

# AppleScript

アップル・スクリプトをメニューバーから使用するには、ユーザ / ホーム / ライブラリ / iTunes フォルダに "Scripts" フォルダを作成し、そこにスクリプトファイルを置きます。

AppleScript のマニュアル（英語）は、スクリプトエディタ.app のメニューバー > ウィンドウ > ライブラリ（shift+コマンド+L）リストから「iTunes」をダブルクリック。

- アップル AppleScript  
<http://www.apple.com/jp/macosx/features/applescript/>
- iTunes AppleScript  
<http://www.apple.com/applescript/itunes/>

- サンプルスクリプト

- [pause](#)
- [play](#)
- [play \(playlist\)](#)
- [playpause](#)
- [stop](#)
- [next track](#)
- [back track](#)
- [previous track](#)
- [fast forward](#)
- [rewind](#)
- [eject](#)
- [update](#)
- [album](#)
- [artist](#)
- [count](#)
- [name \(playlist\)](#)
- [volume](#)
- [rating](#)
- [time](#)

- 実用スクリプト

- [選択した曲の「シャッフル時にスキップ」をチェックする](#)
- [選択した曲の「再生回数」を変更する（数値選択タイプ）](#)
- [選択した曲の「再生回数」を変更する（数値入力タイプ）](#)
- [iTMSのToday's トップ100 ソングを順次再生する](#)
- [歌詞に"..."が含まれるプレイリストを作成](#)
- [複数のファイルのビデオの種類をまとめて変更](#)

## サンプルスクリプト

pause

現在のトラック（が再生中だった場合）を一時停止。

```
tell application "iTunes"  
    pause  
end tell
```

### **play**

現在のトラックを再生。

```
tell application "iTunes"  
    play  
end tell
```

### **play (playlist)**

プレイリストを指定して再生。

```
tell application "iTunes"  
    play playlist "トップ 25"  
end tell
```

### **playpause**

再生中なら一時停止、一時停止中なら再生を実行。

```
tell application "iTunes"  
    playpause  
end tell
```

### **stop**

停止を実行。

```
tell application "iTunes"  
    stop  
end tell
```

### **next track**

次のトラックに移る。

```
tell application "iTunes"  
    next track  
end tell
```

### **back track**

トラックの始めに戻る。トラックの頭数秒以内なら前のトラックに移る。

```
tell application "iTunes"  
    back track  
end tell
```

### **previous track**

前のトラックに移る。

```
tell application "iTunes"  
    previous track  
end tell
```

### **fast forward**

現在のトラックを早送りする。

```
tell application "iTunes"  
    fast forward  
end tell
```

### **rewind**

現在のトラックを逆方向に早送りする（巻き戻す）。

```
tell application "iTunes"  
    rewind  
end tell
```

### **eject**

iPod を取り出す。

```
tell application "iTunes"  
    eject  
end tell
```

### **update**

iPod をアップデートする。

```
tell application "iTunes"  
    update  
end tell
```

### **album**

アルバム名。

現在のトラックのアルバム名を返す。

```
tell application "iTunes"  
    set hoge to album of current track  
    return hoge  
end tell
```

### **artist**

アーティスト名。

現在のトラックのアーティスト名を返す。

```
tell application "iTunes"  
    set hoge to artist of current track  
    return hoge  
end tell
```

## count

カウント。  
現在のトラックの再生回数を返す。  
tell application "iTunes"  
    set hoge to played count of current track  
    return hoge  
end tell

## name (playlist)

名前。  
現在のプレイリスト名を返す。  
tell application "iTunes"  
    set hoge to name of current playlist  
    return hoge  
end tell

## volume

音量。  
0-100で指定。  
tell application "iTunes"  
    set sound volume to 60  
end tell

## rating

マイレート。整数。  
0 = 星x0  
20 = 星x1  
40 = 星x2  
60 = 星x3  
80 = 星x4  
100 = 星x5  
現在のトラックのマイレートを整数で返す。  
tell application "iTunes"  
    set hoge to rating of current track  
    return hoge  
end tell

