

字幕時刻の秒ずらしスクリプト

対象となるvttファイルを右クリックして、クイックアクション->「字幕時刻の秒ずらしを行う」を選択->ずらす秒数を入力、マイナスの入力もOK。
注意点としては、以前にtimeadded(時刻情報を文章の先頭に追加する)を行って、字幕文字列に(00:00:20)などの開始時刻が入っているものについては、エラーになります。timeaddedになっているものは、「字幕文章の時刻を削除する」を実行してtimeadded_timeremovedにした後で、この「字幕時刻の秒ずらしを行う」を行うとtimeadded_timeremoved_timeshiftedとなります。その後で、「vttからタイムありvttの作成」を行い、timeadded_timeremoved_timeshifted_timeaddedとなれば修正が完成です。文章の先頭に時刻情報を入れると、動画の尺が変更になった場合、修正に手間が掛かるので、注意が必要です。

便利な使い方としては、
動画に差し込みを行ったり、尺を削除したりすると、字幕の時刻にずれが発生して、その手直しが面倒な訳ですが、
どこの位置に何秒追加した、どこの位置から何秒削除した、などの場合は、影響を受けるその場所以降の字幕だけのvttファイルをコピペで作り、そこに対して、このスクリプトを作動させて時刻をずらす、そうするのが、わかりやすいです。
これらの操作は、使い方が簡単なテキストエディタで行えます。

////////////////////////////////////

Mac Automator

クイックアクション

ワークフローが受け取る項目: ファイルまたはフォルダ 検索対象: 全てのアプリケーション

入力: なし

イメージ: アクション

<1>「AppleScriptを実行」を選択して右側にボックスを追加

```
on run {input, parameters}
```

```
-- 秒数を入力するダイアログを表示
```

```
display dialog "タイムシフトする秒数を入力してください:(マイナス入力も可)" default answer "0"
```

```
set shiftSeconds to text returned of result
```

```
-- 入力された秒数とファイルパスを返す
```

```
set filePath to POSIX path of (item 1 of input)
return {filePath, shiftSeconds}
end run
```

<2>「シェルスクリプトを実行」を選択して<1>の下にボックスを追加
シェル: /bin/bash
入力の引き渡し方法: 引数として

```
#!/bin/bash
```

```
# AppleScriptから受け取った入力
input_file="$1"
shift_seconds="$2"
```

```
# 元のファイルと同じディレクトリに出力ファイルを作成
output_file="${input_file%.vtt}_timeshifted.vtt"
```

```
# タイムスタンプをシフトする関数
```

```
shift_timestamp() {
    local timestamp="$1"
    local shift="$2"
```

```
    # 時間を抽出
```

```
    local hours=${timestamp:0:2}
    local minutes=${timestamp:3:2}
    local seconds=${timestamp:6:2}
    local milliseconds=${timestamp:9:3}
```

```
    # 秒数に変換
```

```
    local total_seconds=$((10#$hours * 3600 + 10#$minutes * 60 +
10#$seconds + shift))
```

```
    # 負の時間を扱う
```

```
    if [ "$total_seconds" -lt 0 ]; then
        total_seconds=0
    fi
```

```
    # 時間に戻す
```

```
    local new_hours=$(printf "%02d" $((total_seconds / 3600)))
    local new_minutes=$(printf "%02d" $(((total_seconds % 3600) / 60)))
    local new_seconds=$(printf "%02d" $((total_seconds % 60)))
```

```

    echo
    "${new_hours}:${new_minutes}:${new_seconds}.${milliseconds}"
}

# ファイル処理
{
    while IFS= read -r line || [ -n "$line" ]; do
        if [[ "$line" =~ (-->) ]]; then
            # タイムスタンプを抽出
            start_time="${line:0:12}"
            end_time="${line:17:12}"

            # タイムスタンプをシフト
            new_start_time=$(shift_timestamp "$start_time"
"$shift_seconds")
            new_end_time=$(shift_timestamp "$end_time" "$shift_seconds")

            # 新しい行を生成
            echo "${new_start_time} --> ${new_end_time}"
        else
            # タイムスタンプ行以外はそのまま出力
            echo "$line"
        fi
    done
} < "$input_file" > "$output_file"

echo "タイムシフトされたファイルが ${output_file} に保存されました。"

```

////////////////////////////////////