

# CONTENTS

<b>Volume II User's Guide: II-5 Waves #1</b> .....	2
ウェーブの作成 .....	2
ウェーブの名前 .....	3
次元の数.....	5
ウェーブのデータタイプ .....	6
数値データタイプ.....	6
デフォルトのウェーブプロパティ .....	8
Make コマンド .....	9
Make コマンドの例 .....	9
ウェーブと Miscellaneous Settings ダイアログ.....	11
次元とデータスケーリングの変更 .....	12
高度な次元とデータスケーリング .....	12
日付、時刻、日付/時刻の単位.....	13
Duplicate (複製) コマンド .....	15
Duplicate コマンドの例.....	16
ウェーブの Kill (削除) .....	17
KillWaves コマンドの例.....	18
ウェーブのブラウズ.....	19
ウェーブの名前の変更 .....	20
ウェーブの次元の変更 .....	21
ポイントの挿入 .....	22
ポイントの削除 .....	23
本セクションで使用したコマンドの説明 (Volume V Reference) .....	24
Make.....	24
Duplicate.....	26
Redimension .....	28
SetScale.....	30
KillWaves .....	31
Rename .....	32
InsertPoints.....	33
DeletePoints .....	33

# Volume II User's Guide: II-5 Waves #1

---

## ウェーブの作成

Igor Pro マニュアル : II-65 ページをもとに編集

ウェーブは次のようにして作成できます :

- ファイルからデータをロードする
- テーブルに入力する、または貼り付ける
- Make コマンドを使う (ダイアログ経由またはコマンドラインから直接)
- Duplicate コマンドを使う (ダイアログ経由またはコマンドラインから直接)

ほとんどの人は、まずファイルからデータを読み込むことから始めます。

Igor はテキストファイルからデータを読み込むことができます。

この場合、Igor はファイル内のテキストの各列に対してウェーブを作成します。

Igor は他のプログラムで作成されたバイナリファイルやアプリケーション固有のファイルからデータを読み込むこともできます。

ファイルからのデータの読み込みについての詳細は、マニュアル II-126 Importing Data を参照してください。

テーブルにデータを手動で入力することができます。

これは、データ量が少ない場合にのみ、お勧めします。

マニュアル II-239 Using Table to Create New Waves を参照してください。

データを数学的な式で合成するには、まず Make コマンドでウェーブを作成します (本ファイルの末尾かマニュアル V-526 を参照)。

このコマンドは、一時的に使用するウェーブを作成するために、Igor プロシージャ内で頻繁に使用されます。

Duplicate コマンド (本ファイルの末尾かマニュアル V-185 を参照) は、重要で便利なツールです。

多くの組み込み関数では、データをその場で変換します。

したがって、元のデータと変換後のコピーの両方を保存したい場合は、Duplicate を使って元のデータのクローンを作成します。

## ウェーブの名前

Igor のすべてのウェーブには名前が付けられているため、コマンドから参照することができます。また、Igor のダイアログのリストやポップアップメニューからウェーブを選択したり、ウェーブフォームの代入文で参照したりするときにも、ウェーブ名を使います。

ダイアログを使って Make、Duplicate、Rename 操作を行う場合、コマンドラインから直接操作を行う場合、Data Browser を使う場合には、ウェーブ名を選択する必要があります。

Igor 内のすべての名前の大文字と小文字は区別されません。  
wave0 と WAVE0 は同じウェーブを参照しています。

ウェーブ名に使うことができる文字の種類に関するルールは、標準とリベラルという2つのカテゴリに分類されます。

標準およびリベラルな名前の長さは、どちらも 255 バイトに制限されています。

Igor 8.0 より前のバージョンでは、ウェーブ名は 31 バイトに制限されていました。  
長いウェーブ名を使う場合には、ウェーブファイルと Experiment ファイルに Igor Pro 8.0 以降が必要です。

標準名は、アルファベット文字 (A-Z または a-z) で始まり、ASCII アルファベット文字、数字文字とアンダースコア文字のみを含む必要があります。

スペース、ダッシュ、ピリオド、非 ASCII 文字を含むその他の文字は使用できません。

標準名にこのような制限を設けることで、Igor がコマンド (ウェーブフォームの代入文を含む) で標準名を明確に識別できるようになります。

一方、リベラル名には、タブやキャリッジリターンなどの制御文字と、次の4文字を除く任意の文字を含めることができます。

“ ` ` ` ;

標準名は、コマンドや式の中でクオートなしで使うことができますが、リベラル名はクオートで囲む必要があります。

例えば、

```
Make wave0; wave0 = p // wave0 は標準名
Make 'wave 0'; 'wave 0' = p // 'wave 0' はリベラル名
```

Igor は、引用符で囲まれていない限り、コマンド内のリベラル名を明確に識別することができません。

例えば、

```
wave0 = miles/hour
```

において、miles/hour とは、1つのウェーブである場合もあれば、2つのウェーブの割り算である場合もあります。

これらを明確にするには、コマンドまたはウェーブフォームの演算式でリベラル名を使う時には、常にシングルクオートで囲む必要があります。

例えば、

```
wave0 = 'miles/hour'
Display 'run 98', 'run 99'
```

## 注記

リベラル名で動作するプロシージャを記述するには、プログラマーの側で追加の作業とテストが必要になります（マニュアル IV-168 Programming with Liberal Names を参照）。

