

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

```
$ cd $HOME
```

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

Install fonts into \$HOME/deckfonts

```
$ decksh -? ; pdfdeck -?
```

Do a test run of decksh and pdfdeck

Installing decksh and pdfdeck binaries

Mac (M1, Intel), Linux (Intel), Windows (32, 64-bit)

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








 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

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

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

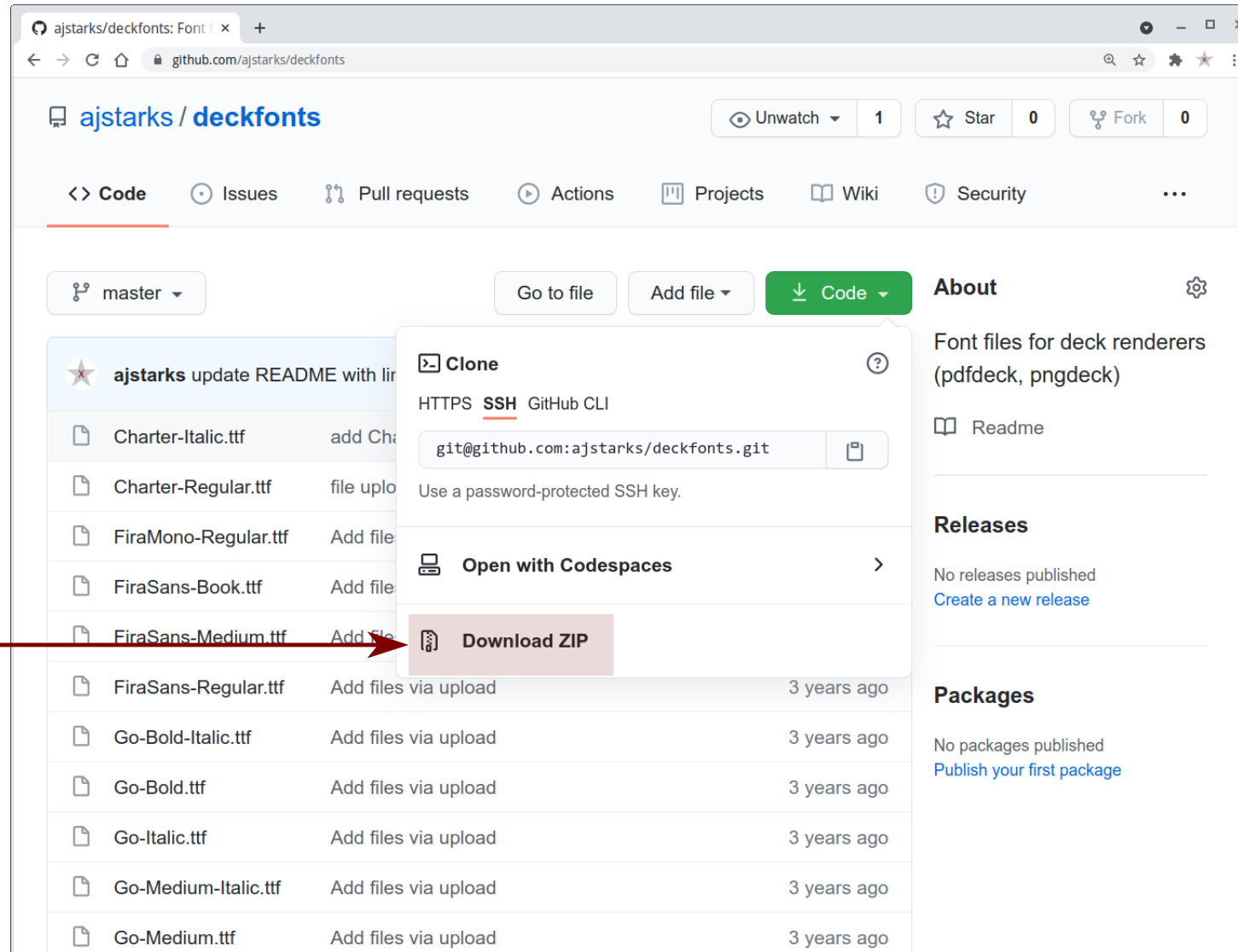


 darwin-amd64-pdfdeck	1 hour ago
 darwin-arm64-pdfdeck	1 hour ago
 linux-amd64-pdfdeck	1 hour ago
 windows-386-pdfdeck.exe	1 hour ago
 windows-amd64-pdfdeck.exe	1 hour ago

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

Downloading the fonts

Download, unzip
to your home
directory.



