

macOS, iTerm, ZSH

# iTerm2 Customize on macOS

```
ref from : https://gist.github.com/kevin-smets/8568070

get and install iTerm2

iTerm->Preferences->Profiles->colors->load presets. : Solarized Dark theme

get and install Oh My Zsh

sh -c "$(curl -fsSL
https://raw.githubusercontent.com/robbyrussell/oh-my-zsh/master/tools/install.sh)"
edit ~/.zshrc -> ZSH_THEME="agnoster"

install a patched font (Meslo)

https://github.com/powerline/fonts/blob/master/Meslo%20Slashed/Meslo%20LG%20M%20Regular%20for%20Powerline.ttf

iTerm->Preferences->Profiles->Text->Change Font

install zsh-syntax-highlighting
git clone https://github.com/zsh-users/zsh-syntax-highlighting.git
echo "source ${(%)-PWD}/zsh-syntax-highlighting/zsh-syntax-highlighting.zsh"
>> ${ZDOTDIR:-$HOME}/.zshrc

test font
echo "\ue0b0 \u00b1 \ue0a0 \u27a6 \u2718 \u26a1 \u2699"

using multi-line prompt

ref from : https://wayhome25.github.io/etc/2017/03/12/zsh-alias/

vi ~/.zshrc
# to delete computer name in prompt
prompt_context() {
  if [[ "$USER" != "$DEFAULT_USER" || -n "$SSH_CLIENT" ]]; then
    prompt_segment black default "%(!.%{F{yellow}}%)$USER"
  fi
}

vi ~/.oh-my-zsh/themes/agnoster.zsh-theme
## Main prompt
build_prompt() {
```

```
RETVAL=$?  
prompt_status  
prompt_virtualenv  
prompt_context  
prompt_dir  
prompt_git  
prompt_bzr  
prompt_hg  
prompt_newline # 멀티라인 적용  
prompt_end  
}  
  
# 멀티라인 적용, 커버모양 변경  
prompt_newline() {  
    if [[ -n $CURRENT_BG ]]; then  
        echo -n "%k%F{$CURRENT_BG}%}$SEGMENT_SEPARATOR  
%(?.%F{$CURRENT_BG}.%F{red})>%f"  
  
    else  
        echo -n "%k%"  
    fi  
  
    echo -n "%f%"  
    CURRENT_BG=' '  
}
```

## Preparation

1. get and install iTerm
2. install brew
3. get and install font
4. get solarized theme
5. install zsh, oh-my-zsh

## Settings

1. setting on item  
  1. theme
  2. text
2. setting zsh as default
3. set theme in ~/.zshrc
4. set other stuffs in ~/.oh-my-zsh/themes/agnoster.zsh-theme

```
echo "\ue0b0 \u00b1 \ue0a0 \u27a6 \u2718 \u26a1 \u2699"
```

## Others

### 프롬프트 사용자 이름 삭제

```
$ vim ~/.zshrc
```

```
prompt_context() {  
  if [[ "$USER" != "$DEFAULT_USER" || -n "$SSH_CLIENT" ]]; then  
    prompt_segment black default "%(!.%{F{yellow}%.)$USER"  
  fi  
}
```

### Word jump 활성화하기

```
## word jumps 활성화 하기
```

- 기본적으로 oh-my-zsh에서는 `option + 방향키` 조합의 단어별 이동이 불가능하다. 이를 수정하려면 별도의 설정수정이 필요하다.
- iTerm -> Preferences -> Profiles -> Keys 로 이동한다.
- `+` 버튼을 누르고 아래 설정을 추가한다.

```
``shell  
# Option + right  
↵→  
Send Escape Sequence  
f  
# Option + left  
↵←  
Send Escape Sequence  
b
```

### Syntax Highlighting

```
# 설치
```

```
$ brew install zsh-syntax-highlighting
```

```
# ~/.zshrc 에 아래 문구 추가
```

```
source /usr/local/share/zsh-syntax-highlighting/zsh-syntax-highlighting.zsh
```

## 테마변경

```
vi .zshrc  
ZSH_THEME="robbyrussell"
```

## Multiline Prompt

```
vi ~/.oh-my-zsh/themes/agnoster.zsh-theme
```

```
## Main prompt  
build_prompt() {  
    RETVAL=$?  
    prompt_status  
    prompt_virtualenv  
    prompt_context  
    prompt_dir  
    prompt_git  
    prompt_bzr  
    prompt_hg  
    prompt_newline # 멀티라인 적용  
    prompt_end  
}  
  
# 멀티라인 적용, 커버모양 변경  
prompt_newline() {  
    if [[ -n $CURRENT_BG ]]; then  
        echo -n "%k%F{$CURRENT_BG}%}$SEGMENT_SEPARATOR  
%(?.%F{$CURRENT_BG}.%F{red})$SEGMENT_SEPARATOR" # }%f"  
  
    else  
        echo -n "%k%"  
    fi  
  
    echo -n "%f%"  
    CURRENT_BG=''  
}
```

```
}
```

## Update manually

```
$ upgrade_oh_my_zsh
```

```
Updating Oh My Zsh  
error: cannot pull with rebase: You have unstaged changes.  
error: please commit or stash them.  
There was an error updating. Try again later
```

```
$ cd ~/.oh-my-zsh  
$ git status  
$ git stash  
or  
$ git add .  
$ git commit -m  
$ upgrade_oh_my_zsh
```

- [Manually update oh-my-zsh](#)

From:

<http://www.theta5912.net/> - reth

Permanent link:

[http://www.theta5912.net/doku.php?id=public:computer:iterm\\_customize&rev=1616435622](http://www.theta5912.net/doku.php?id=public:computer:iterm_customize&rev=1616435622)

Last update: **2021/03/23 02:53**