潜在的な問題と解決方法について理解するまでは、リベラルな名前の使用は避けることをお勧めします。

オブジェクト名に関する一般的な説明は、マニュアル III-501 Object Names を参照してください。

## 次元の数

Igor Pro マニュアル : II-66 ページをもとに編集

ウェーブは1次元から4次元で構成されます。

ウェーブを作成するときに、これを決定します。

Redimension コマンド (本ファイルの末尾かマニュアル V-788) を使ってこれを変更することができます。

詳細はマニュアル II-6 Multidimensional Waves を参照してください。

## ウェーブのデータタイプ

各ウェーブには、格納するデータの形式を決定するデータタイプがあります。

ウェーブのデータタイプは作成時に設定します。

Data Browser、Redimension コマンド（本ファイルの末尾かマニュアル V-788 を参照）、Redimension ダイアログを使って変更できます。

ウェーブデータのタイプには3つのクラスがあります。

- 数値データタイプ
- テキスト
- 参照（ウェーブ参照およびデータフォルダー参照）

各数値データタイプは、実数または複素数になります。

テキスト、参照データタイプは複素数にはできません。

参照データタイプはプログラミングでのみ使用されます。

WaveType 関数を使うと、プログラムでウェーブのデータタイプを決定することができます。

### 数値データタイプ

この表は、Igor で使うことができる数値の精度を示しています。

精度	範囲	バイト/ポイント
倍精度浮動小数点	$10^{-324} \sim 10^{+307}$ (～15 十進数)	8
単精度浮動小数点	$10^{-45} \sim 10^{+38}$ (～7 十進数)	4
符号付き 64-bit 整数	$-2^{63} \sim (2^{63})-1$	8
符号付き 32-bit 整数	$-2,147,483,647 \sim 2,147,483,648$	4
符号付き 16-bit 整数	$-32,768 \sim 32,767$	2
符号付き 8-bit 整数	$-128 \sim 127$	1
符号なし 64-bit 整数	$0 \sim (2^{64})-1$	8
符号なし 32-bit 整数	$0 \sim 4,294,967,295$	4
符号なし 16-bit 整数	$0 \sim 65,535$	2
符号なし 8-bit 整数	$0 \sim 255$	1

64-bit 整数型は、Igor Pro 7.0 で追加されました。

ほとんどの作業では、単精度ウェーブが適切です。

単精度ウェーブは、倍精度の半分しかメモリとディスクスペースを必要としません。

FFT と一部の特殊な演算を除き、ソースウェーブの数値精度に関係なく、計算は倍精度で行われます。

しかし、単精度の狭いダイナミックレンジと低い精度は、すべてのデータに適しているわけではありません。

範囲と精度の制限による数値誤差に詳しくない場合は、倍精度を使って分析を行うほうが安全です。

整数ウェーブはデータ収集を目的としていて、分析での使用を目的としたものではありません。

詳細はマニュアル II-85 Integer Waves を参照してください。

## デフォルトのウェーブプロパティ

Igor Pro マニュアル : II-67 ページをもとに編集

オプションフラグを指定しない Make コマンド (本ファイルの末尾がマニュアル V-526 を参照) でウェーブを作成すると、次のデフォルトのプロパティが設定されます。

プロパティ	デフォルト
ポイント数	128
データタイプ	実数、単精度浮動小数点
X スケーリング	x0=0、dx=1 (ポイントスケーリング)
X 単位	空白
データ単位	空白

これらはウェーブの主なプロパティです。

プロパティの完全なリストは、マニュアル II-88 Wave Properties を参照してください。

ファイルから読み込んだり、テーブルに入力してウェーブを作成した場合、ポイント数以外のデフォルトのプロパティは同じです。

どのような方法でウェーブフォームを作成する場合でも、XY ペアではなくウェーブフォームで表示する場合は、Change Wave Scaling ダイアログ (メニュー Data → Change Wave Scaling) を使って X スケールと単位を設定する必要があります。

## Make コマンド

ほとんどの場合、ファイルからデータを読み込む（マニュアル II-126 Importing Data を参照）、テーブルに入力する（マニュアル II-239 Using a Table to Create New Waves を参照）、または既存のウェーブを複製する（マニュアル II-70 Duplicate Operation を参照）ことで、ウェーブを作成できます。

Make コマンドは、新しいウェーブを作成するために使います。詳細は本ファイルの末尾かマニュアル V-526 Make コマンドを参照してください。

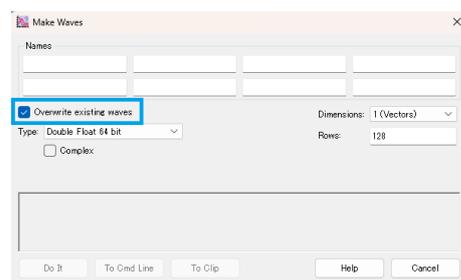
Make コマンドを使う理由をいくつか紹介します：

- ウェーブを作って遊んでみる
- 数学関数をプロットする
- 解析コマンドの出力を保存する
- Igor プロシージャ内のカーブフィッティングに使われるパラメーターや一時的な結果など、さまざまなデータを保持する

Make Waves ダイアログは、Make コマンドへのインターフェースを提供します。

これを使うには、Data メニューから Make Waves を選択します。

ウェーブには明確なポイント数が存在します。



列の末尾にある空白セルを自動的に無視するスプレッドシートプログラムとは異なり、Igor には未使用のポイントという概念はありません。

ウェーブのポイント数を変更するには、Redimension Waves ダイアログ（メニュー Data → Redimension Waves）、または Redimension コマンド（本ファイルの末尾かマニュアル V-788 を参照）を使います。