Default Fonts

Times Roman

times
timesi
timesb
timesbi

Hamburgevons

Helvetica

helvetica
helveticai
helveticab
helveticabi

Hamburgevons

Courier

courier
courieri
courierb
courierbi

Hamburgevons

Alternative serif, sans, mono

Charter

Charter-Regular

Charter-Italic

Hamburgevons

Fira Sans

FiraSans-Book

FiraSans-Medium

FiraSans-Regular

Hamburgevons

Inconsolata

Inconsolata-Regular

Inconsolata-Bold

Inconsolata-Medium

Inconsolata-Condensed

Inconsolata-Black

Hamburgevons

Sans fonts

IBM Plex

IBMPlexSans-Regular

IBMPlexMono-Regular

Hamburgevons

Noto Sans

NotoSans-Regular

NotoMono-Regular

Hamburgevons

Public Sans

PublicSans-Italic

PublicSans-Light

PublicSans-Medium

PublicSans-Regular

PublicSans-SemiBold

PublicSans-BoldItalic

PublicSans-Bold

Hamburgevons

Symbol fonts

Zapf Dingbats

zapfdingbats



Gophers

Gophers



State Face

stateface



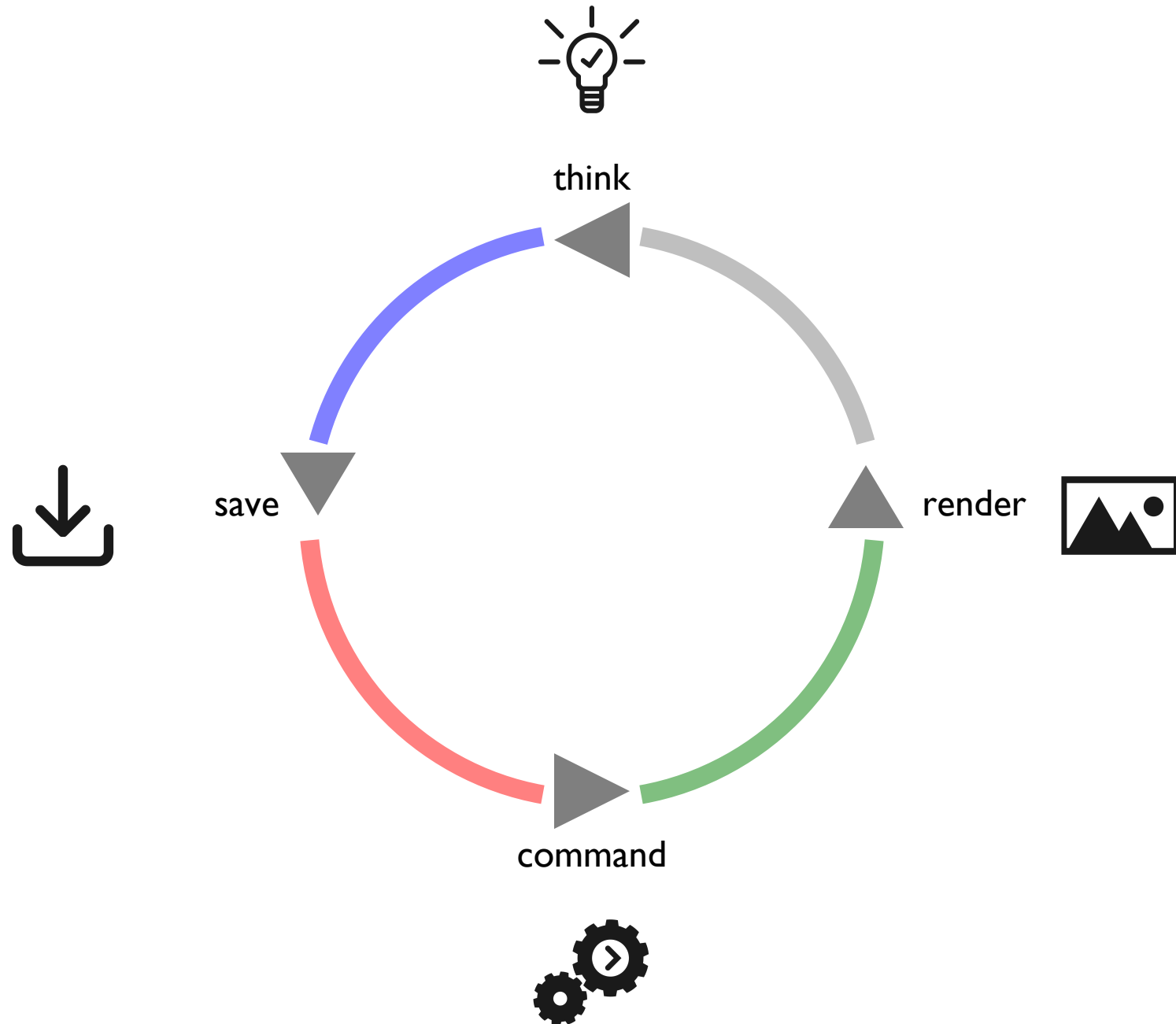
Wee People

weepeople



Running

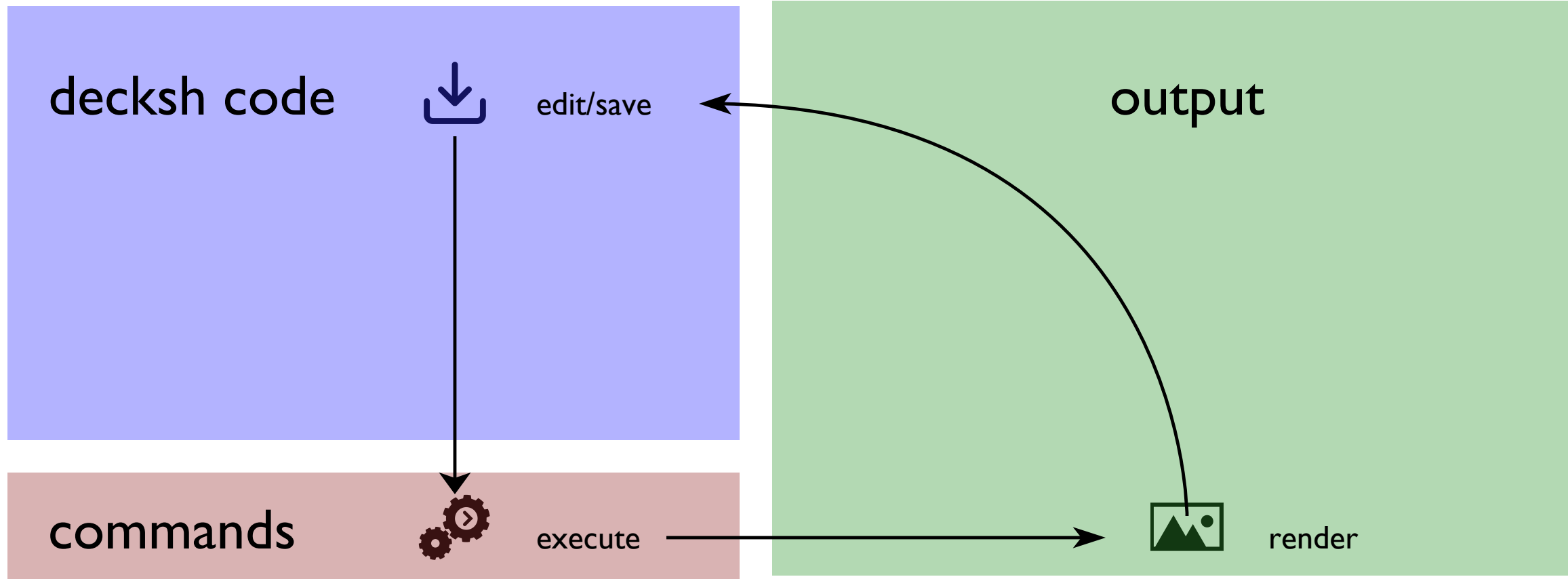
workflow



Working setup

editor

viewer

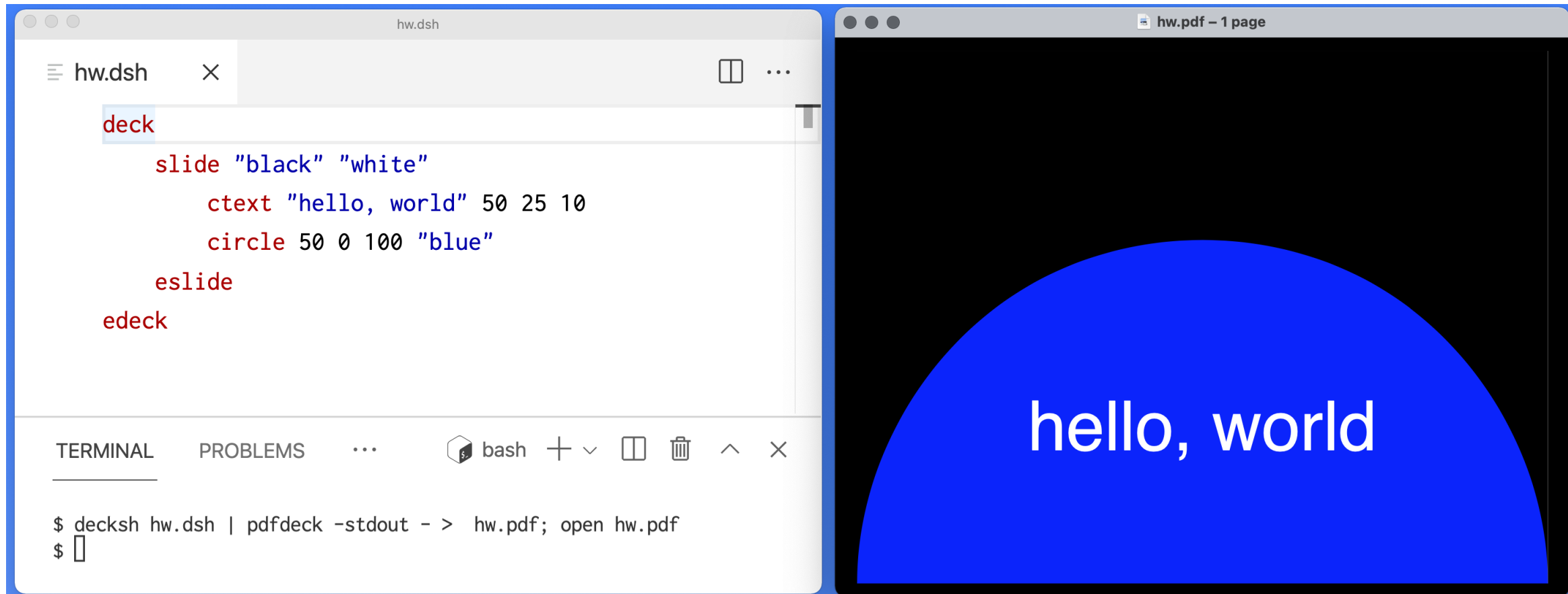


