

WeVel

Version.1.0

操作マニュアル

合同会社 ハイルディングシステム

2019/03/11

<http://hyldingsystem.co.jp/>

目次

1. はじめに.....	1
1-1. WeVel (ウェヴェル) について.....	1
1-2. WeVel の利用シーン.....	1
1-3. 動作環境.....	2
1-4. ライセンスおよび免責事項.....	2
1-5. 連絡先について.....	2
2. インストールとアンインストールについて.....	3
2-1. インストール.....	3
2-2. アンインストール.....	4
3. WeVel の強み！.....	5
3-1. 直ぐに利用できる！.....	5
3-2. 日本語による簡単スクリプト！.....	5
3-3. IE 互換の組み込みブラウザを採用！.....	5
3-4. バックグラウンドで実行！.....	6
4. WeVel を使ってみる.....	7
4-1. 事前準備.....	7
4-2. バッチファイルで実行する.....	9
4-3. コマンドプロンプトで実行する.....	11
5. スクリプトを作成する.....	12
5-1. URL を表示する.....	12
5-2. 画面をキャプチャする.....	13
5-3. クリックする.....	15
5-4. 入力する.....	16
5-5. 属性を設定する.....	17
5-6. 属性を追加する.....	18
5-7. 待機する.....	19
5-8. 値を取得して利用する.....	20
5-9. 値を設定して利用する.....	22
5-10. 一連の処理を再利用する.....	23
5-11. 値を判定する.....	24
5-12. タグの要素を繰り返して処理する.....	26
5-13. 行データをリストに設定する.....	27
5-14. リストの値を繰り返して処理する.....	29
6. WeVel でサポートされていない機能.....	30

1. はじめに

1-1. WeVel (ウェヴェル) について

WeVel は、IE 互換組み込みブラウザの自動操作をツールです。テキストファイルに簡単なスクリプト (手順) を記述するだけで、任意の URL へアクセスして画面のキャプチャを行うことや、テキストボックスに値を入力して送信ボタンをクリックするといった操作を自動で行うことが可能です。

1-2. WeVel の利用シーン

WeVel は、以下のシチュエーションでのご利用を想定しております。

1-2-1. システム開発での自動テスト

Web の UI テスト自動化と言えば「Selenium」が有名ですが、実行環境を整えるだけでも手間がかかり、スクリプト作成には Java の知識が必要となります。システム開発で既存部分を修正した際に、自動化されたリグレッションテストを実行することで、デグレートを防ぐ効果があることは分かりつつも、必要となるコストの高さや、作成した大量のスクリプトをメンテナンスできない現実を考えるとなかなか導入に踏み切れません。

これらのハードルを下げるべく WeVel は開発されました。Selenium のような細やかな制御は行えませんが、必要最低限の機能を簡単なスクリプトで記述できるようにしております。WeVel 本体とスクリプトが記載されたテキストファイルの 2 つがあれば、Web の UI テスト自動化が行える手軽さを備えています。

1-2-2. 定型作業の自動化

株式投資を行っている人は、「株価の状況が分かる Web サイトへアクセスして、自分の条件に一致した割安と思われる銘柄を購入候補にする」といったことを日常的に行っています。日々変動する 3600 を超える上場企業の株式情報を 1 つずつ精査することは困難を極めます。最終的な判断は自分が行うにしても、「自分の基準に一致する銘柄を探す」という定型作業は自動で行いたいものです。

こういった定型作業を自動化するべく WeVel は開発されました。Web ページの値を取得して比較するといった操作が可能であるため、条件に一致した情報だけを画面キャプチャとして出力することで意思決定をサポートすることができます。

1-3. 動作環境

OS	Microsoft Windows7/10 の動作する環境
メモリ	1.0GB 以上
HDD	出力する画面キャプチャの画像が格納できる程度

1-4. ライセンスおよび免責事項

・本ソフトウェアを利用する際は年間ライセンスの購入が必要となります。

ただし、試用期間を 180 日設けておりますので、試用期間中は自由にご利用いただけます。

※年間ライセンスについては以下を参照ください。

<http://hyldingsystem.co.jp/製品/wevel 各プラン料金/>

・本ソフトウェアは安全に配慮して作成しておりますが、本ソフトウェア利用に伴う逸失利益、逸失売上もしくはデータの紛失、金銭的損失、または間接損害、特別損害、結果損害もしくは懲罰的損害について責任を負いません。