Overwrite existing waves オプション (/O フラグに対応) は、これから作成するウェーブと同じ名前のウェーブが存在するかどうかを気にしない、または知らない場合に便利です。

## Make コマンドの例

カーブフィッティングで使う係数を作成する：

```
Make/O coefs = {1.5, 2e-3, .01}
```

数学関数をプロットするためのウェーブを作成する：

```
Make/O/N=200 test; SetScale x 0, 2*PI, test; test = sin(x)
```

画像または等高線プロット用の 2D ウェーブを作成する：

```
Make/O/N=(20,20) w2D; w2D = (p-10)*(q-10)
```

カテゴリプロット用のテキストウェーブを作成する

```
Make/O/T quarters = {"Q1", "Q2", "Q3", "Q4"}
```

既存のウェーブのクローンを作成することはしばしば便利です。

これには Make を使わないでください。

代わりに Duplicate コマンドを使います（本ファイルの末尾かマニュアル V-185 を参照）。

Make/O は既存のウェーブの内容を保存しません。

ポイント数、数値精度、数値タイプを変更すると、ウェーブ内にゴミが残ります。

したがって、Make/O を実行した後は、ウェーブの内容について何も想定すべきではありません。

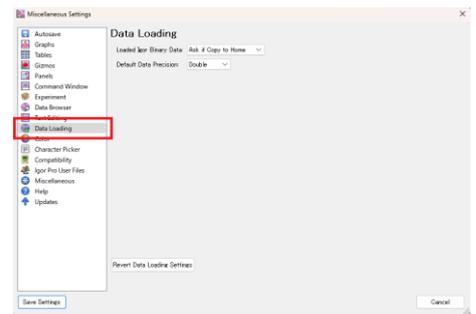
対象のウェーブが存在することがわかっている場合は、Make の代わりに Redimension コマンドを使うことができます。

Redimension はウェーブの内容を保持します。

## ウェーブと Miscellaneous Settings ダイアログ

Igor Pro マニュアル : II-68 ページをもとに編集

Make Waves ダイアログの Type ポップアップメニューの状態、  
テーブルに入力して作成したウェーブの精度、Igor バイナリウェーブのロード方法（コピーまたは共有）は、Miscellaneous Settings  
ダイアログ（メニュー Misc → Miscellaneous Settings）の Data  
Loading カテゴリでプリセットされています。



Loaded Igor Binary Data は、Igor のバイナリファイルを読み込む時の挙動を示します。

- Share (do not copy) : コピーせずに共有状態にする
- Always Copy to Home : 常にホームにコピーする
- Ask if Copy to Home : ホームにコピーするか聞く

Default Data Precision は、ウェーブを作成したときのデフォルトの精度です。

- Double : 倍精度
- Single : 単精度

Revert Data Loading Settings ボタンを押すと、Igor のデフォルト設定に戻ります。

## 次元とデータスケールの変更

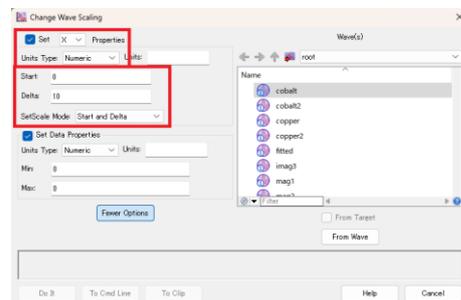
Igor Pro マニュアル : II-68 ページ以降をもとに編集

1D ウェーブを作成すると、デフォルトで X スケール、X の単位、データの単位が設定されます。これらのプロパティを変更するには SetScale コマンド（本ファイルの末尾かマニュアル V-853 を参照）を使う必要があります。

Change Wave Scaling ダイアログは、SetScale コマンドへのインターフェースを提供します。

これを使うには、Data メニューから Change Wave Scaling を選択します。

スケールされた次元インデックスは、通常の数値、日付、時刻、日付/時刻の値を表すことができます。



もっとも一般的なケースでは、これらは通常の数値を表しており、ダイアログの Set X Properties セクションにある Units Type ポップアップメニューはデフォルト値の Numeric のままにしておいてください。

データがウェーブフォームデータの場合は、適切な Start と Delta X の値を入力してください。

データが XY データの場合は、Start に 0、Delta に 1 を入力してください。

これにより、デフォルトの「ポイントスケール」が適用され、ポイントの X 値はポイント番号と同じになります。

通常、Set X Properties と Set Data Properties チェックボックスは選択されたままにしておくべきです。

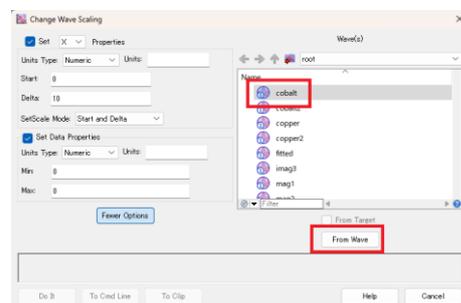
X のみ、またはデータプロパティのみを設定するコマンドをダイアログで生成したい場合は、そのうちの1つを選択解除します。

