

記事

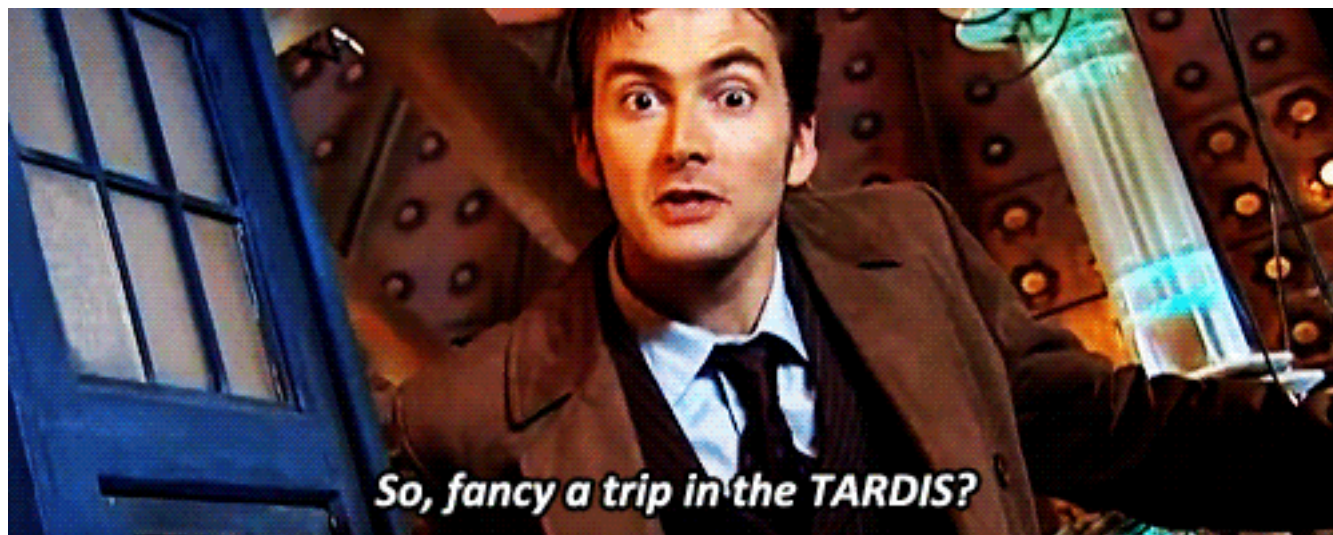
[Toshihiko Minamoto](#) · 2022年11月22日 5m read

時間の支配者になるには - タイムトラベル

タイムトラベルとはバリを訪ねるようなもの。
ガイドを読むだけでなく、そこに足を踏み入れなければならない。
ご飯を食べて、動詞の使い方を間違えたり、2
倍の料金を請求されたり、見ず知らずの人とキスするものさ。

ドクター

これから時空を超えた旅をしましょう。未来と過去の日付を見て、様々なフォーマットで計算する方法を説明します。ターディスは待ちません。コントロールに就いてしっかりつかまりましょう。



明日へ向かおう

[前回の記事](#)で見たように、内部フォーマットの日付は 1841 年 1 月 1 日からの日数と 00:00 時間からの秒数に分けられています。

日にちを増やす場合は、日付の部分に値を追加するかそれから差し引けば...
すでに時間を旅している...と考えられます。

```
SET Today = $ZDATE($HOROLOG,4)
SET InternalDate = $ZDATEH(Today, 4)
SET Future = InternalDate + 2
SET FutureDate = $DATE(Future, 4)
WRITE "Today: "_Today,!, "Internal date: "_InternalDate,!, "Future: "_Future,!, "Future
date: "_FutureDate
```

```
> Today: 02/09/2022
Internal date: 66149
Future: 66151
```

Future date: 04/09/2022

とても簡単です。日にちを移動する方法はわかっていますが、
ただし、次の月に移動するとなると、こうはいきません。
カレンダーには、毎月同じ日数があるわけではないため、日付に 30 日を足しても意味がありません。
時間の加算と減算を行うコマンドをご紹介します。