現在のトラックのマイレートを の数で返す。  
tell application "iTunes"  
    set hoge to rating of current track  
end tell  
if hoge = 0 then  
    set hoshi to " . . . . ."  
end if  
if hoge = 20 then  
    set hoshi to " . . . . ."  
end if  
if hoge = 40 then  
    set hoshi to " . . . . ."

```
end if
if hoge = 60 then
    set hoshi to "    . ."
end if
if hoge = 80 then
    set hoshi to "    ."
end if
if hoge = 100 then
    set hoshi to "    "
end if
```

## time

```
時間, "m:ss"
現在のトラックの合計時間を"m:ss"で返す。
tell application "iTunes"
    set hoge to time of current track
    return hoge
end tell
```

```
現在のトラックの経過時間を秒で返す。
tell application "iTunes"
    set hoge to player position
    return hoge
end tell
```

```
現在のトラックの経過時間を"m:ss"で返す。
tell application "iTunes"
    set hoge to player position
    set min to "0"
    set sec to "00"
```

--秒値が60以上の場合

```
if 59 < hoge then
    set min to hoge div 60
    set sec to hoge mod 60
else
    set min to "0"
    set sec to player position
end if
```

--秒値が9以下の場合

```
if sec < 10 then
    set sec to "0" & sec
end if
```

```
return min & ":" & sec as text
end tell
```

## 実用スクリプト

**選択した曲の「シャッフル時にスキップ」をチェックする**

動作条件

iTunes5.0以降

## ダウンロード

[Shuffleable On.scpt](#)

```
tell application "iTunes"
    repeat with theTrack in selection
        set shuffleable of theTrack to 0
    end repeat
end tell
```

## 選択した曲の「再生回数」を変更する（数値選択タイプ）

### 動作条件

iTunes5.0以降

### ダウンロード

[Played Count Up.scpt](#)

sourceは省略

## 選択した曲の「再生回数」を変更する（数値入力タイプ）

### 動作条件

iTunes5.0以降

```
tell application "iTunes"
```

```
try
```

```
    set trk to {}
```

```
    set slct to a reference to selection
```

```
    if slct as list is {} then return
```

```
    repeat with x in slct
```

```
        set end of trk to (name of x as Unicode text) & (" (" & played count of x & ")")
```

```
    end repeat
```

```
    set AppleScript's text item delimiters to ", "
```

```
    set y to text returned of (display dialog ((trk as Unicode text) & " の再生回数を変
```

```
    if y is false then return
```

```
    set played count of slct to y
```

```
end try
```

```
end tell
```

## iTMSのToday's トップ100 ソングを順次再生する

### 動作条件

iTunes6.0以降

### 注意事項

常駐型のスクリプトです。 .appの方をスクリプトフォルダへ入れてください。

### ダウンロード

[iTMS Top 100 Songs.app](#)

sourceは省略

sourceをダウンロード

[iTMS Top 100 Songs.scpt](#)

## 歌詞に"..."が含まれるプレイリストを作成

```
tell application "iTunes"
display dialog "歌詞の一部を入力してください" default answer "" buttons {"Cancel", "OK"} def
copy the result as list to {text_returned, button_pressed}

set thePlaylist to make new playlist with properties {name:text_returned}
with timeout of 60000 seconds
duplicate (every track of playlist 1 whose lyrics contains text_returned) to thePlaylist
end timeout
end tell
```

## 複数のファイルのビデオの種類をまとめて変更

### 使い方

変更したいビデオファイルを選んでスクリプト実行

### ダウンロード

[Change Video Kind.scp](#)

```
activate
tell application "iTunes"
if selection is not {} then
set videoKindSetting to display dialog "形式を指定してください" buttons {"ムービー", "ミュージックビデオ", "テレビ番組"}
if button returned of videoKindSetting is "ムービー" then
repeat with seltrack in selection
set video kind of seltrack to movie
end repeat
end if
if button returned of videoKindSetting is "ミュージックビデオ" then
repeat with seltrack in selection
set video kind of seltrack to music video
end repeat
end if
if button returned of videoKindSetting is "テレビ番組" then
repeat with seltrack in selection
set video kind of seltrack to TV show
end repeat
end if
else
display dialog "ファイルが選択されていません" buttons {"閉じる"} default button 1 w
end if
end tell
```