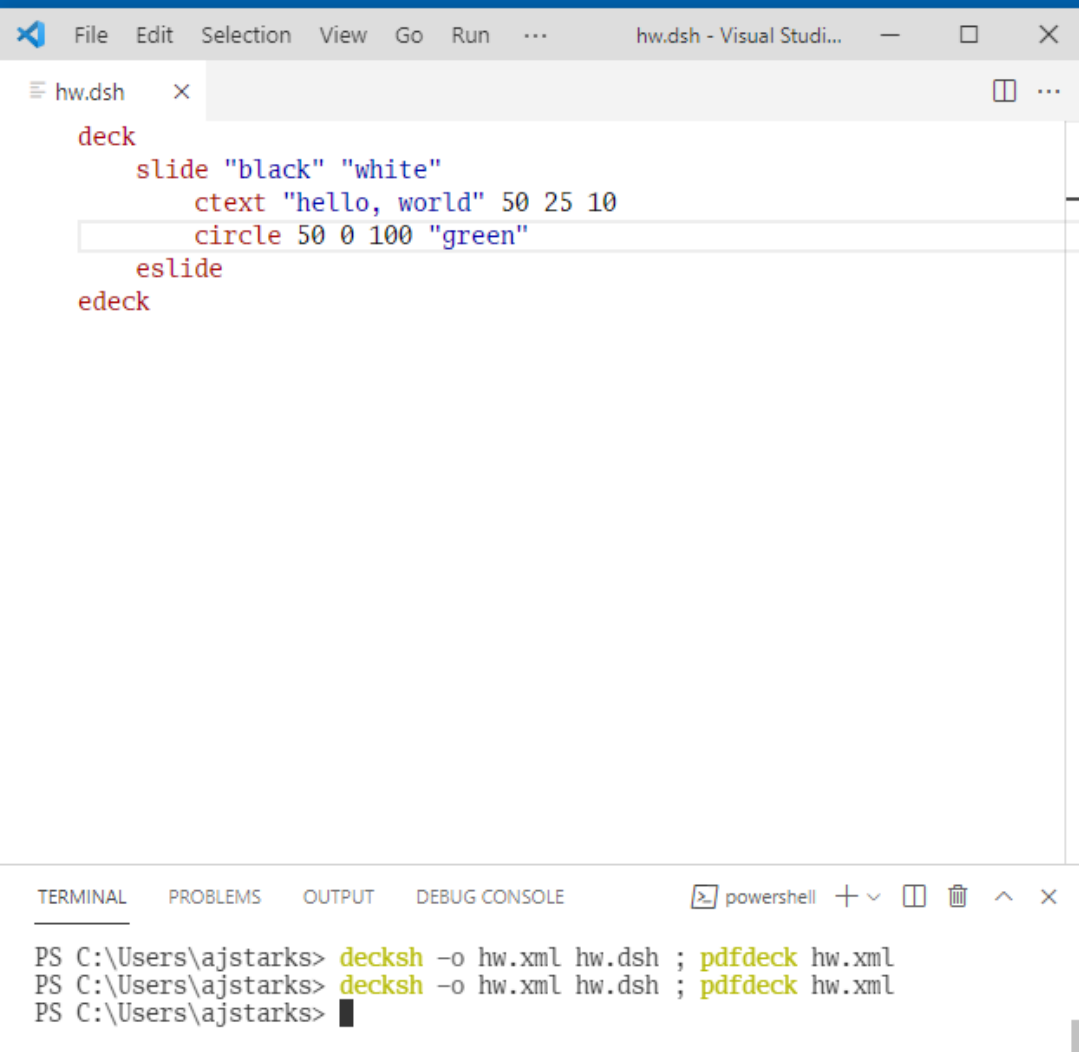


terminal: VSCode

Render



Update

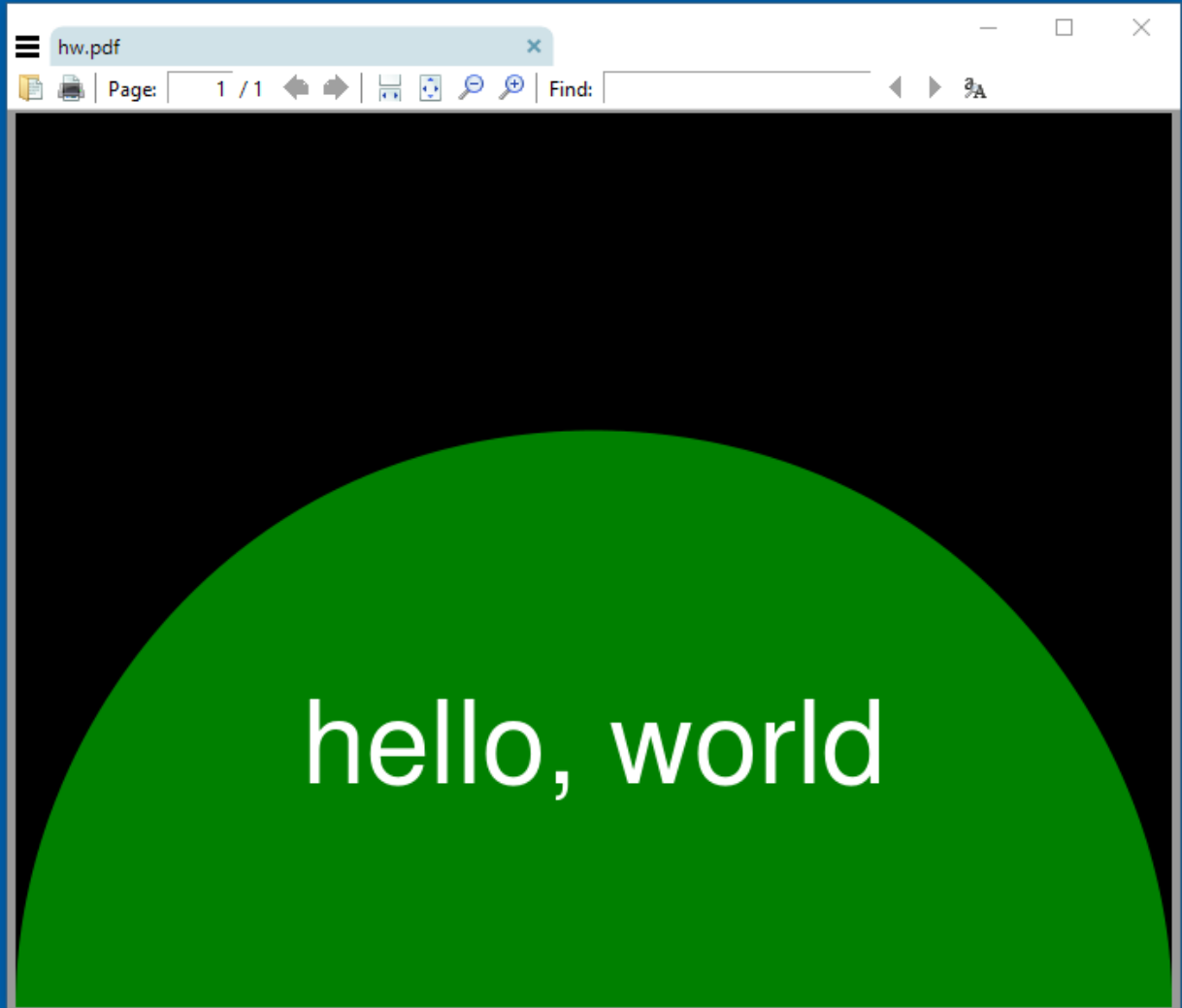


The screenshot shows the Visual Studio Code editor with a file named `hw.dsh` open. The file contains a sequence of commands for a presentation deck. Below the editor, the terminal window shows the execution of these commands using `decksh` and `pdfdeck`.