terminal

Mac OS

editor: VSCode

viewer: Preview

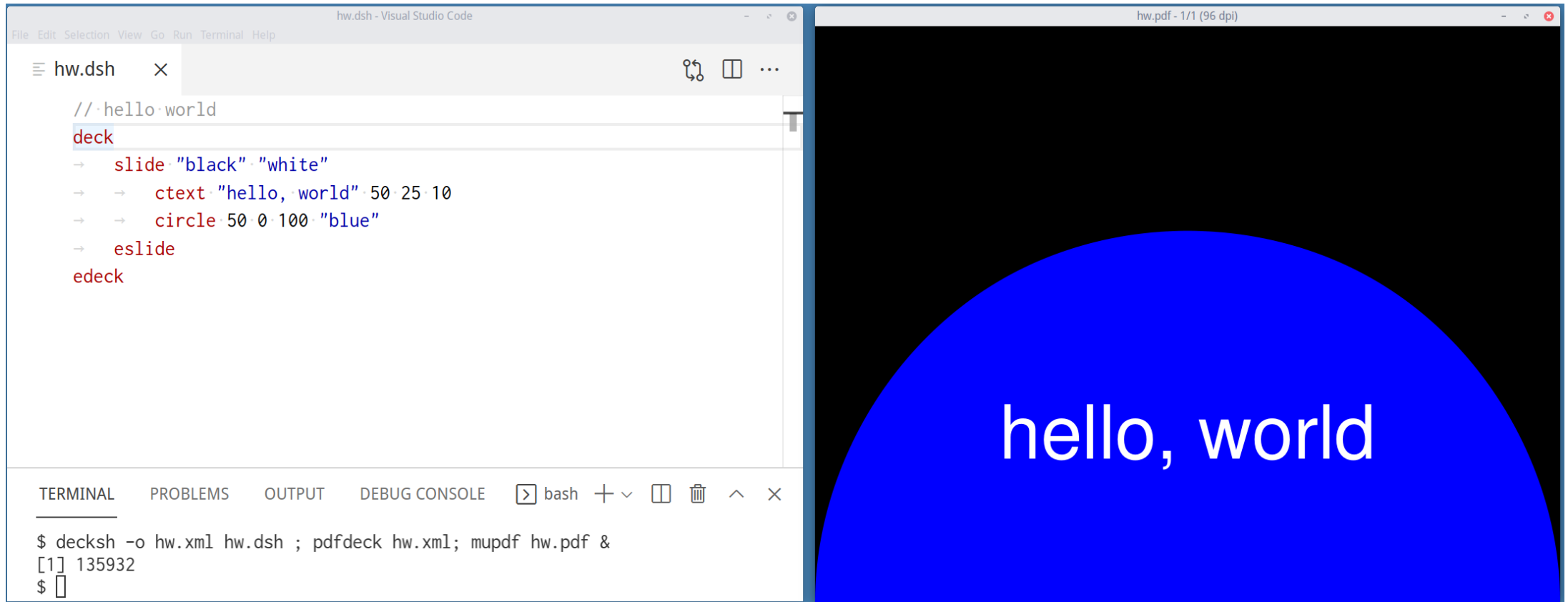


terminal: VSCode

Linux

editor: VSCode

viewer: mupdf

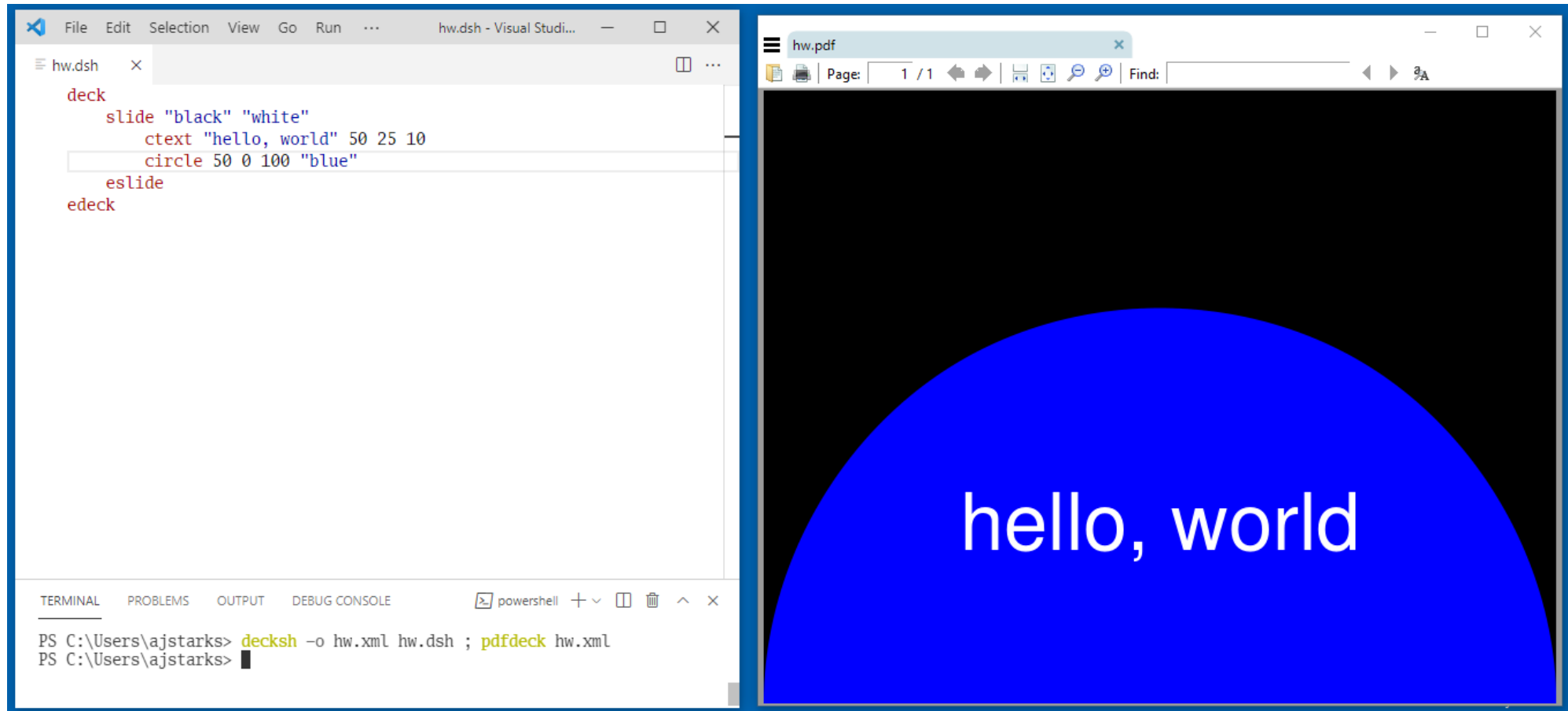


terminal: VSCode

Windows

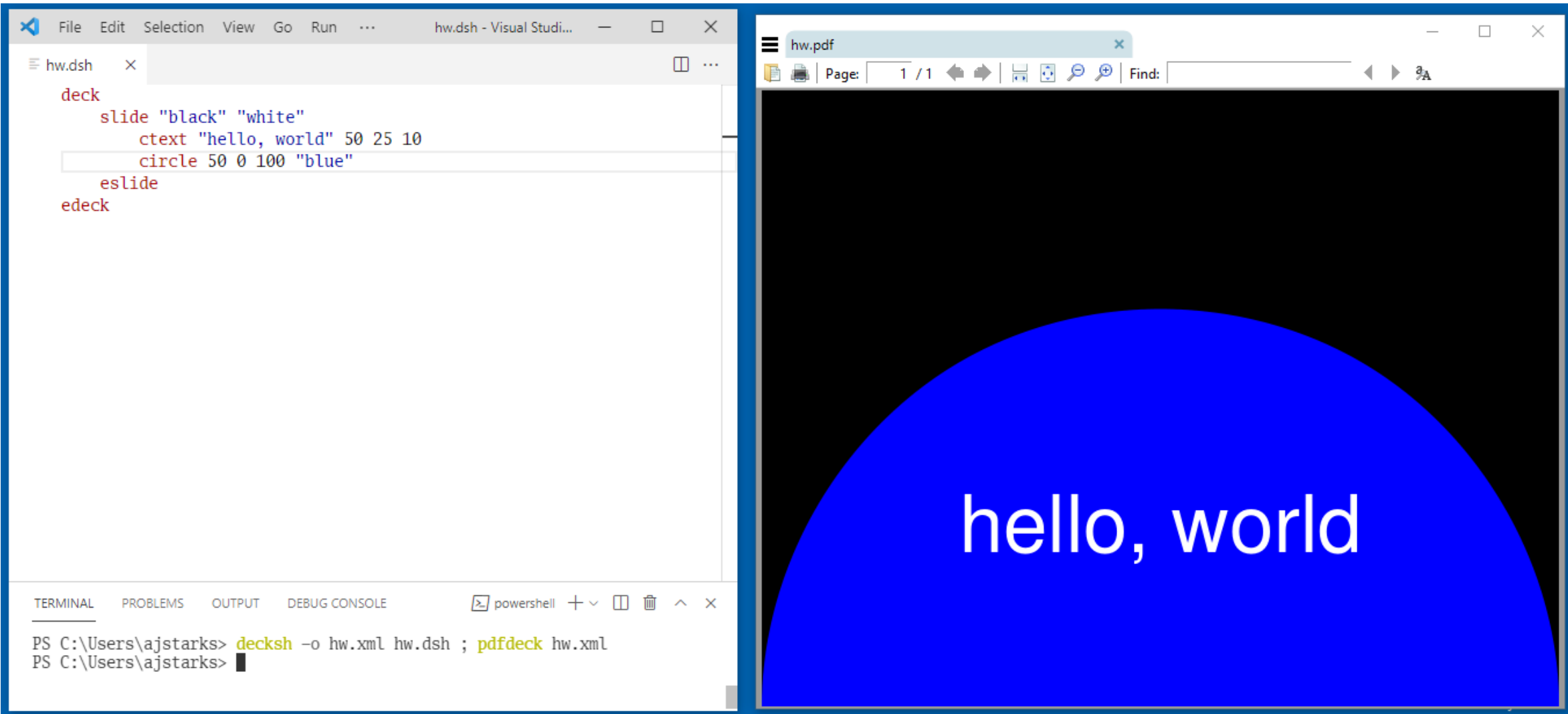
editor: VSCode

viewer: Sumatra PDF



terminal: VSCode

Render

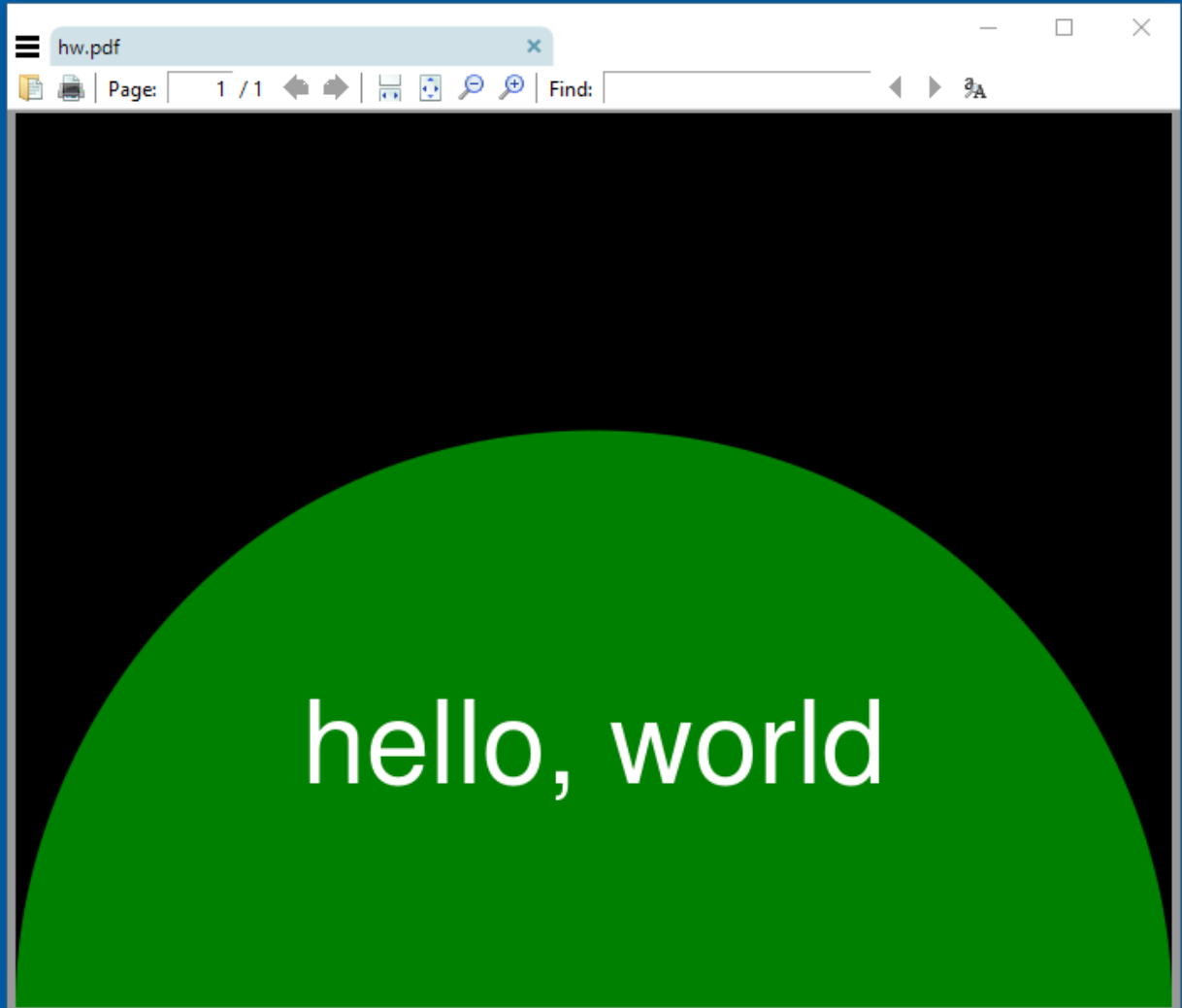


