□ Pri90-01個人ノウハウ備忘録

□ 変更履歴

- 【2019年9月12日】Prediction One追加
- 【2019年9月6日】14T09779記述修正
- 【2019年9月1日】SikuliX追加
- 【2019年7月27日】VSCode関係更新
- 【2019年7月20日】Git関係更新
- 【2019年7月14日】Windows Subsystem for Linux追加
- 【2019年2月19日】SurfaceGo設定関係更新
- 【2019年2月18日】SurfaceGo再インストール履歴更新
- 【2019年2月16日】カテゴリ分け、SurfaceGo追加
- 【2018年11月22日】TCYSS_Repository等の記述修正
- 【2018年11月1日】VSCodeアプリのアップデート方法記載
- 【2018年9月1日】
- ☐ Windows Subsystem for Linux
 - □ コントロールパネル
 - □ Windowsの機能
 - [コントロールパネル] [プログラムと機能] [Windowsの機能]ダイアログを開き、[Windowsの機能の有効化または無効化]をクリックしてダイアログを開き、そこで「Windows Subsystem for Linux」にチェックを入れる。再起動が要求されるので、再起動を実行すること。
 - □ Ubuntuインストール
 - □ Microsoft store
 - □ Ubuntu
 - 221MB
 - □ Ubuntuセットアップ
 - □ 初期化
 - User,Password設定
 - □ 保存フォルダー
 - C:¥Users¥mskn5¥AppData¥Local¥Packages¥CanonicalGroupLimited.UbuntuonWindows_79rhkp1fndgsc
 - file:///C:/Users/mskn5/AppData/Local/Packages/CanonicalGroupLimited.UbuntuonWindows_79rhkp1fndgsc/
 - □ Ubuntu更新
 - bash on Windows ਦ
 - sudo apt update
 - sudo apt upgrade
 - sudo apt -y install language-pack-ja
 - sudo update-locale LANG=ja_JP.UTF8
 - sudo apt -y install manpages-ja manpages-ja-dev
 - □ 各種設定
 - □ bash on Windowsでコピー&ペースト
 - bash on Windowsを起動して、ウィンドウタイトルバーを右クリックするとシステムメニューが現れます。「プロパティ」をクリックして、
 - 「簡易編集モード」をOn
 - □ SDカードをマウント 🗾
 - □ マウントポイントが無い場合は作成する
 - sudo mkdir /mnt/d

- ネットワーク接続に対して警告を出すので、接続を許可する。
- □ 起動
 - 初期設定ファイルを開く
 - X11 Configuration File 「config」を開く
- □ 自動起動設定
 - スタートメニュー右クリックの「ファイル名を指定して実行」から、"shell:startup"を実行
 - スタートアップのウィンドウに、VcXsrvのショートカットをコピー
- □ bash on Windowsで
 - echo 'export DISPLAY=localhost:0.0' >> ~/.bashrc
- □ wsltty
 - □ インストール
 - ☐ mintty/wsltty github 🗾
 - https://github.com/mintty/wsltty/releases/download/3.0.2/wsltty-3.0.2-install.exe
 - Windowsでインストール
 - add to context menu
 - □ 実行
 - Ubuntu Terminal
 - WSL Terminal
- □ Python3インストール
 - sudo apt install python3
 - sudo apt install python3-pip
 - □ 参考
 - □ 削除
 - sudo apt-get remove python-pip
 - sudo apt autoremove python-pip
- □ Anacondaインストール
 - Anacondaサイト 🗾
 - □ Linux版ダウンロード