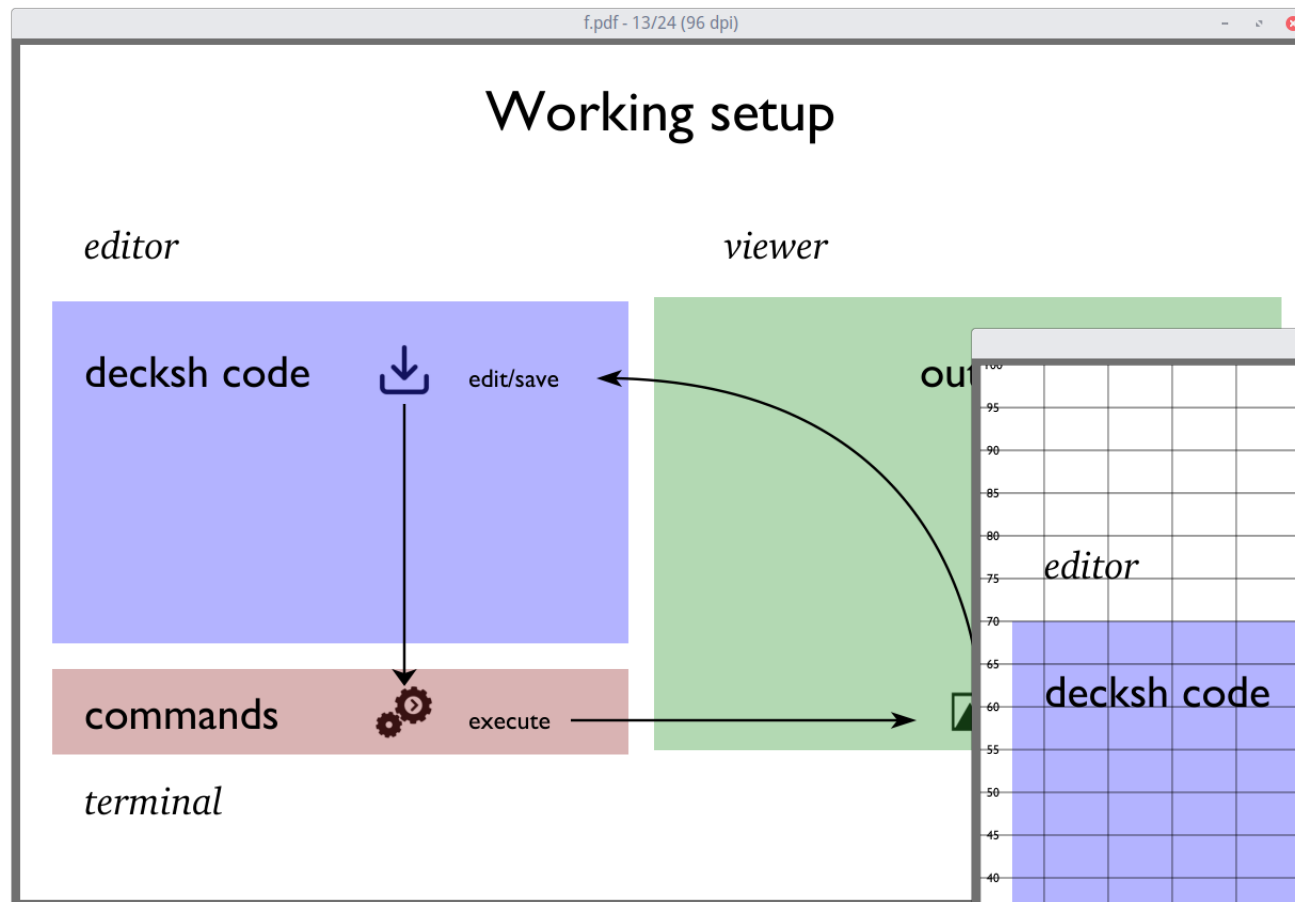
```
deck
  slide "black" "white"
    ctext "hello, world" 50 25 10
  circle 50 0 100 "green"
eslide
edeck
```

Terminal output:

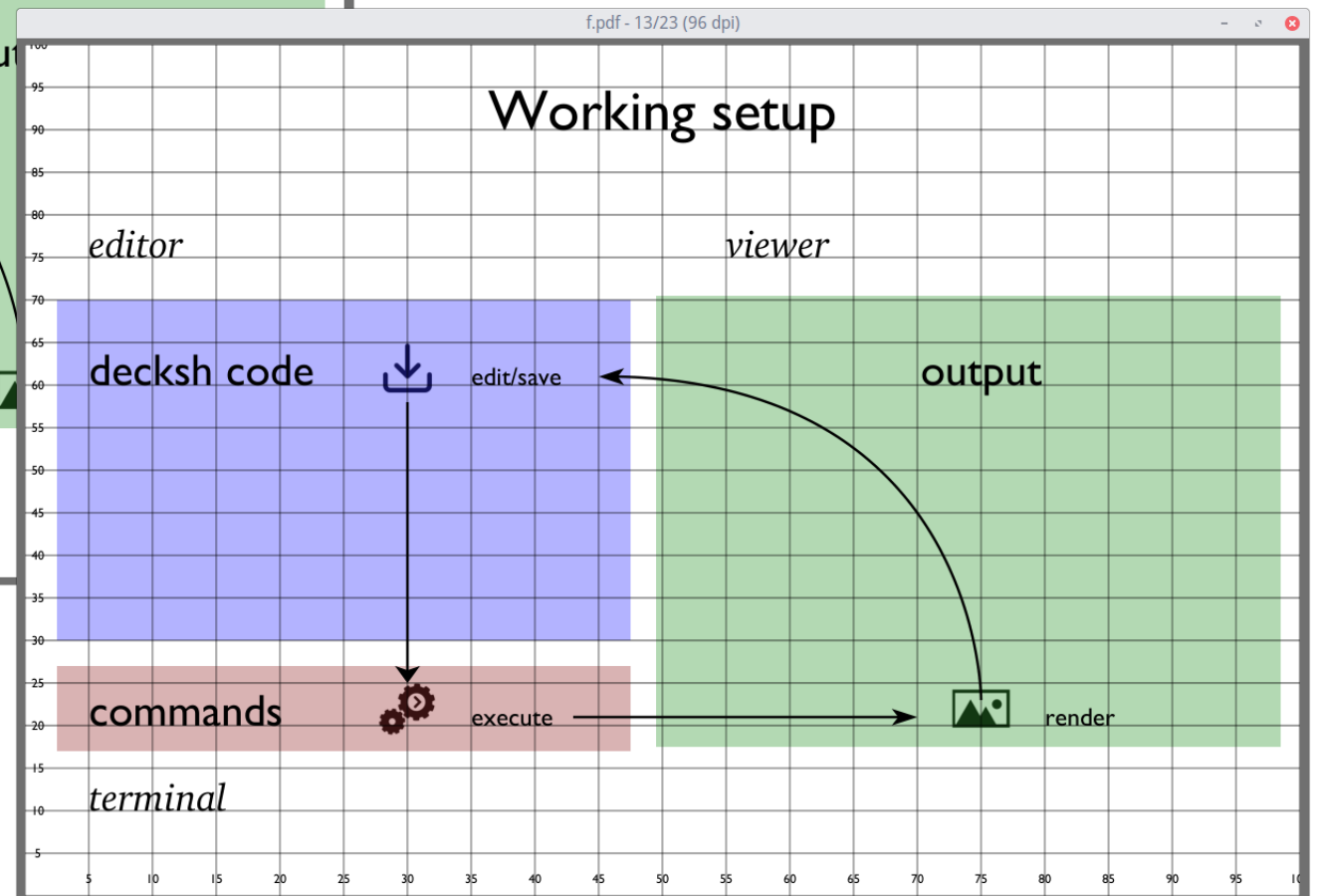
```
PS C:\Users\ajstarks> decksh -o hw.xml hw.dsh ; pdfdeck hw.xml
PS C:\Users\ajstarks> decksh -o hw.xml hw.dsh ; pdfdeck hw.xml
PS C:\Users\ajstarks>
```