多次元データを使う時には、Set X Properties の X をポップアップメニューから Y、Z、T に変更することができます。

マニュアル II-6 Multidimensional Waves を参照してください。

特定のウェーブのプロパティを見たい場合は、リストでそのウェーブをダブルクリックするか、ウェーブを選択してから From Wave ボタンをクリックします。

これにより、そのウェーブのプロパティに従って、すべてのダイアログの項目が設定されます。



Igor は、次元とデータ単位を使ってグラフの軸を自動的にラベル付けします。

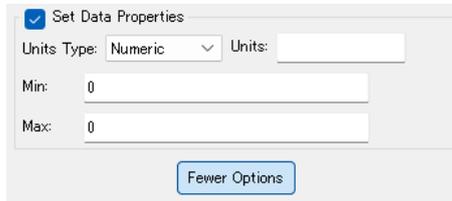
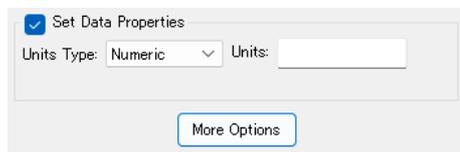
Igor は、49 バイト以下の単位を扱うことができます。

通常、単位は「m」、「s」、「g」などの短い標準的な略語を使うべきです。

データに複雑な単位が含まれている場合、そのまま複雑な単位を入力することもでき、また、単位を空白のままにしておくこともできます。

## 高度な次元とデータスケール

More Options をクリックすると、ダイアログにいくつかの追加項目を表示します。



これらは、X スケーリングを指定する2つの方法を追加し、ウェーブの「データフルスケール」値を設定できるようにします。

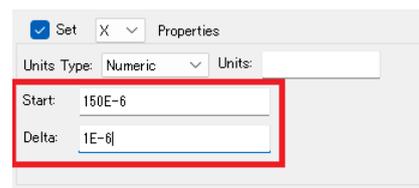
X 値の計算方法は1つだけですが、x0 と dx の値を指定する方法は3つあります。

SetScale Mode ポップアップメニューは、上記のスケーリングエントリーの意味を変更します。

もっとも簡単な方法は、x0 と dx を直接指定する方法です。

これはダイアログの Start and Delta モードであり、More Options ボタンをクリックしない限り、これが唯一のスケーリング設定方法です。

例えば、t=0 の 150  $\mu$ s 後に 1MHz でサンプリングするように設定されたデジタイザーで取得したデータがある場合、Start には 150E-6、Delta には 1E-6 と入力します。



X スケーリングを指定する他の2つの方法は、開始と終了の X 値を設定することと、点の数から dx を計算することです。

Start and End モードでは、最後のデータポイントの X 値を指定します。

Start and Right モードでは、最後の区間の最後の X 値を指定します。

例えば、上記のデジタイザーが 100 ポイントのウェーブを作成したと仮定すると、どちらのモードでも Start として 150E-6 を入力します。

Start and End モードを選択した場合、End には 249E-6 (150E-6 + 99\*1E-6) を入力します。

もし、Start and Right を選択した場合は、Right に 250E-6 を入力します。

最小値 (Min) と最大値 (Max) の入力により、ウェーブの「データフルスケール」と呼ばれるプロパティを設定することができます。

このプロパティは重要な役割を果たすものではありません。

Igor は、計算やグラフ作成の目的でこれを使うことはありません。

これは、ウェーブデータが取得された状況を記録するための方法にすぎません。

例えば、デジタルオシロスコープから取得したデータが  $\pm 10$ V の範囲であった場合、最小値には -10、最大値には +10 を入力します。

ウェーブを作成すると、両方とも最初はゼロに設定されます。

データに意味のあるデータフルスケールがある場合は、適切に設定することができます。

そうでない場合は、ゼロのままにしておきます。

一方、データ単位は、次元単位と同様に、グラフ作成の目的で使用されます。

## 日付、時刻、日付/時刻の単位

「dat」という単位は特殊で、ウェーブのスケーリングされた次元インデックスまたはデータ値に日付、時刻、日付/時刻の情報が含まれていることを指定します。

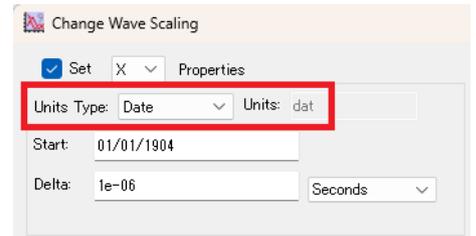
ウェーブデータを持っている場合は、ウェーブフォームの X 単位を「dat」に設定します。

XY データを持っている場合は、X ウェーブのデータ単位を「dat」に設定します。  
この場合、日付を正確に表現するには、X ウェーブは倍精度浮動小数点でなければなりません。

例えば、1日1回測定される何らかの数量を含むウェーブフォームの場合、ウェーブの X 単位を「dat」に設定し、最初の測定が行われた日付を X の開始値に設定し、Delta X 値を1日に設定します。

Units Type ポップアップメニューから Date を選択すると、X 単位が「dat」に設定されます。

開始値は、Igor が内部的に日付を表現する方法である 1904 年 1 月 1 日からの秒数ではなく、日付として入力することができます。



Igor がウェーブフォームをグラフ化しようとするとき、X 軸の単位が「dat」であることを認識し、X 軸に日付を表示します。

ウェーブフォームではなく XY ペアがある場合、ダイアログの Set Data Properties セクションにある Units Type ポップアップメニューから「Date」を選択し、X ウェーブのデータ単位を「dat」に設定します。  
XY ペアをグラフ化すると、Igor は X ウェーブに日付が含まれていることを認識し、X 軸に日付を表示します。