1-5. 連絡先について

本ソフトウェアについてのご質問などについては、以下のメールアドレスまでお問い合わせください。

support@hyldingsystem.co.jp

2. インストールとアンインストールについて

2-1. インストール

WeVel の Zip ファイルを解凍すると、以下のような各ファイルが存在します。

名前	更新日時	種類	サイズ
out	2018/09/21 22:34	ファイルフォルダー	
sample	2018/09/21 22:14	ファイルフォルダー	
WeVel.exe	2018/09/19 23:53	アプリケーション	41 KB
WeVel_run.bat	2018/09/21 22:14	Windows バッチ ファ	1 KB
ブラウザ設定削除.reg	2018/09/21 22:27	登録エントリ	1 KB
ブラウザ設定追加.reg	2018/09/21 22:34	登録エントリ	1 KB

WeVel.exe	WeVel のプログラム本体
WeVel_run.bat	WeVel はコマンドラインにて起動するアプリケーションです。インストール後にすぐにサンプルを実行できるようにバッチファイルを同梱しています。
ブラウザ設定追加.reg	WeVel は IE 互換ブラウザを内包しております。デフォルトでは IE7 相当で画面描画を行うので、事前に本ファイルを実行して、最新ブラウザである Edge 相当で画面描画を行うように変更してください。 ※本ファイルを実行することでレジストリが変更されます。
ブラウザ設定削除.reg	ブラウザ設定追加.reg により追加されたレジストリを削除します。
out	WeVel_run.bat を実行した際に、キャプチャした画像ファイルの格納先です。
sample	WeVel で行えることのサンプルを格納しております。スクリプトを作成する際の参考にしてください。

2-2. アンインストール

「ブラウザ設定追加.reg」にてレジストリを変更された場合は、「ブラウザ設定削除.reg」を実行した後に、WeVel のフォルダを削除してください。

3. WeVel の強み！

ブラウザの自動操作には、Selenium や JavaScript などいくつかの方法が存在します。そんな中で WeVel の強みについて解説します。

3-1. 直ぐに利用できる！

Selenium でブラウザの自動操作を行おうとすると、開発環境である Eclipse、Selenium、ドライバーなどをダウンロードして、動作するように環境構築を行う必要があります。これが意外と手間がかかり、何かにつまずくと悪戯に時間を浪費することになります。自分の環境だけを構築するなら良いのですが、チーム全員が環境構築するとそれだけで相当な工数を使ってしまいます。つまり、ブラウザの自動操作を試したいと思っても、準備に工数がかかるため、あまり気軽には試せない状況と言えます。

WeVel は、ファイルを解凍すればすぐに利用できるのも、チームの人数が多くても環境構築に時間を使うことはありません。

3-2. 日本語による簡単スクリプト！

ブラウザを自動的に操作するには、行った操作を記録する方法とスクリプトを記述する方法が考えられます。スクリプトを記載する場合、プログラム言語（Java や JavaScript など）を利用することが多く、技術者でなければスクリプト自体の作成が困難です。例え技術者であっても、大量のテストコードを記載するには工数が必要です。

WeVel は日本語によるスクリプトなので、直感的にスクリプトを記載できます。

3-3. IE 互換の組み込みブラウザを採用！

WeVel は IE 互換の組み込みブラウザを採用しております。クライアント PC で利用する IE と完全一致とはいきませんが、ほぼ同一のイメージで画面をキャプチャすることが可能です。そのため、IE のみを許容する Web サービスなどでもご利用いただけます。

3-4. バックグラウンドで実行！

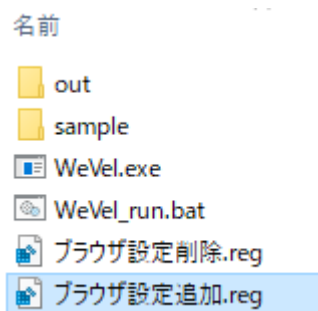
Selenium で見られる自動的に IE が起動して、自動的に操作が行われていく様子を見ているのは楽しいのですが、定型作業などはバックグラウンドでひっそりと行ってもらい、その間に別の仕事をしていたいものです。

WeVel は GUI のないヘッドレスブラウザなので、時間のかかる処理もバックグラウンドでひっそりと行えます。また、Windows サーバを用意すれば、サーバのサービスとして動作させることも可能です。

4. WeVel を使ってみる

4-1. 事前準備

WeVel のブラウザはデフォルトでは IE7 相当で描画されるため、アクセスするサイトによってはブラウザが古い旨の警告が表示されます。そのため、レジストリを変更して Edge 相当に変更する必要があります。



レジストリを変更する「ブラウザ設定追加.reg」をダブルクリックすると、ユーザアカウント制御により「アプリがデバイスに変更を加えることを許可するか」のメッセージが表示されるので、「はい」を選択してレジストリを変更してください。