Using the -grid option



no grid



-grid 5

VSCode setup

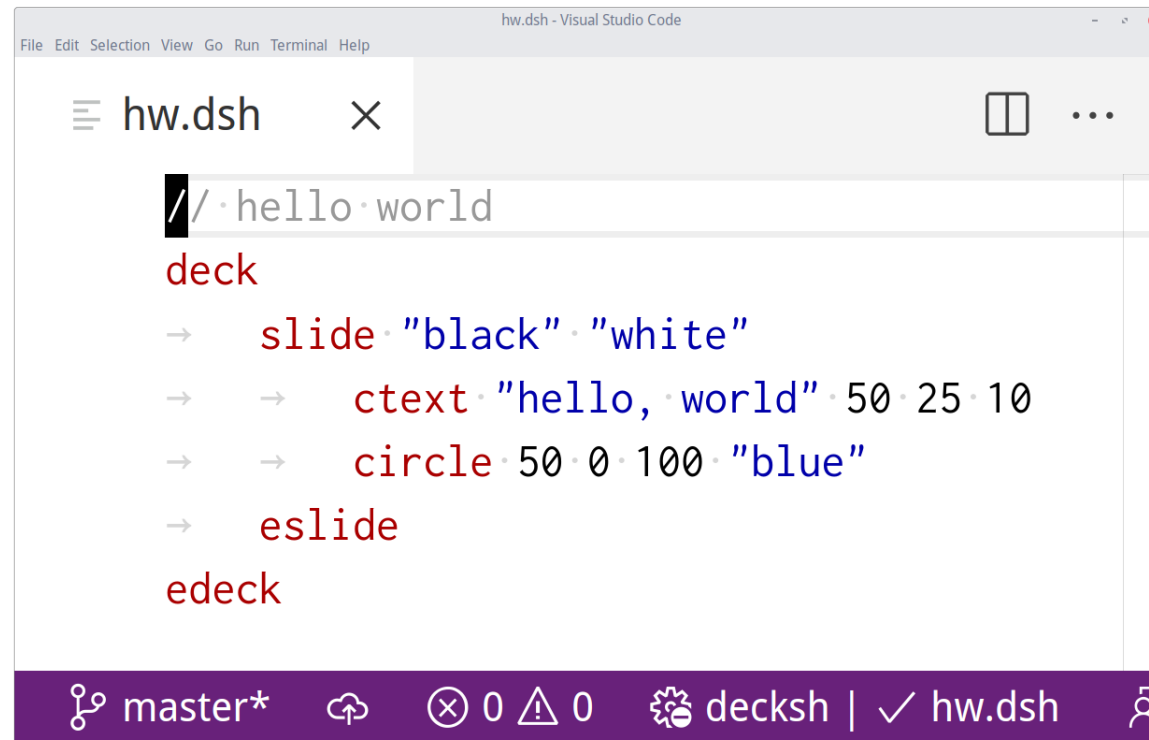
copy this to your settings

```
"editor.tokenColorCustomizations": {  
  "textMateRules": [  
    {  
      "scope": "keyword.other.command.decksh",  
      "settings": {  
        "foreground": "#AA0000"  
      }  
    },  
    ...  
  ]  
}
```

copy to .vscode/extensions/ajstarks.decksh-1.0.0

<https://github.com/ajstarks/decksh/tree/master/vscode>

| | | |
|-------------------------------|--|---------------|
| .. | | |
| 📁 syntax | add textbox as an alternative to textblock | yesterday |
| 📄 README.md | add documentation and testing files | 16 months ago |
| 📄 language-configuration.json | add documentation and testing files | 16 months ago |
| 📄 package.json | add documentation and testing files | 16 months ago |



vim setup

copy to .vim

<https://github.com/ajstarks/decksh/tree/master/vim>

| | | |
|----------|-----------------|---------------|
| .. | | |
| ftdetect | add vim support | 2 minutes ago |
| syntax | add vim support | 2 minutes ago |



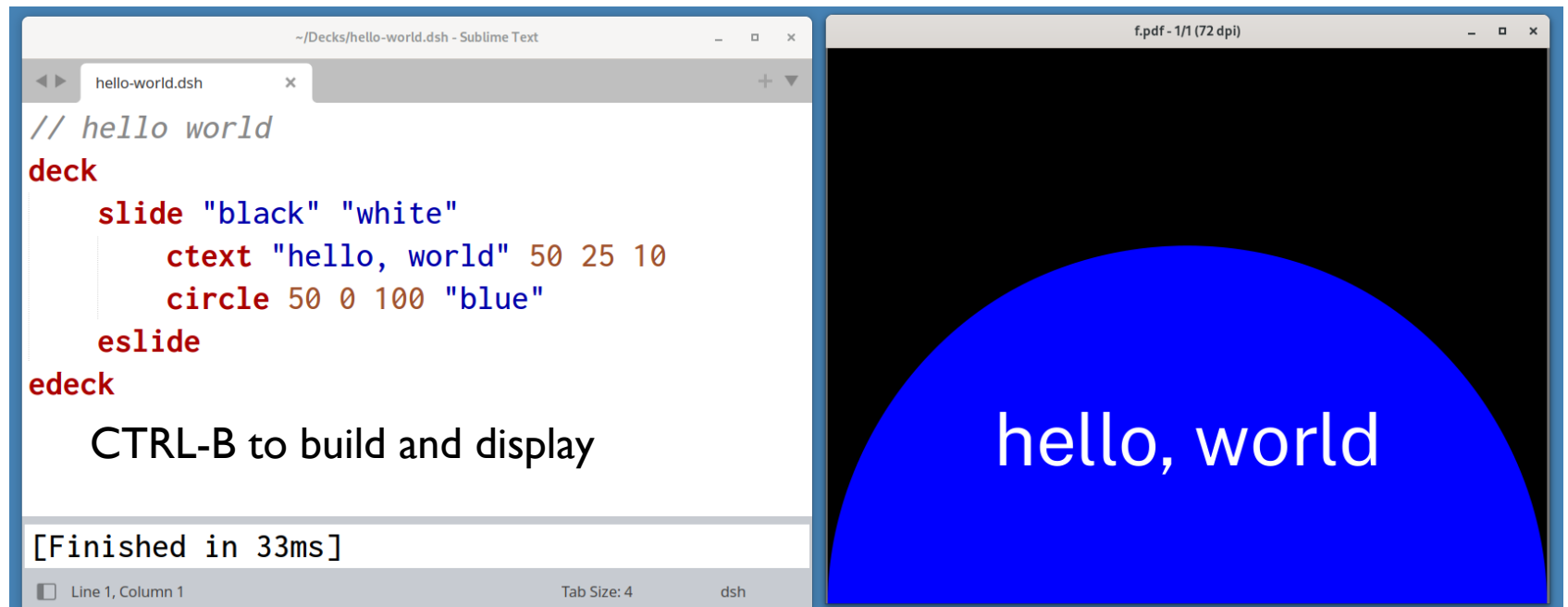
```
ajstarks@slab:~/gocode/decksh/doc — vim code/hw.dsh
// hello world
deck
    slide "black" "white"
        ctext "hello, world" 50 25 10
        circle 50 0 100 "blue"
    eslide
edeck
~
1,1 All
```

