Full Stack Snippets.

From Chris' Full Stack Blog.

Bash

---

buildColorPrompt

Letter-level color changes for your bash prompt!

From post: Awesome Colors for Shell Prompts!

`buildColorPrompt.sh`
bash

```bash
function buildColorPrompt() {

    # I always like showing what directory I am in (special character "\w" in
PS1) - store the equivalent in this 'directory' variable
    directory=$(pwd)

    # Modify these to whatever you'd like!
    PROMPT_TEXT="awesome-shell-prompt-colors@awesome-machine [$directory] "

    # Colors seperated by comma - acceptable values are:
    # black, white, red, green, yellow, blue, magenta, cyan, light gray, light
red, light green, light yellow, light blue, light magenta, light cyan
    PROMPT_COLORS="red,white,blue"

    # Colors!
    BLACK="\e[30m"
    WHITE="\e[97m"
    RED="\e[31m"
    GREEN="\e[32m"
    YELLOW="\e[33m"
```

```
BLUE="\e[34m"
MAGENTA="\e[35m"
CYAN="\e[36m"
LIGHT_GRAY="\e[37m"
DARK_GRAY="\e[90m"
LIGHT_RED="\e[91m"
LIGHT_GREEN="\e[92m"
LIGHT_YELLOW="\e[93m"
LIGHT_BLUE="\e[94m"
LIGHT_MAGENTA="\e[95m"
LIGHT_CYAN="\e[96m"

# End formatting string
END_FORMATTING="\[\e[0m\]"

# split PROMPT_COLORS into array
count=0
IFS=','
for x in $PROMPT_COLORS
do
    colors_array[$count]=$x
    ((count=count+1))
done
unset IFS

# break PROMPT_TEXT into character array
letters=()
for (( i=0 ; i < ${#PROMPT_TEXT} ; i++ )) {
    letters[$i]=${PROMPT_TEXT:$i:1}
}

# build prompt with colors
color_index=0
ps1='\['
for (( i=0 ; i < ${#letters[@]} ; i++ )) {
    # Determine color in this giant case statement
    color="${colors_array[color_index]}"
    case $color in
        "black")
            COLOR=$BLACK
            ;;
        "red")
            COLOR=$RED
            ;;
        "green")
            COLOR=$GREEN
            ;;
        "yellow")
```

```
            COLOR=$YELLOW
            ;;
        "blue")
            COLOR=$BLUE
            ;;
        "magenta")
            COLOR=$MAGENTA
            ;;
        "cyan")
            COLOR=$CYAN
            ;;
        "light gray")
            COLOR=$LIGHT_GRAY
            ;;
        "dark gray")
            COLOR=$DARK_GRAY
            ;;
        "light red")
            COLOR=$LIGHT_RED
            ;;
        "light green")
            COLOR=$LIGHT_GREEN
            ;;
        "light yellow")
            COLOR=$LIGHT_YELLOW
            ;;
        "light blue")
            COLOR=$LIGHT_BLUE
            ;;
        "light magenta")
            COLOR=$LIGHT_MAGENTA
            ;;
        "light cyan")
            COLOR=$LIGHT_CYAN
            ;;
        "white")
            COLOR=$WHITE
            ;;
        *)
            COLOR=$WHITE
            ;;
    esac

    # add to ps1 var - color, then letter, then the end formatter
    ps1+=$COLOR"${letters[$i]}"

    # reset color index if we are at the end of the color array, otherwise
increment it
```

```
        if (( $color_index == ${#colors_array[@]} - 1 ))
        then
            color_index=0
        else
            ((color_index=color_index+1))
        fi
    }
    ps1+="$END_FORMATTING\]"

    # Finally: set the PS1 variable
    PS1=$ps1
}

# Set the special bash variable PROMPT_COMMAND to our custom function
PROMPT_COMMAND=buildColorPrompt;
```

Usage

bash

```
# Assuming the buildColorPrompt function is in your .bash_profile:
# Set the special bash variable PROMPT_COMMAND to our custom function
PROMPT_COMMAND=buildColorPrompt;
```

---

sendSlackMessage

Util function to send a Slack message from bash.

From post: The Last Bitbucket Pipelines Tutorial You'll Ever Need: Mastering CI and CD

sendSlackMessage.sh

bash

```
function sendSlackMessage {
    curl -X POST -H 'Content-type: application/json' --data '{"text":"$1"}' $2
}
```

Usage

bash

```
# the two parameters are 1. the message, and 2. the webhook url
sendSlackMessage "Hello World!" https://yourslackwebhookurl/secret/supersecret
```

---

supercurl

Get detailed network times for a website.

From post: Magento 2 IP Location Detection (GeoIP) and Store Context Control Using the ipstack API

supercurl.sh

bash

```
function supercurl() {
    curl -s -w '\nLookup time:\t%{time_namelookup}\nConnect time:\t%
{time_connect}\nAppCon time:\t%{time_appconnect}\nRedirect time:\t%
{time_redirect}\nPreXfer time:\t%{time_pretransfer}\nStartXfer time:\t%
{time_starttransfer}\n\nTotal time:\t%{time_total}\n' -o /dev/null $1
}
```

Usage

bash

```
# simply pass the website you want "super" curl as the first argument :)
supercurl google.com

# example output:
# Lookup time:    0.061647
# Connect time:   0.281195
# AppCon time:    0.000000
# Redirect time:  0.000000
# PreXfer time:   0.281248
# StartXfer time: 0.759128

# Total time:     0.761387
```

zsh

---

buildColorPrompt

Letter-level color changes for your zsh prompt!

From post: Awesome Colors for Shell Prompts!

buildColorPrompt.sh

bash

```bash
function buildColorPrompt() {

    # I always like showing what directory I am in
    directory=$(pwd)

    # Modify these to whatever you'd like!
    PROMPT_TEXT="youruser@yourmachine [$directory]"

    # Comma seperated colors - as many or as few as you'd like
    PROMPT_COLORS="15"

    # This will be the color of everything in the input part of the prompt
(here set to 15 = white)
    PROMPT_INPUT_COLOR="15"

    # split PROMPT_COLORS into array
    colors_array=("${(@s/,/)PROMPT_COLORS}") # @ modifier

    # break PROMPT_TEXT into character array
    letters=()
    for (( i=1 ; i < ${#PROMPT_TEXT}+1 ; i++ )) {
        letters[$i]=${PROMPT_TEXT:$i-1:1}
    }

    # build prompt with colors
    color_index=1
    ps1=""
    for (( i=1 ; i < ${#letters[@]}+1 ; i++ )) {
        # Determine color in this giant case statement
        color="${colors_array[color_index]}"

        # add to ps1 var - color, then letter, then the end formatter
        ps1+="%F{$color}${letters[$i]}"

        # reset color index if we are at the end of the color array, otherwise
increment it
        if (( $color_index == ${#colors_array[@]} ))
        then
            color_index=1
        else
            ((color_index=color_index+1))
        fi
    }

    # end color formating
    ps1+="%F{$PROMPT_INPUT_COLOR} %# "
```

```
    # Finally: set the PROMPT variable
    PROMPT=$ps1
}

# set the precmd() hook to our custom function
precmd() {
   buildColorPrompt;
}
```

Usage

bash

```
# Assuming the buildColorPrompt function is in your .zprofile:
# Set the precmd() hook to our custom function
precmd() {
   buildColorPrompt;
}
```