- □ インストール
 - sh Anaconda3-2019.03-Linux-x86_64.sh
- □ 起動
 - anaconda-navigator
- □ git
 - □ 初期設定
 - git config --global user.name "XXXX"
 - git config --global user.email "XXXX@hogehoge.com"
 - git config --list
 - □ qit パスワード を毎回聞かれる問題の解決方法 🗾
 - □ パスワードの省略
 - sudo git config --global credential.helper 'cache --timeout=3600'
 - GitHubでssh接続する手順~公開鍵・秘密鍵の生成から~ 🗾
- 🗉 gui gui 🔼
 - □ インストール
 - sudo apt-get -y install git-gui
 - □ 起動
 - git gui
- □ VS Codeインストール
 - WSL(Ubuntu 18.04)上でVS Codeを動かす 🗾
 - Visual Studio Code on Linux <a>Z
 - # ダウンロード
 - \$ wget -O code.deb https://go.microsoft.com/fwlink/?LinkID=760868
 - # インストール
 - \$ sudo dpkg -i code.deb
 - □ VS Codeの日本語化
 - □ 日本語フォントのインストール
 - sudo apt-get install fonts-noto-cjk fonts-noto-hinted
 - 拡張機能の「Japanese Language Pack for Visual Studio Code」をインストール
 - □ 日本語入力の設定
 - 当面インストールせず
 - VSCode extension
 - Beautify
 - Bracket Pair Colorizer
 - C/C++
 - Code Runner
 - Docker
 - Dracula Theme
 - file-icons
 - Git History
 - Git Project Manager
 - GitLens -- Git supercharged
 - Go
 - indent-rainbow
 - Japanes Language Pack for Visual....

- Jupyter
- Local History
- Markdown All in One
- markdownlint
- Material Icon Theme
- Python
- View In Browser
- VS Code Jupter Notebook Previewer
- vscode-icons
- VSNotes
- □ VS Code起動
 - code
- □ Windows版VSCodeとの連携
 - ターミナルの選択
 - ☐ C:¥Users¥mskn5¥AppData¥Roaming¥Code¥User
 - - {
 "terminal.integrated.shell.windows": "C:\footnote{\text{C:}\footnote{\text{YWINDOWS}\footnote{\text{YSystem32}\footnote{\text{YWINDOWS}\footnote{\text{YSystem32}\footnote{\text{YWINDOWS}\footnote{\text{YSystem32}\footnote{\text{YWINDOWS}\footnote{\text{YSystem32}\footnote{\text{YWINDOWS}\footnote{\text{YSystem32}\footnote{\text{YWINDOWS}\footnote{\text{YSystem32}\footnote{\text{YSystem32}\footnote{\text{YWINDOWS}\footnote{\text{YSystem32}\footnote
- ☐ Firefox
 - 【WSL】にFirefox&Google-Chromeをインストールする方法! 🗾
- ☐ Chrome
 - WSL の Ubuntu 18.04 に chrome と selenium 環境を作る 🗾
 - □ Xlaunch
 - MultiWindows以外を選択しないとエラーになる
 - sudo google-chrome --no-sandbox
- □ OA
 - プレゼン資料作成
 - □ Xmindからの資料作成
 - 【注意】他のxmindへのリンクがあると出力されない
 - □ FreeMindを活用できるXmind文書
 - □ xmindからFreeMindファイルの作成
 - エクスポート→Freemind
 - 自動生成フォルダーに出力
 - □ FreeMindからxhtmlの作成
 - 【注意】初期表示に開いておきたい項目は、あらかじめ展開しておく
 - 全てのトピックを右に移動させる
 - エクスポート→xhtmlとして(JaveScript版)
 - 【成果物】htmlファイル+imageフォルダ内画像ファイル(複数)
 - Chromeでhtml版を閲覧利用(折り畳み、展開が可能)
 - □ ChromeからPDF版作成
 - 印刷→PDFファイル
 - 【成果物】PDFファイル
 - □ Kindleヘファイルをアップロード
 - 日 リフローフォーマット
 - htmlファイルをメール等で送信

- 固定フォーマット
- □ FreeMindを活用できない文書
 - □ XmindからMS Word文書作成
 - 概要ピクチャーで展開したトピックが多いとエラーになる
 - □ Wordで書式調整
 - Fontをメイリオに変更
 - 行間を最小値Optに
 - 段落前後のスペース削除
 - ページ更新
 - 【成果物】Wordファイル
 - □ XmindマップイメージのPDFファイル作成
 - 印刷→PDFファイル
 - 【成果物】PDFファイル
- □ MetaMoji Noteでのプレゼン、校正
 - PDFを読み込む
 - Wordファイルは、PDFに変換される
- □ OneDriveビジネスへのアップロード
 - PCのフォルダーから更新のあったファイルをアップロード
- □ Kindleパーソナルドキュメントの作成
 - □ アップロード可能フォーマット
 - Kindle形式 (.MOBI、.AZW)
 - Microsoft Word (.DOC、.DOCX)
 - HTML (.HTML、.HTM)
 - RTF (.RTF)
 - Text (.TXT)
 - JPEG (.JPEG、.JPG)
 - GIF (.GIF)
 - PNG (.PNG)
 - BMP (.BMP)
 - PDF (PDF)
 - □ Wordファイルの登録
 - リフロー、リンク、見出しが可能
 - □ PDFファイルの登録
 - リンク、見出しは使えない
 - PowerPoint, ExcelファイルはPDFに変換
 - □ Kindleへのアップロード
 - □ Kindleメールアドレスへ送信
 - 【成果物】Kindleパーソナルドキュメント
 - □ iPad Kindleアプリで、Kindleへファイル転送
 - 【成果物】Kindleパーソナルドキュメント
- □ EPUB, MOBIの作成
 - □ CALIBREで変換
 - □ 本を追加
 - doc