「ブラウザ設定追加.reg」では、以下の設定が行われます。ブラウザのバージョンを変更したい場合は、「WeVel.exe」=dword:00002af9 という値を変更してから、再度「ブラウザ設定追加.reg」を実行してください。

```

Windows Registry Editor Version 5.00
[
[HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main
// IE11
// "WeVel.exe"=dword:00002af8
[HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main
// IE11:Edgeモード
// "WeVel.exe"=dword:00002af9
[HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main
// IE11
// "WeVel.exe"=dword:00002af8
[HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main
// IE11:Edgeモード
// "WeVel.exe"=dword:00002af9

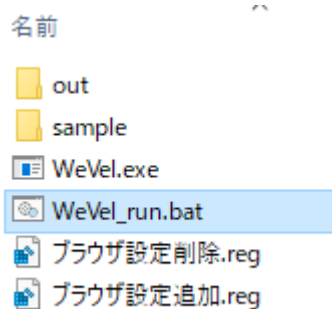
```

各値のバージョンは以下の通りです。

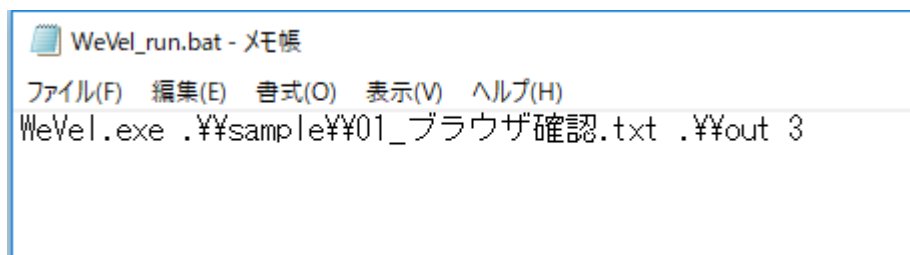
値	バージョン
dword:00002af9	Internet Explorer 11, Edge モード
dword:00002af8	Internet Explorer 11
dword:00002711	Internet Explorer 10, Standards モード
dword:00002710	Internet Explorer 10
dword:0000270f	Internet Explorer 9, Standards モード
dword:00002710	Internet Explorer 9
dword:000022b8	Internet Explorer 8, Standards モード
dword:00001f40	Internet Explorer 8
dword:00001b58	Internet Explorer 7

4-2. バッチファイルで実行する

バッチファイルをダブルクリックすると、WeVel の内部ブラウザのバージョンを確認してキャプチャするサンプルが実行されます。



バッチファイルの内容をテキストエディタで確認すると、以下の内容が記載されています。

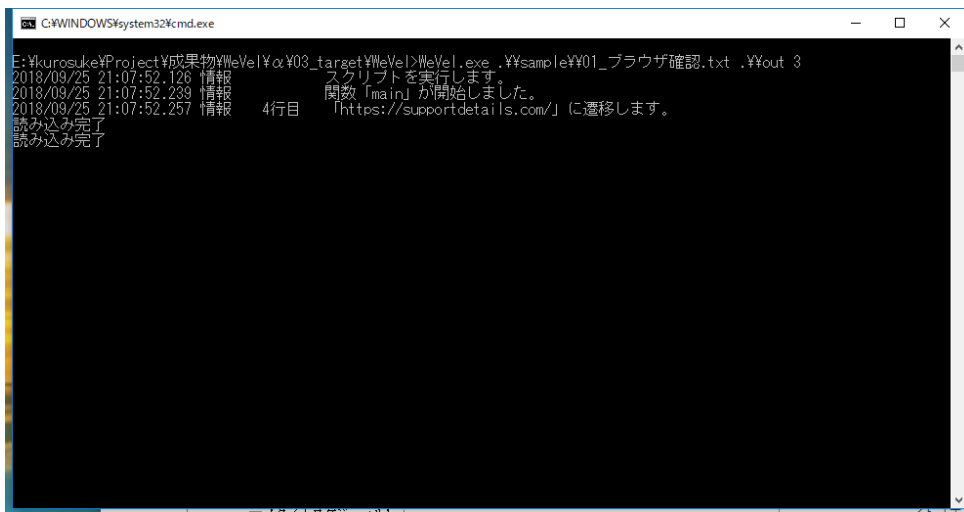


WeVel は第一引数にスクリプトファイルのパス、第二引数に出力先、第三引数に出力レベル（1：エラーレベルを出力、2：警告レベル以上を出力、3：情報レベル以上を出力）を指定します。第二引数に指定された **sample** フォルダを確認すると「01_ブラウザ確認.txt」が格納されていることが分かります。

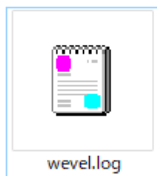
名前	更新日時	種類	サイズ
01_ブラウザ確認.txt	2018/09/21 22:23	TXT ファイル	1 KB