Update

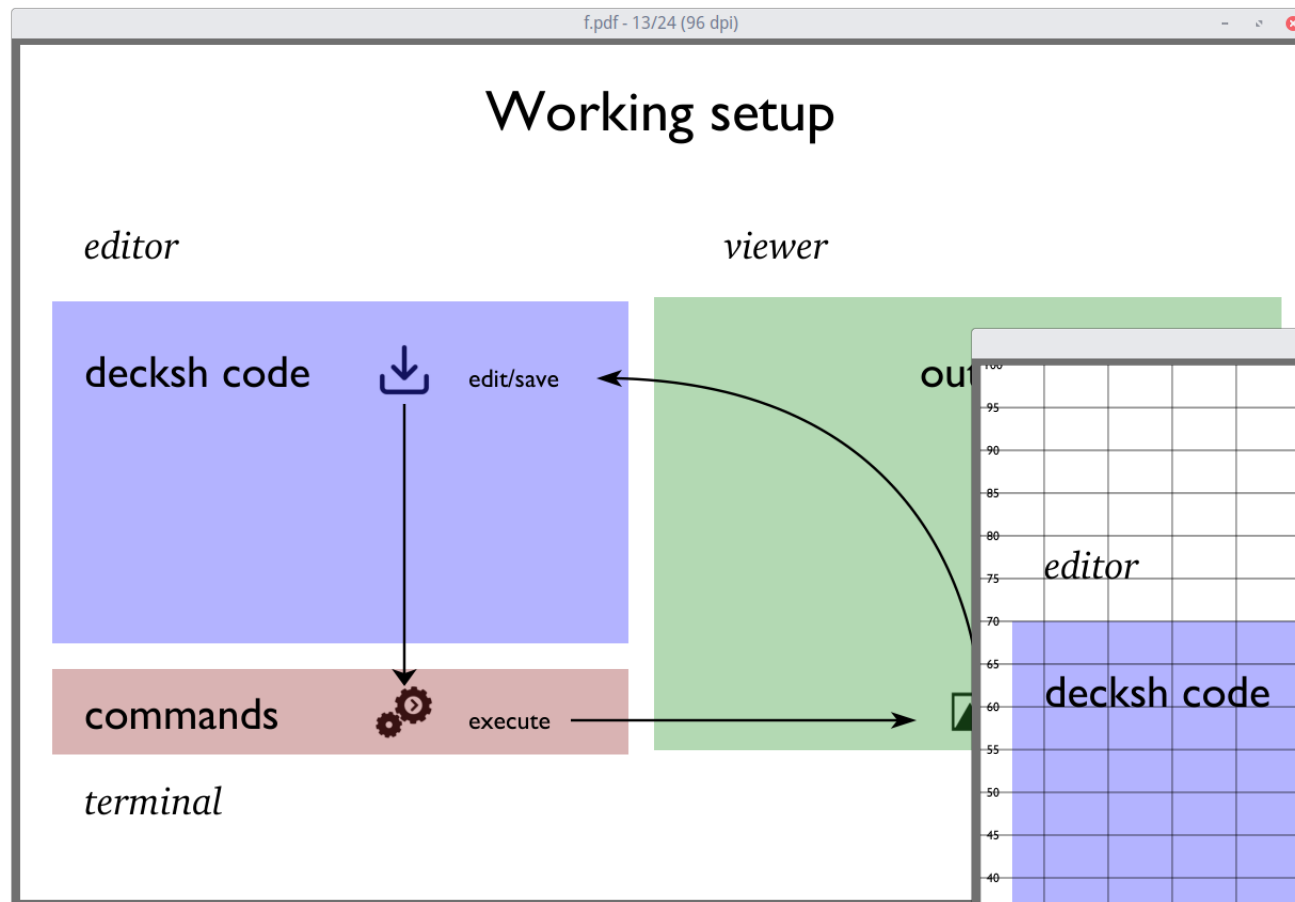
```
hw.dsh
deck
  slide "black" "white"
    ctext "hello, world" 50 25 10
    circle 50 0 100 "green"
  eslide
edeck
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE powershell

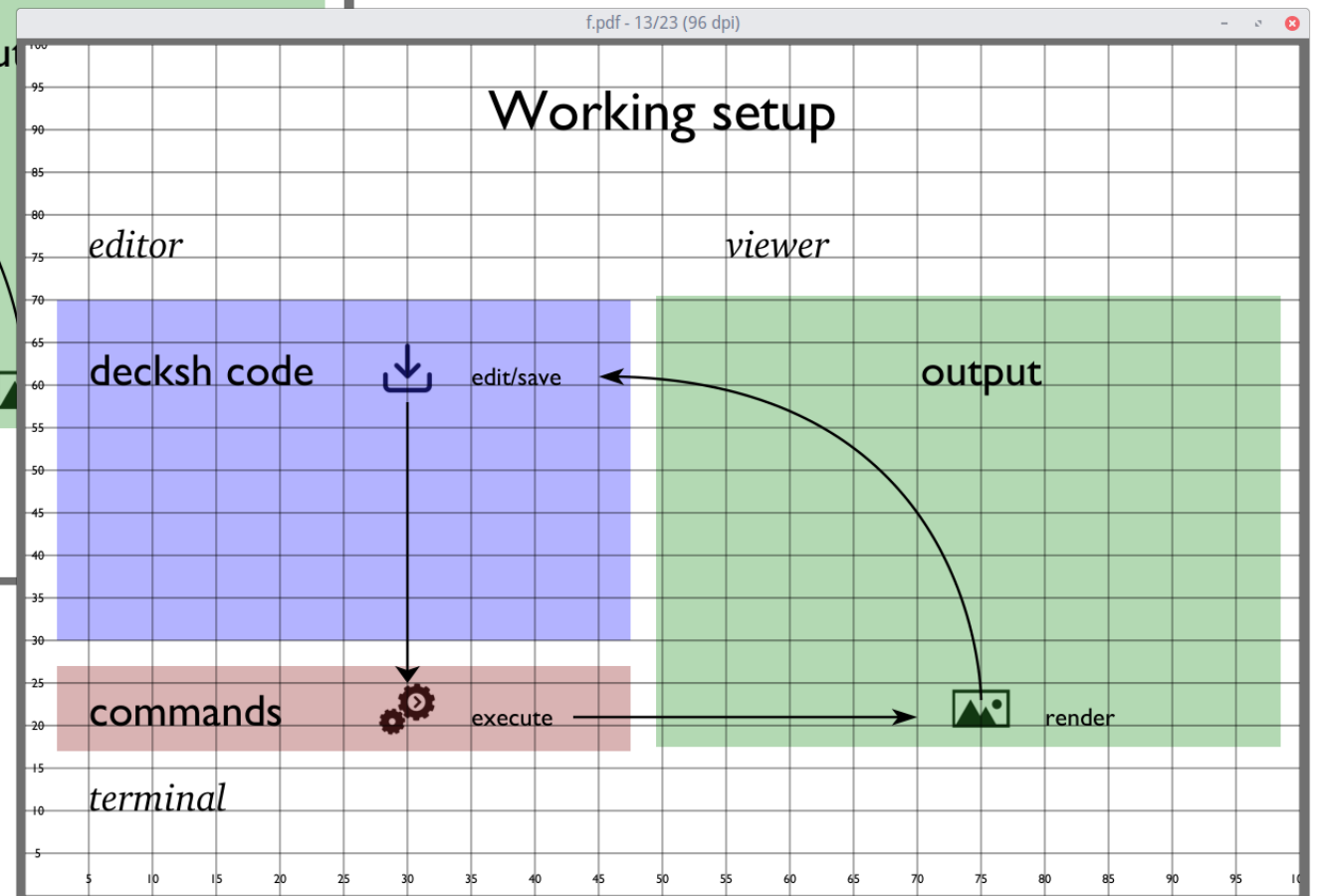
```
PS C:\Users\ajstarks> decksh -o hw.xml hw.dsh ; pdfdeck hw.xml
PS C:\Users\ajstarks> decksh -o hw.xml hw.dsh ; pdfdeck hw.xml
PS C:\Users\ajstarks>
```



Using the -grid option



no grid



-grid 5

The command line

decksh command usage

decksh	<i>read from stdin, write to stdout</i>
decksh in.dsh	<i>read from file, write to stdout</i>
decksh -o out.xml	<i>read from stdin, write to file</i>
decksh -o out.xml in.dsh	<i>read from file, write to file</i>

decksh example.dsh | pdfdeck ...

pdfdeck [options] inputfile

<i>Option</i>	<i>Default</i>	<i>Description</i>
-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)
-pagesize	Letter	Page size (w,h or Legal, Tabloid, A[3-5], ArchA, 4R, Index)
-grid	false	Draw a percent grid
-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