EECS 201: ANSI

Written by Zalan Shah

So… Colors, huh?

- Odds are most (if not all) of your EECS projects have been in the terminal

So… Colors, huh?

- Odds are most (if not all) of your EECS projects have been in the terminal

- Some pretty insane projects, but it can definitely be a little bland

So… Colors, huh?

- Odds are most (if not all) of your EECS projects have been in the terminal

- Some pretty insane projects, but it can definitely be a little bland

- You've probably noticed some programs show colors

So… Colors, huh?

- Odds are most (if not all) of your EECS projects have been in the terminal

- Some pretty insane projects, but it can definitely be a little bland

- You've probably noticed some programs show colors

  - Vim

So… Colors, huh?

- Odds are most (if not all) of your EECS projects have been in the terminal

- Some pretty insane projects, but it can definitely be a little bland

- You've probably noticed some programs show colors

  - Vim

  - Git

So… Colors, huh?

- Odds are most (if not all) of your EECS projects have been in the terminal

- Some pretty insane projects, but it can definitely be a little bland

- You've probably noticed some programs show colors

  - Vim

  - Git

- How can we do that?

Plus…

- Have you ever wondered how Vim clears your screen when you open it, and then brings back your terminal history when you're done with it?

The Answer: ANSI

- American National Standard Institute

The Answer: ANSI

- American National Standard Institute

- Maintain a bunch of escape codes that can be used at the command line

The Answer: ANSI

- American National Standard Institute

- Maintain a bunch of escape codes that can be used at the command line

- They can do a bunch of things:

The Answer: ANSI

- American National Standard Institute

- Maintain a bunch of escape codes that can be used at the command line

- They can do a bunch of things:

The Answer: ANSI

- American National Standard Institute

- Maintain a bunch of escape codes that can be used at the command line

- They can do a bunch of things:

  - Colorize your text

The Answer: ANSI

- American National Standard Institute

- Maintain a bunch of escape codes that can be used at the command line

- They can do a bunch of things:

    - Colorize your text

    - Maintain location of your cursor

The Answer: ANSI

- American National Standard Institute

- Maintain a bunch of escape codes that can be used at the command line

- They can do a bunch of things:

    - Colorize your text

    - Maintain location of your cursor

    - Wipe your screen

    - etc.

The Answer: ANSI

- American National Standard Institute

- Maintain a bunch of escape codes that level up your personal projects

- They tend to get super complicated and hard to understand, so let's walk through how to write them
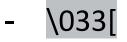
8-bit Colors: Basic Example

- Print out Hello, World in red text

  - echo "\033[0;31mHello, World!"

8-bit Colors: So What Just Happened?

`\033[0;31mHello, World!`

8-bit Colors: So What Just Happened?

\033[0;31mHello, World!

- \033[

8-bit Colors: So What Just Happened?

\033[0;31mHello, World!

- \033[
  - Escape Sequence
    - Let's the terminal know you're about to feed it an ANSI Code

8-bit Colors: So What Just Happened?

\033[0;31mHello, World!

- \033[
  - Escape Sequence
    - Let's the terminal know you're about to feed it an ANSI Code
  - Could also use \e[

8-bit Colors: So What Just Happened?

\033[0;31mHello, World!

- \033[
  - Escape Sequence
    - Let's the terminal know you're about to feed it an ANSI Code
  - Could also use \e[
- 0

8-bit Colors: So What Just Happened?

\033[0;31mHello, World!

- \033[
  - Escape Sequence
    - Let's the terminal know you're about to feed it an ANSI Code
  - Could also use \e[
- 0
  - Reset all color effects

8-bit Colors: So What Just Happened?

\033[0;31mHello, World!

- \033[
  - Escape Sequence
    - Let's the terminal know you're about to feed it an ANSI Code
  - Could also use \e[
- 0
  - Reset all color effects
- 31

8-bit Colors: So What Just Happened?

\033[0;31mHello, World!

- \033[
  - Escape Sequence
    - Let's the terminal know you're about to feed it an ANSI Code
  - Could also use \e[
- 0

  - Reset all color effects
- 31

  - Set the foreground color (i.e. text color) to red

8-bit Colors: So What Just Happened?

\033[0;31mHello, World!

- \033[
  - Escape Sequence
    - Let's the terminal know you're about to feed it an ANSI Code
  - Could also use \e[
- 0
  - Reset all color effects
- 31
  - Set the foreground color (i.e. text color) to red
- m

8-bit Colors: So What Just Happened?

\033[0;31mHello, World!

- \033[
  - Escape Sequence
    - Let's the terminal know you're about to feed it an ANSI Code
  - Could also use \e[
- 0
  - Reset all color effects
- 31
  - Set the foreground color (i.e. text color) to red
- m
  - Finish the sequence

8-bit Colors: All the Colors

Color	Foreground	Background
Black	30	40
Red	31	41
Green	32	42
Yellow	33	43
Blue	34	44
Magenta	35	45
Cyan	36	46
White	37	47
Default	39	49
Reset	0	0

8-bit Colors: So What do These Do?

- echo -e "Hello, \033[31;0;36;44mWor\033[0mld!"