Sublime Text setup

copy to dsh.sublime-build
and dsh.sublime-syntax
to Sublime user
package directory,
copy dpdfr to
your PATH

<https://github.com/ajstarks/decksh/tree/master/sublime-text>

| Name | Last commit message | Last commit date |
|--------------------|----------------------------------|------------------|
| .. | | |
| dpdfr | update sublime text build script | 6 minutes ago |
| dsh.sublime-build | add sublime text build system | 6 months ago |
| dsh.sublime-syntax | add textblockfile | 4 months ago |



The command line

decksh command usage

| | |
|----------------------------|----------------------------------|
| decksh | read from stdin, write to stdout |
| decksh in.dsh | read from file, write to stdout |
| decksh -o out.xml | read from stdin, write to file |
| decksh -o out.xml in.dsh | read from file, write to file |
| decksh -version | show version |
| decksh -var name=value ... | assign a variable |
| decksh -dump ... | dump assignments |

decksh example.dsh | pdfdeck ...

pdfdeck [options] inputfile

| Option | Default | Description |
|-----------|--|---|
| -sans | helvetica | Sans Serif font |
| -serif | times | Serif font |
| -mono | courier | Monospace font |
| -symbol | zapfdingbats | Symbol font |
| -pages | 1-1000000 | Pages to output (first-last) |
| -layers | image:rect:ellipse:curve:arc:line:poly:text:list | Drawing order |
| -grid | 0 | Draw a grid at specified % (0 for no grid) |
| -pagesize | Letter | Page size (w,h or Legal, Tabloid, A[3-5], ArchA, 4R, Index) |
| -fontdir | \$HOME/deckfonts | Font directory |
| -outdir | Current directory | Output directory |
| -stdout | false | Output to standard output |
| -author | "" | Document author |
| -title | "" | Document title |

command examples

```
decksh -o file.xml file.dsh; pdfdeck file.xml
```

process file.dsh to file.xml to file.pdf

```
decksh file.dsh | pdfdeck -stdout - > output.pdf
```

Pipe the output from decksh, making output.pdf

```
pdfdeck -pagesize 1920,1080 -pages 10-20 -grid 5 file.xml
```

render pages 10-20 to file.pdf, page size of 1920 (width) x 1080 (height) pixels, on a 5% grid

```
pdfdeck -sans FiraSans-Regular -serif Charter-Regular -mono Inconsolata-Bold file.xml
```

use FiraSans-Regular.ttf, Charter-Regular.ttf, and Inconsolata-Bold.ttf from the deckfonts directory

```
echo file.dsh|entr -s 'decksh file.dsh|pdfdeck -stdout - > f.pdf; pkill -HUP mupdf'
```

When file.dsh changes, make f.pdf, and refresh the viewer