自作のスクリプトを実行したい場合は、第二引数をスクリプトへのファイルパスに変更してからバッチファイルを実行してください。

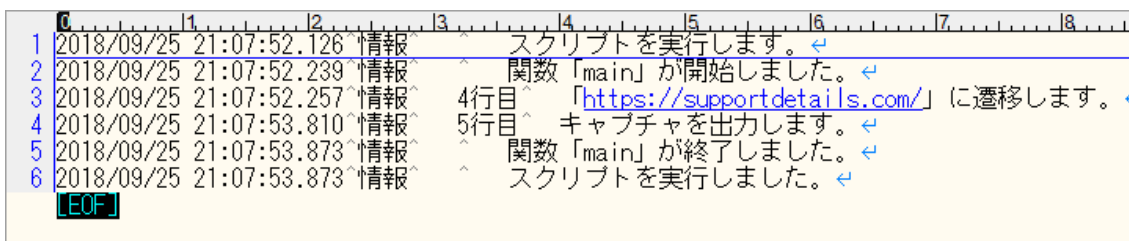
バッチファイルを実行すると、コマンドプロンプトが表示されます。



処理が完了するとコマンドプロンプトが終了し、out フォルダに画像とログが出力されます。

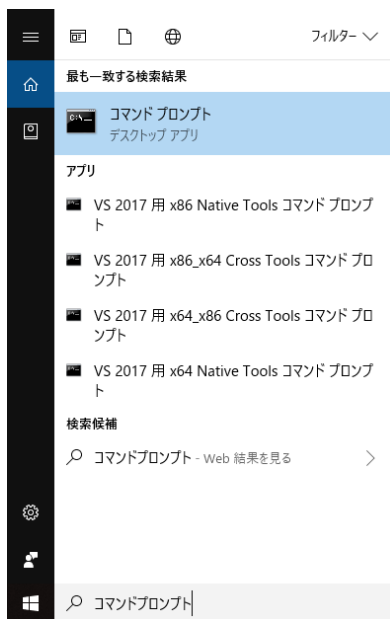


wevel.log には、WeVel の実行結果が出力されます。

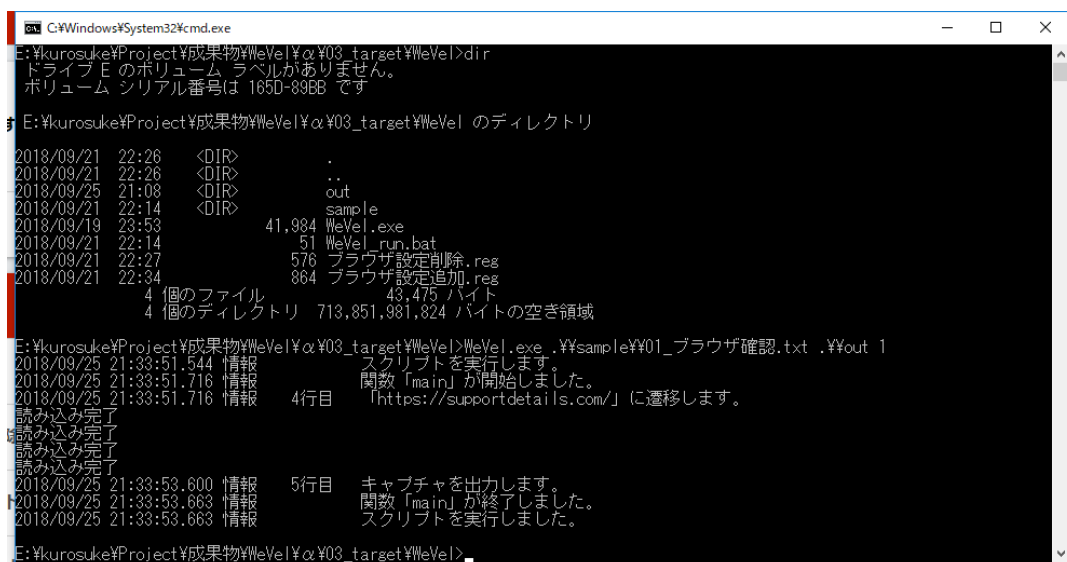


4-3. コマンドプロンプトで実行する

WeVel をコマンドプロンプトから実行する場合、コマンドプロンプトを起動して WeVel.exe が格納されたフォルダまで移動してください。



WeVel.exe を実行すると標準出力に処理内容が表示されます。



WeVel はコマンドラインにて実行できるので、Windows のタスクスケジューラと組み合わせることで、毎日定時に定型作業を自動実行させることができます。

5. スクリプトを作成する

WeVel で利用可能なスクリプトのフォーマットと用途について、サンプルを交えて説明します。スクリプトを作成する場合、文字コードは SJIS、改行コードは CRLF にてテキストファイルを作成してください。