- html
- □ 書誌を作成
 - タイトル、作成者、作成日等をセット
- □ 本を変換
 - epub
 - mobi
- □ SlideShareへのアップロード
 - Facebookでログイン
 - Linkedinでログイン
- □ ドローツール
 - Bomboo Paper
 - サブトピック 2
- □ スクリーンショットのPDF化
 - □ PDF化
 - スクリーンショットを撮る→PNG化
 - 複数のPNGをAcrobtで結合しPDFファイル作成
 - PDFファイルをAcrobatでテキスト認識
 - □ 校正
 - PDFファイルをMetaMojiに転送しコメントを付与
- □ Webサーバで公開
 - Webサーバ立ち上げ
 - Webサーバにアップロード
- □ プログラミング
 - □ Python
 - □ Pythonインストール
 - 3.7.xはC Libraryのアップデートがなくてもインストールできる
 - Pathの設定にチェックを入れる
 - インストール場所はD:¥Programs¥Python¥Python37-32
 - □ whlインストール
 - https://pypi.org/simple/XXXX/から最新のwhlをダウンロード
 - ダウンロードしたファイルをD:¥Programs¥whlに移動
 - d:¥
 - cd ¥Programs¥whl
 - pip install XXXX.whl
 - ☐ 7-ZipPortable
 - D:¥7-ZipPortableにインストール
 - ☐ WinPython (STICK)
 - □ インストール
 - WinPython32-3.6.5.0Qt5ダウンロード
 - D:¥WinPython¥にインストール
 - □ 使えるアプリ
 - IDLEX (Python GUI)
 - IPython Qt Console
 - ☐ Jupyter Notebook
 - 対話型でPYTHONを実行できるようにしたツール

- IDELX、IPYTHON、ファイルエクスプローラ、端末エミュレータ等を1つの画面で使えるようにしたツール
- WinPython Control Panel
- □ モジュール
 - pip install xxxxx
 - D:¥WinPython¥python-3.6.5¥Lib¥site-packages にインストールされる
- ☐ Google Colaboratory
 - https://colab.research.google.com/
 - □ 設定
 - Googleログイン
 - GitHub URLセット
 - ノート(.ipynb)作成
 - □ 保存
 - GitHubに保存
- ☐ GitHub
 - □ ユーザ登録
 - ユーザ名, メールアドレス, パスワード
 - □ Repository
 - https://github.com/BlueMoon55/Python_Test
 - □ htmlファイルプレビュー
 - https://htmlpreview.github.io/?【htmlファイルURL】
- ☐ Portable Git
 - □ 試しインストール
 - PortableGit-2.18.0-32-bit.7z をダウンロード
 - □ スティックPC
 - W:¥PortableGit
 - SSH鍵生成
 - ☐ **1**4T09779 (Portable Git)
 - □ 「SSL certificate problem: self signed certificate」と表示された時の対処法
 - git config --global http.sslVerify false
 - □ 【参考】Pythonモジュールでのエラー回避例
 - requests.get('http://news.yahoo.co.jp/pickup/science/rss.xml', verify=False)
 - □ リモートリポジトリの変更をローカルに反映
 - git-bashの起動
 - cd /d/git_repository/Sharing_Knowledge
 - git status
 - git pull https://github.com/BlueMoon55/Sharing_Knowledge master
 - □ ローカルリポジトリの変更をリモートに反映
 - git-bashの起動
 - cd /d/git_repository/Sharing_Knowledge
 - git status
 - git add --all
 - もしくは、git commit -a -m "Rev.x.x
 - git commit -m "Rev.x.x"
 - git push https://github.com/BlueMoon55/Sharing_Knowledge master