Units Type ポップアップメニューは、ウェーブのプロパティに直接対応するものではありません。

つまり、ウェーブには単位タイプのプロパティはありません。

これらのメニューは、ダイアログが適切なフォーマットで値を表示できるように、どのような種類の値があるかを識別するためのものです。

## Duplicate (複製) コマンド

Duplicate は、便利で頻繁に利用されるコマンドです。

既存のウェーブとまったく同じ新しいウェーブを作成することができます。

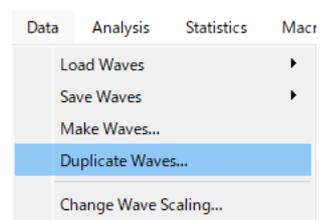
また、ウェーブの一部を複製することもできるため、大きなウェーブをいくつかの小さなウェーブに分割する簡単な方法を提供します。

Duplicate を使う理由をいくつか挙げておきます：

- 元のデータを保持しながら、変換結果（積分、微分、FFT など）を保持する
- カーブフィッティングの「目標」を保持する
- プロシージャ内で一時的な結果を保持する
- ウェーブの一部を抽出する

Duplicate Wave ダイアログは、Duplicate コマンドへのインターフェイスを提供します（本ファイルの末尾かマニュアル V-185 を参照）。

これを使うには、メニュー Data → Duplicate Waves を選択します。

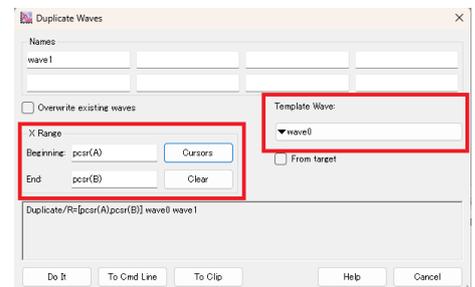


カーソルボタンはグラフと組み合わせて使います。

テンプレートウェーブのグラフを作成することができます。

次に、抽出したいテンプレートのセクションにカーソルを置きます。

Data メニューから Duplicate Waves を選択し、Cursors ボタンをクリックします。



Do It をクリックします。

これにより、カーソルで指定したテンプレートウェーブの該当部分が複製されます。

Duplicate (複製) を使うべきところで、Make (作成) を使ってしまうというミスをすることがあります。

例えば、カーブフィッティングの目標のウェーブは、ソースウェーブと同じポイント数、数値タイプ、数値精度でなければなりません。

ソースウェーブを複製すると、上書きされていない真の値であることが保証されます。

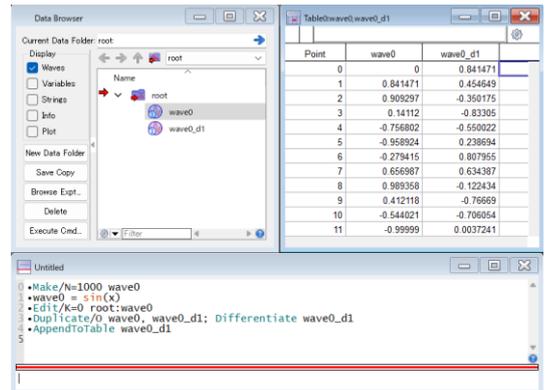
## Duplicate コマンドの例

ウェーブを複製し、そのクローンを変形する：

```
Make/N=1000 wave0
```

```
wave0 = sin(x)
```

```
Duplicate/O wave0, wave0_d1; Differentiate wave0_d1
```

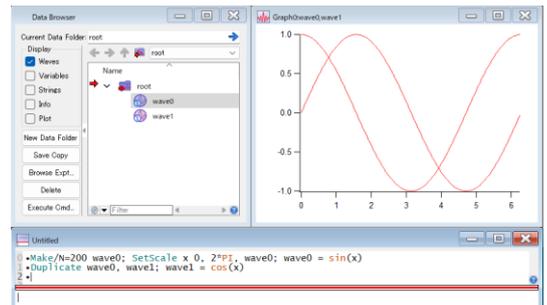


Duplicate を使って、テンプレートウェーブのプロパティを継承する：

```
Make/N=200 wave0; SetScale x 0, 2*PI, wave0; wave0 =
```

```
sin(x)
```

```
Duplicate wave0, wave1; wave1 = cos(x)
```

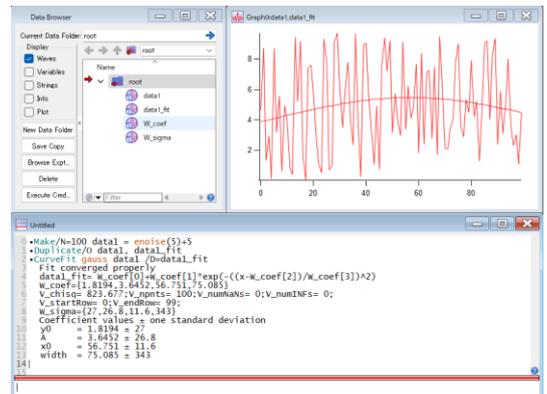


カーブフィッティング用の目標ウェーブを作成する：

```
Make/N=100 data1 = enoise(5)+5
```

```
Duplicate/O data1, data1_fit
```

```
CurveFit gauss data1 /D=data1_fit
```