スクリプト内にコメントを記載したい場合は、#を先頭に記載してください。

なお、スクリプト中の改行は許容していないので、スクリプトは 1 行で記載ください。各種サンプルスクリプトは sample フォルダに格納されておりますので、そちらのファイルを参考ください。

5-1. URL を表示する

■概要

内部ブラウザで指定した URL を表示します。

■フォーマット

URL 表示 => URL

項目	必須	設定内容
URL	○	ブラウザに表示したい URL を記載する。

■記載サンプル

URL 表示 => <http://hyldingsystem.co.jp/>

■サンプルスクリプト

02_URL を表示する.txt

5-2. 画面をキャプチャする

■概要

内部ブラウザで表示中の画面をキャプチャして画像ファイルに出力します。

■フォーマット

キャプチャ::ファイル形式 => name=ファイル名,adjust_height=高さ補正,adjust_width=幅補正,fixed_height=高さ固定,fixed_width=幅固定

項目	必須	設定内容
ファイル形式		出力するファイル形式を指定する。bmp、gif、jpeg、pngのいずれかを指定できる。省略した場合はbmpで出力する。
ファイル名		出力するファイル名を指定する。拡張子はファイル形式に従って自動的に付与される。 省略した場合はタイムスタンプでファイル名を自動生成して出力する。
高さ補正		キャプチャ出力時の高さは、ブラウザが自動で判定したサイズで出力されるが、想定と異なる高さで出力された場合に、本パラメータにて高さを調整する。
幅補正		キャプチャ出力時の幅は、ブラウザが自動で判定したサイズで出力されるが、想定と異なる幅で出力された場合に、本パラメータにて幅を調整する。
高さ固定		キャプチャ出力時の高さは、ブラウザが自動で判定したサイズで出力されるが、想定と異なる高さで出力された場合に、本パラメータにて高さを固定する。高さ固定を指定した場合、高さ補正による指定は無視される。
幅固定		キャプチャ出力時の幅は、ブラウザが自動で判定したサイズで出力されるが、想定と異なる幅で出力された場合に、本パラメータにて幅を固定する。幅固定を指定した場合、幅補正による指定は無視される。

■ 記載サンプル

キャプチャ::png => name=03_サイズ固定, fixed_height=3000, fixed_width=1200

■ サンプルスクリプト

03_キャプチャする.txt

5-3. クリックする

■概要

内部ブラウザで表示中画面のボタンやリンクをクリックする。

■フォーマット

クリック::検索条件 => 待機時間 (ミリ秒)

項目	必須	設定内容
検索条件	○	検索条件には ID を指定する方法とタグを指定する方法があります。タグを指定する場合は、属性で絞り込むことが可能です。 複数の要素が一致した場合、\$にてインデックスを指定することが可能。インデックスが未指定の場合は、最初に見つかった要素を対象とする。
待機時間 (ミリ秒)		クリックを行った後にページ読み込みが発生している場合は、読み込みが完了するか指定した待機時間 (ミリ秒) まで処理を行わない。最大で 60 秒まで指定可能で、省略した場合は 1 秒を設定したものとみなす。

■記載サンプル

- ・ ID で指定する場合、先頭に#を付与する。

クリック::#economy

- ・ タグで指定する場合、タグ名と属性を指定する。

複数一致する場合は、\$にてインデックスも指定する。

クリック::a「class=Tab_link bdA5」\$3

■サンプルスクリプト

04_クリックする.txt

5-4. 入力する

■概要

内部ブラウザで表示中画面のテキストボックス、テキストエリア、ラジオボタン、チェックボックス、プルダウンに任意の値を入力する。

■フォーマット

入力::検索条件 => 任意の値

項目	必須	設定内容
検索条件	○	「5-3. クリックする」と同様
任意の値	○	テキストボックス、テキストエリアの場合は、入力したい任意の文字列を指定する。 ラジオボタン、チェックボックス、プルダウンの場合は、入力したい任意の値 (value) を指定する。

■記載サンプル

- ・テキストボックスに入力する。

入力::#text_test => テキストボックスに入力します

- ・ラジオボタンを選択する。

入力::#radio_2 => 2

■サンプルスクリプト

05_入力する.txt

5-5. 属性を設定する

■概要

内部ブラウザで表示中画面の任意の要素に対して、指定した属性を設定します。既に属性値が設定されていた場合は上書きされます。既存の値に追加したい場合は、属性追加を利用してください。

■フォーマット

属性設定::検索条件 => 属性名=属性値, ...