- □ GitHubで新しいリポジトリを作成
 - TCYSS_Repository
- □ ローカルリポジトリをGitリポジトリに変換する
 - cd D:\forall git_repository\forall TCYSS_Repository
 - git init
- □ ローカルリポジトリのファイルをインデックスに追加、コミットする
 - git add .
 - git commit -m "FirstCommit"
- □ GitHubのリモートリポジトリ情報を追加する
 - git remote add origin https://github.com/BlueMoon55/TCYSS_Repository.git
- □ ローカルリポジトリをリモートリポジトリへ反映させる
 - git push origin master
- □ github.ioユーザサイトの作成→プロジェクトサイトを作成しているため現在作成
 - ☐ GitHub
 - □ GitHub Pages用のリポジトリの作成
 - GitHubでBlueMoon55.github.ioというリポジトリを作成
 - □ git
 - cd d:\fulletgit_repository
 - git clone https://github.com/BlueMoon55/BlueMoon55.github.io
 - cd d:\forall git_repository
 - echo "Hello world" > index.html
 - git add --all
 - git commit -m "Initial commit"
 - git push -u origin master
 - □ Browser
 - https://bluemoon55.github.io/
- □ プロジェクトサイトの作成
 - ∃ git
 - cd d:\fulletgit_repository
 - git branch gh-pages
 - git checkout gh-pages
 - git push -u origin gh-pages
 - ☐ Browser
 - add
- □ gh-pagesとmasterの同一化
 - git checkout master
 - git merge gh-pages
 - git commit -m "XXXXmerge"
 - git push https://github.com/BlueMoon55/Sharing_Knowledge master
 - git checkout gh-pages
- □ master ブランチの削除
 - git branch --delete master
 - git push --delete origin master
- □【参考】
 - □ Git Bashで操作コマンド 🗾

- git init
- git status
- git add [ファイル名]
- git commit -m "projectX version 1.0"
- git log
- git branch [ブランチ名]
- git checkout [ブランチ名]
- git push リモートリポジトリ名 リモートブランチ名
- git clone リモートリポジトリのurl
- git pull [remote repository PATH] [branch]
- git fetch [remote repository]
- git branch --all
- git push --delete origin リモートブランチ名
- git branch --delete ブランチ名
- ☐ PortableGit GUI
 - ☐ git gui
 - cd git_repositry
 - sh git_gui.sh
- ☐ Git
 - □ インストール (VAIO)
 - (https://git-scm.com/downloads)
 - ProgramFilesにインストール
 - □ インストール (スティックPC)
 - (https://git-scm.com/downloads)
 - ProgramFilesにインストール
 - □ Repository生成
 - ¥git_repository¥Python_Test
 - □ Git Bashで操作 🗾
 - git init
 - git status
 - git add [ファイル名]
 - git commit -m "projectX version 1.0"
 - git log
 - git branch [ブランチ名]
 - git checkout [ブランチ名]
 - git push リモートリポジトリ名 リモートブランチ名
 - git clone リモートリポジトリのurl
 - git pull [remote repository PATH] [branch]
 - git fetch [remote repository]
 - git gui
- □ Webサーバ
 - ☐ Flask
 - Winpythonにはインストール済
 - PyMySQL
- □ Jekyll で Web サイト作成 🗾
 - □ 参考サイト
 - GitHub Pages の Jekyll で Web サイト公開 🗾