T-SQL ユーザーであれば、[DATEADD](#) というコマンドに聞きなれていることでしょう。

これは \$SYSTEM.SQL.Functions ライブラリに含まれています。

バック・トゥ・ザ・フューチャー

月数、日数、年数を追加したり、指定された日付から除外したりできます。

```
SET InternalDate = $ZDATEH("09/02/2022", 4)
SET Future = $SYSTEM.SQL.DATEADD("mm",1,InternalDate)
WRITE "InternalDate "_InternalDate,!,"Future: "_Future
```

```
InternalDate: 66149
Future: 2022-03-09 00:00:00
```

日数、月数、年数を除外する場合は、部分ごとにそれぞれ個別に行う必要があります。

```
Set InternalDate = $ZDATEH("09/02/2022", 4)
Set PastDays = $SYSTEM.SQL.DATEADD("dd",12,InternalDate)
Set PastMonths = $SYSTEM.SQL.DATEADD("mm",8,PastDays)
Set PastYears = $SYSTEM.SQL.DATEADD("yy",-7,PastMonths)
WRITE PastYears
```

```
2015-10-21 00:00:00
```

Mc FLY、居るのか～??

今、僕の誕生日... じゃない?

知ろうとしていることが、2 つの日付の差であれば、[DATEDIFF](#) コマンドを使用します。

```
Set internalDate = $ZDATEH("09/02/2022", 4)
Set myBirthday = $ZDATEH("01/02/2023", 4)
Set days = $SYSTEM.SQL.DATEDIFF("dd",internalDate, myBirthday)
WRITE !days," days left until my birthday"
```

```
> 357 days left until my birthday
```

上手くいきましたね... 2 つの日付の差を日数、月数、年数などで見る方法が分かりました。

コロンブスのアメリカ大陸発見に居合わせたい

前にも言ったように、時間フォーマットの値 1 は 01/01/1841 ですが、コロンブスがアメリカ大陸に上陸したのは

1492 年 10 月 12 日でした。そこで、「1 の日付から 1492 年にさかのぼって負の値を付ければいい！」なんて考えるでしょうね。

```
WRITE $ZDATE(-1)
```

```
<VALUE OUT OF RANGE>
```

おっと... ヒューストン、問題が発生しました。負の値を使って 1841 年より前の日付を取得することはできません。
でも、テキストの日付から内部フォーマットへの変換を行って、戻り値を見てみましょう。

```
WRITE $ZDATEH("12/10/1492",4) // European format
```

```
<VALUE OUT OF RANGE>
```

ムムムムム!!! 競ってくれるのはありがたいですが、正しい答えではありません。
どうやって解決できるでしょうか？

これは、日付に、それ以上にならないように最低値が定義されているためで、したがって mindata パラメーターを使用すると、内部フォーマットで負の値を使用できるようになります。

```
WRITE $ZDATEH("12/10/1492",4,,,,,-672045)  
> -127184
```

パラメーターの 7 番目の位置にある値は最低許容値が -672045 となることを示しており、好奇心を満たすために言えば、これは 01/01/0001 に対応する値です。また、ここで使用できる最低値でもあります。

「カンマ」を数えてこのパラメーターを追加する際に混乱してしまった場合は、各国語サポート (NLS) を構成してアクティブのままにするとよいでしょう。

```
SET originalValue= ##class(%SYS.NLS.Format).GetFormatItem("DateMinimum")  
WRITE originalValue,!  
DO ##class(%SYS.NLS.Format).SetFormatItem("DateMinimum", -672045)  
WRITE $ZDATEH("12/10/1492", 4),!  
DO ##class(%SYS.NLS.Format).SetFormatItem("DateMinimum", originalValue) // Return to  
  the minimum value originally established  
  
> 0  
-127184
```

すると、問題なく時間を超えた旅を行えるようになります。

次回の記事: 世界時

謝辞

負の \$Horolog に関する記事を執筆していただいた [@Robert Cemper](#) に感謝します。この記事を書き上げる上で役立ちました。参考記事は次のリンクをご覧ください。

<https://community.intersystems.com/post/date-dec1840-negative-horolog>

[#ObjectScript](#) [#ヒントとコツ](#) [#Caché](#)

ソースURL:

<https://jp.community.intersystems.com/post/%E6%99%82%E9%96%93%E3%81%AE%E6%94%AF%E9%85%8D%E8%80%85%E3%81%AB%E3%81%AA%E3%82%8B%E3%81%AB%E3%81%AF-%E3%82%BF%E3%82%A4%E3%83%A0%E3%83%88%E3%83%A9%E3%83%99%E3%83%AB>