ウェーブの前半と後半を比較する：

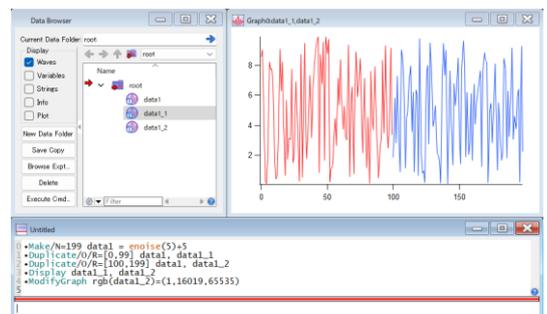
```
Make/N=199 data1 = enoise(5)+5
```

```
Duplicate/O/R=[0,99] data1, data1_1
```

```
Duplicate/O/R=[100,199] data1, data1_2
```

```
Display data1_1, data1_2
```

```
ModifyGraph rgb(data1_2)=(1,16019,65535)
```



Duplicate コマンドでは、/O フラグ (上書き) をよく使います。

これは、新しく作成したいウェーブ名でウェーブが既に存在しているかどうかを知る必要も、気にする必要もないためです。

## ウェーブの Kill (削除)

KillWaves コマンド (本ファイルの末尾かマニュアル V-471 を参照) は、現在の Experiment からウェーブを削除します。

これにより、ウェーブによって使われていたメモリが解放されます。

必要がなくなったウェーブは、ダイアログのリストやポップアップメニューを煩雑にします。

それらを削除することで、この煩雑さを軽減できます。

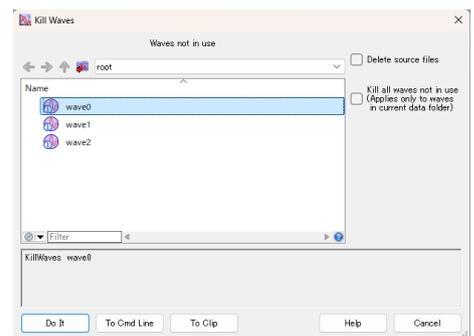
KillWaves を使う状況をいくつか紹介します :

- ファイルから読み込んだデータの確認が終了した
- 実験用に作成したウェーブの使用が終了した
- プロシージャで一時的に使うために作成したウェーブが必要なくなった

Kill Waves ダイアログは、KillWaves コマンドへのインターフェースを提供します。

これを使うには、Data メニューから Kill Waves を選択します。

Igor は、グラフ、テーブル、ユーザー定義関数で使われているウェーブを Kill して、リストに表示されないようにします。



### 注記 :

Igor は、マクロから参照されているウェーブかどうかを判断できません。

そのため、マクロから参照されているが他の方法では使用されていないウェーブは削除することができます。

最もよくあるケースは、グラフを閉じて再作成マクロとして保存する場合です。

グラフで使われていたウェーブは、現在はマクロでのみ使われているため、それらを Kill することができます。

その場合、削除すると、グラフ再作成マクロを実行すると、グラフを再作成できなくなります。

KillWaves は、ウェーブが読み込まれた「ソースファイル」と呼ばれる Igor バイナリウェーブファイルを削除することができます。

これは通常、必要ありません。

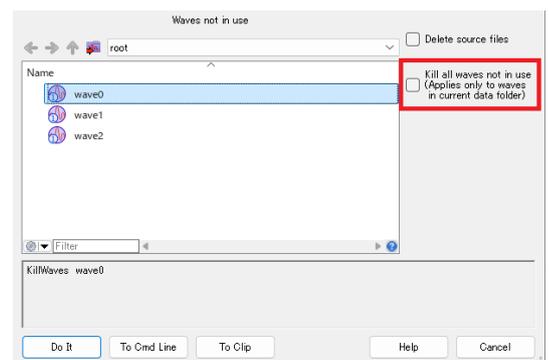
削除しようとしているウェーブは、もともとディスクに保存されたことがないか、パックされた Experiment ファイルの一部として保存され、スタンドアロンファイルからロードされていないためです。

Kill all waves not in use オプションは、ロード、グラフ化、バッチ処理などのプロシージャを含む Experiment を作成するような状況を想定しています。

1つのウェーブのバッチを処理した後、すべてのグラフとテーブルを消去し、次のバッチをロードする準備として、Experiment 内のすべてのウェーブを消去することができます。

これは現在のデータフォルダー内のウェーブのみに影響します。

他のデータフォルダー内のウェーブは削除されません。



## KillWaves コマンドの例

KillWaves を使った簡単な例を紹介します。

```
// すべてのターゲットウィンドウとすべてのウェーブを Kill
// ターゲットではないウィンドウは Kill しない (プロシージャ、ヘルプウィンドウ)
Function KillEverything()
  String windowName
  do
    windowName = WinName(0, 1+2+4+16+64) // 次のターゲットウィンドウを取得
    if (CmpStr(windowName, "") == 0) // 名前が "" (空白) の場合
      break // 完了なのでループを停止
    endif
    KillWindow $windowName // このターゲットウィンドウを Kill
  while (1)
  KillWaves/A // すべてのウェーブを Kill
End
```