- WindowsでのJekyllの使い方 (その1) 🗾 ■ GitHubPages×Jekyllでこのサイト作った話 🗾 ■ github pageとJekyllでブログ作成 🔼 ■ Github Pagesからリダイレクト 🗾 ☐ GitHub Pages Dependency versions https://pages.github.com/versions/
- □ Rubyインストール
 - RubyInstaller for WIndows
- □ github-pages インストール
 - gem install github-pages
 - gem install bundler
- □ Jekyll プロジェクト生成
 - cd d:¥git_Repository
 - jekyii new jekyll_site
 - cd jekyll_site
 - rm Gemfile Gemfile.lock
- □ Jekyll稼働確認
 - jekyll s
 - http://http://localhost:4000/
- □ Python_TestでWeb公開
 - jekyll_siteディレクトリの内容をPython_Testにコピー
 - gitコマンドでリモートにpush
- □ Jekyll Architectテーマサイト作成
 - □ コピーサイト作成
 - architect-master.zip ダウンロード
 - git_Repositoryフォルダーに解凍
 - 既存ローカルリポジトリにオーバーライト
 - git push
 - □ 初期設定及び適用
 - cd script
 - sh bootstrap
 - cd ..
 - ☐ bundle exec jekyll serve --watch
 - jekyll 3.7.3 | Error: incompatible encoding regexp match (Windows-31J regexp with UTF-8 string)
 - http://127.0.0.1:4000
 - □ カスタマイズ
 - ☐ _config.yml
 - encoding: UTF-8
- GitHub Desktop
- □ atom
 - □ メニューバー再表示
 - Ctrl + Shift + P
 - Window:Toggle Menu Bar を選択
 - □ メニュー日本語化
 - □ sslエラー

- □ %USERPROFILE%¥.atom¥内に.apmrcファイル
 - strict-ssl = false
 - 解決せず
- ☐ apm config rm https-proxy
 - 解決
- パッケージインストール(1)
- □ パッケージインストール(2)
 - https://github.com/syon/atom-japanese-menu 🗾
 - zipダウンロード、解凍
 - %USERPROFILE%¥.atom¥packages にコピー
- □ パッケージインストール(3)
 - □ apmコマンドでインストール
 - apm install atom-file-icons
- ☐ atom-runner
 - Atom上でPythonコードが実行できる
 - Pythonコードの実行: Alt+R
 - Pythonコードの停止: Ctrl+Shift+C
 - Pythonコードの実行ウインドウを閉じる: Esc
- ☐ ¥.atom¥packages
 - advanced-open-file
 - atom-beautify
 - atom-clock
 - atom-file-icons
 - atom-japanese-menu-master
 - atom-runner
 - atom-runner-master
 - atom-space-pen-views-master
 - autocomplete-paths
 - autocomplete-python
 - git-plus
 - ☐ highlight-column
 - **■** ×
 - language-markdown
 - merge-conflicts
 - pdf-view
 - remote-edit
 - script
 - tag
 - web-search
- 🛘 全文検索システム『ひまわり』 💆
 - ダウンロード 🗾
 - 利用者マニュアル 🗾
 - □ Mecabとの連携設定
 - $\ \ \boxdot \ \ C: \verb| Himawari| \verb| A.himawari| annotator_config.xml|$
 - C:¥MeCab¥bin¥mecab
 - □ インポート
 - 2018年(30FY)保存版¥2018_Conversion_products¥2018年度自動生成_Cyber

■ ダウンロードした jar ファイルは C:¥SikuliX フォルダへ配置

■ 機械学習やプログラミングのスキルなしでも予測分析が実行できるWindows のGUI ソフトウェア

■ ① Visual Studio 2019 の Microsoft Visual C++ 再頒布可能パッケージ

■ java -jar sikulixsetup-1.1.3.jar

■ Windows 10 (64 ビット版)

□ SikulixIDE の起動
■ ./runsikulix

☐ Prediction One

□ 動作環境

Expand - Collapse

□ インストール

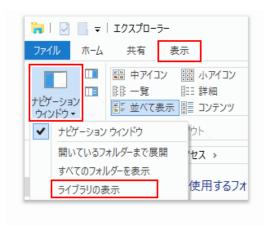
- □ ダウンロード
 - https://predictionone.sony.biz/download2/
 - Setup_PredOne_x64_1.2.0.09020.zip
- □ インストール
 - Zip 解凍
 - Setup_PredOne_x64_1.2.0.09020.msi
 - インストーラ起動
- □ ファイル配置
 - □ 実行ファイル (GUI)
 - C:¥Program Files¥Sony¥Prediction One¥PredictionOne.exe
 - □ 実行ファイル (コマンドライン)
 - C:¥Program Files¥Sony¥Prediction One¥PredictionOneCmd.exe
 - □ マニュアル
 - C:\Program Files\Sony\Prediction One\ja-JP\doc\Prediction_One_Document.pdf
 - □ チュートリアル
 - C:\Program Files\Sony\Prediction One\ja-JP\doc\sample_dataset\use_case\
 - □ ワークスペース
 - C:¥Users¥[ユーザー名]¥AppData¥Local¥Sony Corporation¥Prediction One¥WorkSpace
- □ 実行