※属性名と属性値はカンマ (,) で複数指定することが可能です。

項目	必須	設定内容
検索条件	○	「5-3. クリックする」と同様
属性名	○	設定したい属性の名前を指定する。
属性値	○	設定したい属性の値を指定する。

■記載サンプル

- ・属性やスタイルを同時に設定する。

属性設定::h2 => align=center, style=color: blue; font-weight: bold

■サンプルスクリプト

06_属性を設定する.txt

5-6. 属性を追加する

■概要

内部ブラウザで表示中画面の任意の要素に対して、指定した属性を追加します。既に属性値が設定されていた場合は半角スペースを付与して値が連結されます。既存の値をクリアしたい場合は、属性設定を利用してください。

■フォーマット

属性追加::検索条件 => 属性名=属性値, ...

※属性名と属性値はカンマ (,) で複数指定することが可能です。

項目	必須	設定内容
検索条件	<input type="radio"/>	「5-3. クリックする」と同様
属性名	<input type="radio"/>	設定したい属性の名前を指定する。
属性値	<input type="radio"/>	追加したい属性の値を指定する。

■記載サンプル

- ・属性を設定後、追加で属性値を加える。

属性設定::h2 => align=center, style=color: blue; font-weight: bold;

属性追加::h2 => style=border-bottom: dotted 2px;

■サンプルスクリプト

07_属性を追加する.txt

5-7. 待機する

■概要

内部ブラウザで表示中画面に指定した条件と一致する要素が出現するまで待機します。ただし、最大待機時間を超過した場合は処理を続行します。

WeVel でリンクやボタンをクリックして画面遷移を行う際に、正しく読み込みが完了できない場合があります。そういった場合に一致するタイトルが表示されるまで待機することで回避できることがあります。

■フォーマット

待機::検索条件 => min=最小待機時間 (ミリ秒) ,max=最大待機時間 (ミリ秒)

項目	必須	設定内容
検索条件	○	「5-3. クリックする」と同様
最小待機時間 (ミリ秒)		検索条件が一致するまで待機する最小の時間を指定する。省略した場合は、100 ミリ秒を設定したものとみなす。
最大待機時間 (ミリ秒)		検索条件が一致するまで待機する最大の時間を指定する。最大で 300 秒まで指定可能で、省略した場合は 5 秒を設定したものとみなす。

■記載サンプル

- ・ 遷移したいページのタイトルが取得できるまで、3~30 秒待機する。

待機::title 「タグの値=合同会社ハイルディングシステム」 => min=3000,max=30000

■サンプルスクリプト

08_待機する.txt

5-8. 値を取得して利用する

■概要

内部ブラウザで表示中画面から指定した条件と一致する値（タグの値/属性値）を取得します。取得した値は、各種スクリプトにて「%{変数名}」という記載を行うことで利用することができます。既に同一の変数名が存在した場合は上書きされます。

■フォーマット

値取得::検索条件 => name=変数名,regex=正規表現,prefix=接頭辞,suffix=接尾辞,targetAttribute=取得対象の属性名

※正規表現にカンマを使用する場合は、「@COMMA@」と記載する。

項目	必須	設定内容
検索条件	○	「5-3. クリックする」と同様
変数名		取得した値を格納する変数名を指定する。未指定の場合、value という変数名で格納する。
正規表現		取得した値に対して、正規表現を利用して最初にマッチングした部分だけを抜き出して、変数に格納する。 未指定の場合は取得した値をそのまま変数に格納する。
接頭辞		取得した値に接頭辞を付与したい場合に指定する。
接尾辞		取得した値に接尾辞を付与したい場合に指定する。
取得対象の属性名		条件に一致したタグの値ではなく、属性値を変数に格納したい場合は、取得したい属性名を指定する。未指定の場合はタグの値を取得する。