// プロシージャ内で一時的に使われたウェーブを Kill する方法

```
Function Median(w) // ウェーブ w の中央値を返す
  Wave w
  Variable result
  Duplicate/O w, temp // ウェーブのクローンを作成
  Sort temp, temp // クローンをソート
  result = temp[numpts(temp)/2]
  KillWaves temp // クローンを Kill
  return result
End
```

その他の例については、WaveMetrics Procedures フォルダ内の Kill Waves プロシージャファイルを参照してください。

## ウェーブのブラウズ

Igor Pro マニュアル : II-72 ページをもとに編集

Data Browser (Data メニュー) では、任意の時点で存在するウェーブ (および文字列や変数) を確認できます。また、どのようなデータフォルダーが存在するのかが確認し、カレントデータフォルダーを設定することができます。

Data Browser についての詳細は、マニュアル II-8 Data Folders を参照してください。

(Igor Pro 6 では、Data → Browse Waves メニューからアクセスする Browse Waves ダイアログがありました。これは、Data Browser と同じ機能を提供していたため、Igor Pro 7.0 でダイアログとメニュー項目が削除されました。)

## ウェーブの名前の変更

次の方法でウェーブの名前を変更することができます :

- Data Browser
- Rename ダイアログ (Data メニュー)
- コマンドラインの Rename コマンド

Rename コマンド (本ファイルの末尾がマニュアル V-796 を参照) はウェーブだけでなく、他のオブジェクトの名前も変更します。

ウェーブの名前を変更する理由をいくつか紹介します。

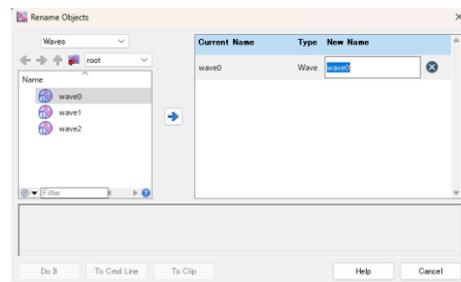
- ファイルから多数のウェーブを読み込み、Igor が自動的にウェーブに名前を付けたとき
- ウェーブの命名規則を決め、既存のウェーブをその規則に合わせるとき
- 既存のウェーブと同じ名前のウェーブのセットを読み込もうとしていて、既存のウェーブをメモリから削除せずに、そのまま残しておきたいとき (新しいデータフォルダーに移動させることでも、同じ結果を得ることができます)

Rename コマンドを使うには、Data メニューから Rename を選択してください。

Rename Objects ダイアログが表示されます。

変更したいオブジェクトの種類を左上のポップアップメニューから選択します。

名前を変更するオブジェクトを選択して、矢印ボタンで右に移動し名前を変更します。



## ウェーブの次元の変更

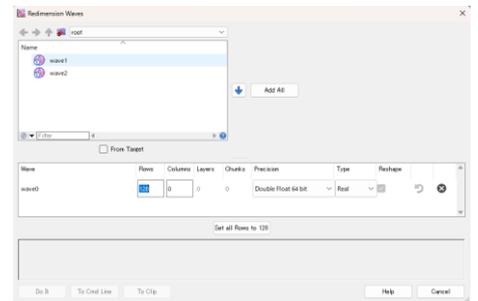
Redimension コマンドは、ウェーブの次のプロパティを変更することができます。

- ウェーブの次元数
- 各次元の要素の数
- 数値の精度 (例: 単精度から倍精度へ)
- 数値タイプ (例: 実数から複素数へ)

Redimension Waves ダイアログは、Redimension コマンド (本ファイルの末尾かマニュアル V-788 を参照) へのインターフェースを提供します。

これを表示するには、Data メニューから Redimension Waves を選択します。

Redimension がウェーブに新しい要素を追加するとき、数値ウェーブの場合はゼロに、テキストウェーブの場合は空白に設定します。



以下のコマンドは、ウェーブの数値精度を変更する 2 つの方法を示しています。

Redimension はウェーブの内容を保持しますが、Make は保持しません。

```
Make/N=5 wave0=x
```

```
Edit wave0
```

```
Redimension/D wave0 // これは wave0 の内容を保持する
```

```
Make/O/D/N=5 wave0 // これは保持しない
```

データを保持したまま 1D ウェーブを 2D ウェーブに変換する方法 (つまり、リシェイプ) については、マニュアル II-98 Vector (Waveform) to Matrix Conversion を参照してください。

ウェーブを数値からテキストに、またはテキストから数値に変更することはできません。

次の例では、数値ウェーブのテキストコピーと、テキストウェーブの数値コピーの作成方法を説明します。

```
Make/N=10 numWave = p
```

```
Make/T/N=(numpnts(numWave)) textWave = num2str(numWave)
```

```
Make/N=(numpnts(textWave)) numWave2 = str2num(textWave)
```