- □ システム
 - ☐ Power Shell
 - コマンドプロンプトからPowerShellに乗り換えるための小さな本 🗾
- □ etc
 - 🛭 文書に含まれる制御文字の検出 🗾
 - cat *.txt | grep \$'¥cK' | more
 - □ Python SSLエラー
 - requests.get('https://対象URL',verify=False)
- ☐ SurfaceGo
 - 更新日: 2019年2月19日
 - □ 設定
 - □ システム
 - □ システム情報
 - デバイスマネージャ
 - リモートの設定
 - システムの保護
 - □ システムの詳細設定
 - □ 詳細設定タブ
 - □ 環境変数
 - UPath等の設定変更
 - □ ディスプレイ

- □ 明るさと色
 - 自動調整
- □ 夜間モード
 - OFF
- □ 拡大縮小とレイアウト
 - □ テキストサイズ
 - **150%**
 - □ 表示スケール
 - □ 解像度
 - 1800×1200
 - □ マルチディスプレイ
 - MiraCastで接続する
 - □ HDMIで接続する
 - 拡張モードで表示できない時は、リフレッシュレートを48Hzに変更してみる
 - □ ディスプレイの詳細設定
 - □ Surface Display
 - 1800×1200
 - 59Hz
 - 8ビット
 - 標準ダイナミックレンジ(SDR)
 - □ プロパティ
 - □ アダプタ
 - ☐ Intel HD Graphics 615
 - □ 全般
 - ディスプレイアダプター
 - PCI バス0, デバイス2,機能0
 - □ ドライバー
 - **2**018/08/15
 - **2**4.20.100.6287
 - Intel HD Graphics Family
 - Internal
 - Intel Video BIOS
 - 4177MB
 - 128MB
 - 0MB
 - 4049MB
 - □ モード
 - 1800×1200, True Color(32ビット),60ヘルツ
 - □ モニター
 - Surface Display
 - 60ヘルツ
 - □ 色の管理
 - 特になし
 - □ グラフィックの設定
 - クラシックアプリ

- □ 通知とアクション
- □ 電源とスリープ
 - ■画面
 - スリープ
 - ネットワーク接続
- □ バッテリ
 - バッテリー残量に影響を及ぼしtげいるアプリを表示する
- □ ストレージ
 - ローカルストレージ
 - □ ストレージセンサー
 - □ 一時ファイルの自動削除
 - OFF
 - □ その他のストレージ設定
 - 新しいコンテンツの保存先を変更する
 - アプリ
 - 映画
 - オフラインマップ
- このPCへのプロジェクション
- 共通エクスペリエンス
- 日 システムとセキュリティ
 - □ バックアップ
 - □ バックアップと復元(Windows7)
 - ☐ HDCL-UT(E:)
 - ☐ System Volume Information
 - サブトピック 1
 - ☐ WindowsImageBackup
 - □ UDESKTOP-GFLB15S
 - 更新日: 2019年2月17日
 - □ UDuo-RN
 - 更新日: 2019年1月1日
 - ☐ HDCL-UT(F:)
 - ☐ STICKPC-RR16KRN
 - 更新日: 2018年9月2日
 - ☐ System Volume Information
 - **?**
 - □ VAIO-FIT14A
 - 更新日: 2018年10月22日
 - ☐ WindowsImageBackup
 - □ UDESKTOP-GFLB15S
 - 更新日: 2019年2月18日
 - 日 ネットワークとインターネット
 - □ 携帯電話

- □ 詳細オプション
 - □ 従量制課金接続
 - ON
 - □ APN設定
 - □ プロファイル名: docomo_SP
 - APN: spmode.ne.jp
- □ 携帯データネットワークを使えるアプリ
 - 使用許可: ON
 - アプリ選択
- □ Wi-Fi
 - □ 利用できるネットワークの表示
 - □ 回線毎
 - □ 従量制課金接続
 - ON/OFF
- □ データ使用状況
 - アプリごとの使用状況の表示
 - □ 携帯電話、WiFi回線毎
 - データ通信量の上限
 - バックグラウンドデータ
- モバイルホットスポット
- □ ファイル履歴
 - ☐ 【?】RecoveryFolder (¥¥wdcloud)
 - DESKTOP-GFLB15S
 - ☐ TimeMachineBackup (¥¥wdcloud)
 - - DESKTOP-GFLB15S
- 管理ツール
- □ デバイス
 - □ ペンとWindows Ink
 - 右利き
 - □ 視覚効果を表示する
 - OFF
 - カーソルを表示する
 - □ ペンの使用中はタッチ入力を無視
 - OFF
- □ ライブラリ
 - □ ライブラリの表示 🗾