■ 記載サンプル

- Web ページから 2 番目に出現する h3 タグの値を取得する。

URL 表示 => <http://hyldingsystem.co.jp/%E8%A3%BD%E5%93%81/>

値取得::h3\$1 => name=h3 の値

- Web ページから 2 番目に出現する h3 タグの値を取得する。

ただし、正規表現で a から z までの文字列が一致する部分を抜き出す。

URL 表示 => <http://hyldingsystem.co.jp/%E8%A3%BD%E5%93%81/>

値取得::h3\$1 => name=h3 の値,regex=[a-z]+

- 取得した値を検索に利用する。

URL 表示 => <https://www.yahoo.co.jp/>

入力::#srchtxt => %{h3 の値}

クリック::#srchbtn

■ サンプルスクリプト

09_値を取得して利用する.txt

5-9. 値を設定して利用する

■概要

指定した変数に任意の値を設定します。設定元の値は変数も指定することができるので、「値取得」にて取得した値に対して四則演算を行うことが可能です。

■フォーマット

値設定 => name=変数名,value1=値 1,value2=値 2,operator=演算子,digits=小数桁数

項目	必須	設定内容
変数名		取得した値を格納する変数名を指定する。未指定の場合、value という変数名で格納する。
値 1		変数に設定する値を指定する。
値 2		値 1 に対して四則演算を行う場合に、対象となる数値を指定する。
演算子		四則演算を行う場合に指定する。+/*が指定可能。
小数桁数		値が数値の場合に四捨五入する小数点以下桁数を指定する。小数点第一位で四捨五入する場合は 0 を指定し、小数点第二位以上で四捨五入する場合は 1 以上を指定する。

■記載サンプル

・Web ページから日経平均株価を取得した後に、0.8 を掛けて小数点第二位で四捨五入する。

値取得::div 「class=fsn」 => name=日経平均,regex=[0-9@COMMA@]+[¥.]?[0-9]+

値設定 => name=計算結果,value1=%{日経平均},value2=0.8,operator=*,digits=1

■サンプルスクリプト

10_値を設定して利用する.txt

5-10. 一連の処理を再利用する

■概要

作成した一連の処理を他の場所でも使用したい場合に、関数として再利用可能な状態にすることができます。関数内で関数を利用することも可能ですが、記述方法によっては無限ループに陥る恐れがあるため、10 階層以上の関数利用を検知した場合には処理を中断します。

■フォーマット

関数::処理区分 => 関数名

項目	必須	設定内容
処理区分	○	処理区分には開始、終了、実行の何れかを指定します。開始は一連の処理の開始位置を表します。終了は一連の処理の終了位置を表します。実行は定義した関数の実行を行います。
関数名	○	変数に設定する値を指定する。 処理区分に開始を指定した場合に、関数名を指定する必要があります。

■記載サンプル

・会社の HP をキャプチャする処理を関数にして、キャプチャ時の幅を任意に指定して実行できるようにする。

値設定 => name=幅,value1=600

関数::実行 => 会社の HP をキャプチャ

関数::開始 => 会社の HP をキャプチャ

URL 表示 => http://hyldingsystem.co.jp/

キャプチャ::png => fixed_width=%{幅}

関数::終了

■サンプルスクリプト

11_一連の処理を再利用する.txt

5-1-1. 値を判定する

■概要

変数の値を判定し、その結果に従って以降の動作を制御します。値が一致した場合に関数にて処理を行うことや、スクリプトの終了や処理中の関数を中断することができます。

■フォーマット

値判定 => name=変数名,operator=演算子,value=判定値,resultName=判定結果名,matchStr=一致時の文言,umMatchStr=不一致時の文言,afterFunction=一致時の処理,afterProcess=一致時の挙動

項目	必須	設定内容
変数名	○	判定する対象となる変数名を指定します。
演算子	○	変数の値と判定値の一致条件を指定します。変数の値が数値の場合は、一致 (eq)、不一致 (ne)、以上 (ge)、以下 (le)、大きい (gt)、未満 (lt) の何れかを指定します。変数の値が文字の場合は、一致 (eq)、不一致 (ne)、正規表現による一致 (regex)、正規表現による不一致 (notRegex) の何れかを指定します。
判定値	○	判定する対象となる判定値を指定します。
判定結果名		指定した変数名に判定結果を変数に格納することができます。ただし、変数名には「@値判定.」という接頭辞が自動的に付与されるので、利用する場合は接頭辞も含めて変数名を指定する必要があります。
一致時の文言		値判定で一致した場合に、判定結果名に設定する文言を指定します。省略した場合は、「一致」という文言を設定します。
不一致時の文言		値判定で不一致の場合に、判定結果名に設定する文言を指定します。省略した場合は、「不一致」という文言を設定します。

項目	必須	設定内容
一致時の処理		値判定で一致した際に、定義してある関数に処理を行わせたい場合に関数名を指定します。関数の処理が完了すると値判定の次の行から処理を再開します。
一致時の挙動		値判定で一致した際に、以降の挙動を指定します。スクリプトを終了する場合は、 exit を指定します。値判定が関数内での処理の場合に、現在の関数の以降の処理を行いたくない場合は、 return を指定します。省略した場合は処理が継続されます。 一致時の処理に関数を指定した場合、関数を実行した後に一致時の挙動に指定された動作を行います。