- echo -e "\033[33;43mThis'll print in magenta"

Color	Foreground	Background
Black	30	40
Red	31	41
Green	32	42
Yellow	33	43
Blue	34	44
Magenta	35	45
Cyan	36	46
White	37	47
Default	39	49
Reset	0	0

8-bit Colors: Whoopsie!

- Playing with these, you might notice colors bleeding into your prompt

Color	Foreground	Background
Black	30	40
Red	31	41
Green	32	42
Yellow	33	43
Blue	34	44
Magenta	35	45
Cyan	36	46
White	37	47
Default	39	49
Reset	0	0

8-bit Colors: **Whoopsie!**

- Playing with these, you might notice colors bleeding into your prompt
- Let's fix that with reset!

Color	Foreground	Background
Black	30	40
Red	31	41
Green	32	42
Yellow	33	43
Blue	34	44
Magenta	35	45
Cyan	36	46
White	37	47
Default	39	49
Reset	0	0

Truecolor: This isn't the 70's anymore

- If all you really needed was a few basic colors, the basic 8-bit ones work fine
- This is 2022
  - We can do better!
- Let's use RGB colors to display anything we want

Truecolor: What's RGB?

- RGB stands for Red, Green, and Blue

Truecolor: What's RGB?

- RGB stands for Red, Green, and Blue
  - Three Primary Colors

Truecolor: What's RGB?

- RGB stands for Red, Green, and Blue
  - Three Primary Colors
  - Combining them in different intensities allows you to create 16777216 colors

Truecolor: What's RGB?

- RGB stands for Red, Green, and Blue
  - Three Primary Colors
  - Combining them in different intensities allows you to create 16777216 colors
- [Live Demo](#)

Truecolor: The Scheme

\033[38;2;{r};{g};{b}mHello, World!

\033[48;2;{r};{g};{b}mHello, World!

Manipulating the Terminal

- Colors are just a facet of all the things you can manipulate in the terminal
- Let's wipe the screen temporarily like in Vim!

Manipulating the Terminal

- Colors are just a facet of all the things you can manipulate in the terminal
- Let's wipe the screen temporarily like in Vim!
    - \033[?1049h
        - Take terminal to alternate buffer
    - \033[?1049l
        - Return terminal to main buffer
    - \033[2J
        - Clear buffer

Manipulating the Terminal

- How about moving the cursor?

Manipulating the Terminal

- How about moving the cursor?
  - \033[{N}A
    - Up {N} rows

Manipulating the Terminal

- How about moving the cursor?
    - \033[{N}A
        - Up {N} rows
    - \033[{N}B
        - Down {N} rows

Manipulating the Terminal

- How about moving the cursor?
  - \033[{N}A
    - Up {N} rows
  - \033[{N}B
    - Down {N} rows
  - \033[{N}C
    - Right {N} columns

Manipulating the Terminal

- How about moving the cursor?
  - \033[{N}A
    - Up {N} rows
  - \033[{N}B
    - Down {N} rows
  - \033[{N}C
    - Right {N} columns
  - \033[{N}D
    - Left {N} columns

Some Other Cool Ones

- `\007`
  - bell noise

Some Other Cool Ones

- \007
  - bell noise
- \033[{1-9}m
  - In order: bold, dim, faint, italics, underline, blinking, inverse, invisible, and strikethrough modes

Some Other Cool Ones

- `\007`
  - bell noise
- `\033[{1-9}m`
  - In order: bold, dim, faint, italics, underline, blinking, inverse, invisible, and strikethrough modes
- `\0337`
  - Save cursor's location

Some Other Cool Ones

- \007
  - bell noise
- \033[{1-9}m
  - In order: bold, dim, faint, italics, underline, blinking, inverse, invisible, and strikethrough modes
- \0337
  - Save cursor's location
- \0338
  - jump back to cursor's location

Some Other Cool Ones

- \007
  - bell noise
- \033[{1-9}m
  - In order: bold, dim, faint, italics, underline, blinking, inverse, invisible, and strikethrough modes
- \0337
  - Save cursor's location
- \0338
  - jump back to cursor's location

And many, many, more!

Final Notes

- Always account for what happens if you're redirecting your output

Final Notes

- Always account for what happens if you're redirecting your output
  - If being redirected to a text files, get rid of ANSI sequences!

Final Notes

- Always account for what happens if you're redirecting your output
  - If being redirected to a text files, get rid of ANSI sequences!
- Not every terminal supports every last thing

Final Notes

- Always account for what happens if you're redirecting your output
  - If being redirected to a text files, get rid of ANSI sequences!
- Not every terminal supports every last thing
  - Some terminals don't support truecolor, others don't support color at all, others don't support saving cursors

Final Notes

- Always account for what happens if you're redirecting your output
    - If being redirected to a text files, get rid of ANSI sequences!
- Not every terminal supports every last thing
    - Some terminals don't support truecolor, others don't support color at all, others don't support saving cursors
- Always account for these edge cases!

For More Information…

- https://gist.github.com/fnky/458719343aabd01cfb17a3a4f7296797
    - Personal favorite guide
- https://www.gnu.org/software/screen/manual/html_node/Control-Sequences.html

Thank You!

- If you guys ever want to reach out to talk to me about anything, always feel free to do so!
- Thanks for an awesome semester! Y'all are the best :)