ライブラリがエクスプローラーに表示されます。

☐ System

- Mouse without Borders
- - Download先変更
- □ UneDrive
 - Personal
 - Business
 - □ 設定
 - ファイルオンデマンド
- ☐ U Google Chrome
 - Default設定
- □ 🙂 メール
 - live.jp
- ☐ Programming
 - On Anaconda 3-2018.12-Windows-x86_64
 - ☐ U Git-2.20.1-64-bit
 - すべてDefault?
 - □ USCodeUserSetup-x64
 - ワークスペース
 - □ 機能拡張
 - Japanese
 - Python
 - **U** GitHubDesktop
- $\ \Box$ OA
 - □ **U** Adobe
 - Creative_Cloud
 - 各種アプリ
 - □ **U** Xmind8
 - ライセンス
 - ☐ U Freemind
 - Java SE
 - - ID/PWD
 - □ U Office365 Solo

■ • hm887_signed ☐ Net Service ☐ U Kindle_for_PC_Windows ■ 再度□グイン ■ Wunder for Gmail ■ **じ** Twitterアプリ **⊕** Facebookアプリ ☐ Printer,Scanner ☐ Capture on Touch V4 ■ plug-in-for-yms-v13forwin Driver □ UDRC225W Network Monitor ■ 各種アプリ □ Media ■ UPC TV Plus SeeQvaultPlayer DuetSetup ■ EZCastPro_Win ■ Windows Service for LINUX (WSL) <a>I ☐ Calibre https://calibre-ebook.com/ □ 14T09779 ⊟С □ 2018年 (30FY) 保存版 □ 2018年学習データ □ 2018年Python □ Python逆引き □ chap09 □ 09_06 frequency_table atom ∃D ■ git_repository ■ 7-ZipPortable ■ aria2 ■ Calibre Portable