■記載サンプル

- ・変数「幅」の値が 2000 以上の場合にスクリプトを終了する。

値判定 => name=幅,operator=ge,value=2000,afterProcess=exit

■サンプルスクリプト

12_値を判定する.txt

5-12. タグの要素を繰り返して処理する

■概要

TABLE タグなどで整形されたデータに対して、TR タグなどの単位で任意の関数で処理を行うことができます。関数側では繰り返しタグで指定された要素内のデータのみアクセス可能です。本スクリプトは後述する「リスト値」と「リスト繰り返し」と合わせてご利用ください。

■フォーマット

タグ繰り返し::検索条件 => loopTag=繰り返しタグ,functionNm=関数名

項目	必須	設定内容
検索条件	○	「5-3. クリックする」と同様
繰り返しタグ	○	検索条件にて一致した要素に対して、指定したタグ内のデータを1件ずつ関数に引き渡す。
関数名	○	繰り返して処理させる関数を指定します。

■記載サンプル

- ・特定の表に対して1行ずつ処理を行う。

URL 表示 => <https://minkabu.jp/ranking/stock/21/2/2/0/1>

タグ繰り返し::table 「summary=配当利回り>」 \$0 => loopTag=tr,functionNm=データ取得

■サンプルスクリプト

13_タグの要素を繰り返して処理する.txt

5-13. 行データをリストに設定する

■概要

表にある行ごとのデータを取得して、リスト型の変数に格納して再利用することができます。リスト1件のデータは、「キー = 値」の形式でデータを保持することができます。本スクリプトは後述する「リスト繰り返し」と合わせてご利用ください。

■フォーマット

リスト値::操作 => listName=リスト変数名,keys=キーリスト,values=値リスト,pause=区切り文字

項目	必須	設定内容
操作	○	リスト値への設定方法を指定します。追加を指定した場合、値リストに指定した文字列をリスト変数に格納する。変数追加を指定した場合、値リストに指定した変数名の値を値リストに変数に格納する。削除を指定した場合、値リスト変数を削除する。
リスト変数名	○	値を格納するリストの変数名を指定する。使用する際には「リスト繰り返し」の処理中で「@リスト値。」にキーリストで指定したキーを連結する。
キーリスト	○	リスト内のキーを指定します。複数のキーを指定する場合は、区切り文字で連結します。
値リスト	○	リスト内の値を指定します。複数の値を指定する場合は、区切り文字で連結します。
区切り文字		キーリストおよび値リストを複数指定する際の区切り文字を指定します。

■ 記載サンプル

- ・ 特定の表に対して 1 行ずつ処理を行い、取得したデータをリスト値へ格納する。

関数::開始 => データ取得

配当利回りが 5%未満の場合はデータから除外する。

値取得::td 「class=tar cur vamd」 => name=配当利回り,regex=[0-9]+[¥.]?[0-9]+

値判定 => name=配当利回り,operator=eq,value=,afterProcess=return

値判定 => name=配当利回り,operator=lt,value=5.0,afterProcess=return

銘柄コード、名称、詳細情報へのリンクを取得してリスト値に格納する。

値取得::div 「class=md_sub」 => name=コード

値取得::div 「class=fwb w90p」 => name=名称

値取得::a => name=リンク,targetAttribute=href

リスト値::変数追加 => listName=銘柄リスト,keys=リンク/コード/名称/配当利回り,value=リンク/コード/名称/配当利回り,pause=/

関数::終了

■ サンプルスクリプト

13_タグの要素を繰り返して処理する.txt

5-14. リストの値を繰り返して処理する

■概要

リスト値にて作成したデータに対して、1件ごとに任意の関数で処理を行うことができます。関数側ではリスト名で指定されたデータのみアクセス可能です。

■フォーマット

リスト繰り返し => listNm=リスト変数名,functionNm=関数名

項目	必須	設定内容
リスト変数名	○	繰り返し処理の対象となるリスト変数名を指定する。
関数名	○	繰り返して処理させる関数を指定します。

■記載サンプル

- ・ 特定のリストに対して1行ずつ処理を行う。

リスト繰り返し => listName=銘柄リスト,functionNm=連続キャプチャ

関数::開始 => 連続キャプチャ

URL表示 => %{@リスト値.リンク}

キャプチャ::png => name=%{@リスト値.名称},fixed_width=1200

関数::終了

■サンプルスクリプト

13_タグの要素を繰り返して処理する.txt

6. WeVel でサポートされていない機能

本バージョンの WeVel では以下の機能はサポートされておりません。今後の利用者からのリクエスト状況を見て、技術的に解決可能な事象から対応を進めていく予定です。

- ・ファイルのダウンロード
- ・ファイルのアップロード
- ・ポップアップ画面の操作

変更履歴

バージョン	日付	内容
β	2018/10/08	β 版リリース
1.0	2019/03/11	1.0 版リリース