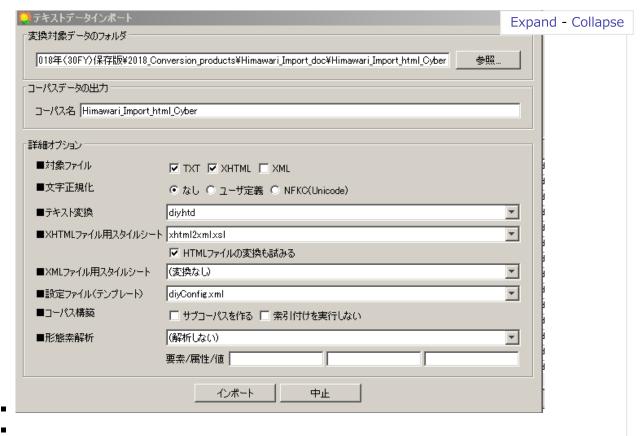
ChainLP40b17
FreemindPortable

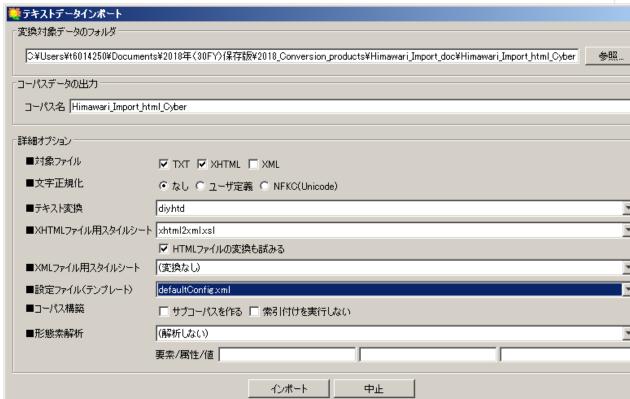
■ Hidemaru

https://account.microsoft.com/services/office/overview

- PortableGit→移行
- PortableGit GUI→移行
- ☐ Programs
 - ☐ Python
 - □ Python37-32
 - D:¥Programs¥Python¥Python37-32¥Scripts¥pip.exe
 - ☐ C:¥Users¥t6014250¥pip¥pip.ini
 - [global]
 trusted-host = pypi.python.org
 pypi.org
 files.pythonhosted.org
 - whl
 - k2pdfopt.exe
- Ralpha
- □ WinPython
 - IDLEX (Python GUI)
 - IPython Qt Console
 - ☐ Jupyter Notebook
 - D:¥WinPython¥notebooks
 - http://localhost:8888/tree#
 - Spyder
 - WinPython Command Prompt
 - ☐ WinPython Control Panel
 - パッケージ管理
 - WinPython Interpreter
 - WinPython Powershell Prompt
- XMind 8 Update 4
- □ VSCodeをポータブル版
 - 【参照】VSCodeをポータブル版として使用するための設定例 🗾
 - □ VSCodeをポータブル版を作成
 - □ VSCodeのzip版をダウンロード
 - https://code.visualstudio.com/Download
 - □ VSCodeのzip版を適当なディレクトリに解凍
 - "D:¥VSCode" へ解凍
 - □ VSCodeの解凍ディレクトリにデータディレクトリを作成
 - "D:¥VSCode¥data" を作成
 - □ VSCodeの解凍ディレクトリに拡張機能ディレクトリを作成
 - "D:¥VSCode¥data¥extensions" を作成
 - □ VSCodeのポータブル版を簡単に起動
 - - @echo off
 - for /f "usebackq tokens=*" %%i in (`cd`) do @set ROOT_DIR=%%i
 - rem echo %ROOT_DIR%
 - %ROOT_DIR%¥Code.exe --user-data-dir "%ROOT_DIR%¥data" --extensions-dir "%ROOT_DIR%¥data¥extensions" %*

- ms-ceintl.vscode-language-pack-ja-1.26.3
- ms-python.python-2018.7.1
- □ git環境設定
 - □ パス追加
 - D:¥PortableGit¥cmd¥
 - > git rev-parse --show-toplevel
 - fatal: not a git repository (or any of the parent directories): .git
- □ 有用な機能拡張
 - Local History
- □ VSCodeのアップデート
 - □ 現在インストールされている"D:¥VSCode"内のdataディレクトリをdata.zipとしてアーカイブ
 - extensionsディレクトリ
 - user-dataディレクトリ
 - "D:¥VSCode"を "D:¥VSCode.old"等にリネーム してバックアップ
 - □ VSCodeのzip版をダウンロード
 - https://code.visualstudio.com/Download
 - VSCodeのzip版を "D:¥VSCode" に解凍
 - data.zipを "D:¥VSCode"に移動して解凍する
 - VSCode.bat及びVSCode.vbsを "D:¥VSCode" に移動する
- □ Makeコマンド
 - Make for Windows のインストール 🗾
- □ himawari-mecabポータブル
 - himawariをインストール
 - rsource内にmecabをインストール
 - Path、環境変数をインストール場所に変更
 - Configをインストール場所に変更
- □ mecab
 - ☐ Mecab 0.996
 - d:¥mecab
 - PathにD:¥MeCab¥bin¥を追加
 - 環境変数mecabrcにD:¥MeCab¥etc¥mecabrcをセット
 - pip install mecab-python-windows
 - ☐ pip install negima
 - エラー
- □ 全文検索システム『ひまわり』 🗾
 - ダウンロード 🗾
 - 利用者マニュアル 💆
 - □ Mecabとの連携設定
 - ☐ D:¥Himawari¥.himawari_annotator_config.xml
 - D:¥MeCab¥bin¥mecab
 - □ インポート
 - C:¥Users¥t6014250¥Documents¥2018年(30FY)保存版¥2018_Conversion_products¥2018年度自動生成_Cyber
 - □ html文書取り込み





□ 形態素分析

□ アノテーション

■ MeCab(IPADIC)

□ インストール用ファイル

- calibre-portable-installer-3.20.0
- ChainLP40b17
- PortableGit-2.18.0-32-bit.7z
- xmind-8-update4-windows

 $\ \ \Box$ Google Colaboratory

■ GitHubとリンク

1

Expand - Collapse