ただし、num2str は 6 桁の精度でしか表示しないため、精度が失われる可能性があります。

## ポイントの挿入

ウェーブに新しいポイントを追加する2つの方法があります。  
次のようにして実行できます：

- InsertPoints コマンドを使う
- テーブルで入力または貼り付け

このセクションでは、InsertPoints コマンド（本ファイルの末尾かマニュアル V-443 を参照）について説明します。

テーブルへの入力、または貼り付けに関する説明は、マニュアル II-12 Tables を参照してください。

InsertPoints コマンドを使うと、1D ウェーブの開始、途中、末尾に新しいデータポイントを挿入することができます。

多次元ウェーブに新しい要素を挿入することもできます。

例えば、2D マトリックスウェーブに新しい列を挿入することができます。

挿入された値は、数値ウェーブの場合は 0、テキストウェーブの場合は "" となります。

Insert Points ダイアログは、InsertPoints コマンドへのインターフェイスを提供します。

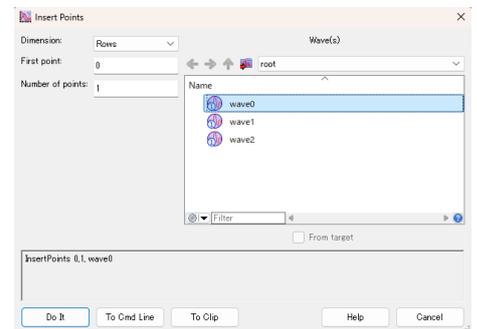
これを表示するには、Data メニューから Insert Points を選択します。

最初のポイントとして入力した値が、選択したウェーブの選択した次元の要素数より大きい場合、新しいポイントは次元の最後に追加されます。

InsertPoints はウェーブの次元を変えることができます。

例えば、1D ウェーブに列を挿入すると、2D ウェーブが作成されます。

メニューから Insert Points を選択した時点で、最前面のウィンドウがテーブルの場合、Igor はテーブル内の選択に基づいてダイアログ項目をプリセットします。



## ポイントの削除

ウェーブからポイントを削除する2つの方法があります。  
次のようにして実行できます：

- DeletePoints コマンドを使う
- テーブルでカットする

このセクションでは、DeletePoints コマンド（本ファイルの末尾かマニュアル V-157 を参照）について説明します。

テーブルでのカットに関する説明は、マニュアル II-12 Tables を参照してください。

DeletePoints コマンドを使うと、1D ウェーブの開始、途中、末尾からデータポイントを削除することができます。

多次元ウェーブから要素を削除することもできます。

例えば、2D マトリックスウェーブから列を削除することができます。

Delete Points ダイアログは、DeletePoints コマンドへのインターフェイスを提供します。

これを表示するには、Data メニューから Delete Points を選択します。

最初のポイントとして入力した値が、選択したウェーブの選択した次元の要素数より大きい場合、DeletePoints はそのウェーブに対して何も行いません。

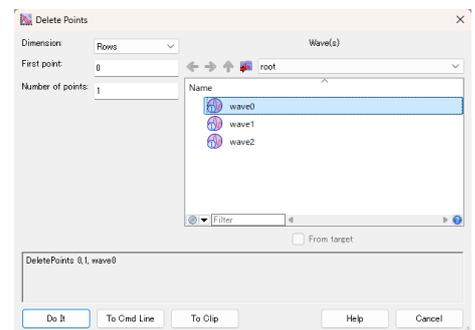
要素の数が多すぎる場合、DeletePoints は指定された最初の要素から次元の終わりまでを削除します。

すべての要素を削除する場合は、ウェーブは元の次元のままです。

DeletePoints はウェーブの次元を変更しません。

次元を変更する場合は、Redimension を使用してください。

メニューから Delete Points を選択した時点で、最前面のウィンドウがテーブルの場合、Igor はテーブル内の選択に基づいてダイアログ項目をプリセットします。



## 本セクションで使用したコマンドの説明 (Volume V Reference)

### Make

Igor Pro マニュアル : V-526 をもとに編集

#### 文法

**Make** [*flags*] *waveName* [, *waveName*]...

**Make** [*flags*] *waveName* [= {*n0*,*n1*,...}]...

**Make** [*flags*] *waveName* [= {{*n0*,*n1*,...},{*n0*,*n1*,...},...}]...

Make コマンドは、指定したウェーブを作成します。

ウェーブを作成するときには、ウェーブデータに値を割り当てるために { } を使います。

#### フラグ

- /B** 8-bit 符号付き整数ウェーブ、または /U が指定されている場合は、符号なしウェーブを作成します。
- /C** 複素数ウェーブを作成します。
- /D** 倍精度ウェーブを作成します。
- /DF** ウェーブはデータフォルダーへの参照を保持します。  
詳細はマニュアル IV-78 Data Folder References を参照してください。
- /FREE[=*nm*]** フリーウェーブを作成します。  
関数内でのみ使用可能で、単純な名前またはウェーブ参照構造フィールドが指定されている場合のみ許可されます。  
詳細はマニュアル IV-91 Free Waves を参照してください。  
  
*nm* が指定され、ゼロでない場合、*waveName* はフリーウェーブの名前として使われ、デフォルト名「\_free\_」を上書きします。  
フリーウェーブの名前を指定する機能は、デバッグ支援として Igor Pro 9.0 で追加されました。  
詳細はマニュアル IV-95 Free Wave Names と IV-207 Wave Tracking を参照してください。
- /I** 32-bit 符号付き整数ウェーブ、または /U が指定されている場合は、符号なしウェーブを作成します。
- /L** 64-bit 符号付き整数ウェーブ、または /U が指定されている場合は、符号なしウェーブを作成します。  
Igor Pro 7.0 以降が必要です。
- /N=*n*** *n* は各ウェーブが持つポイントの数です。  
*n* が式である場合、括弧で囲む必要があります：  
Make/N=(myVar+1) aNewWave
- /N=(*n1*, *n2*, *n3*, *n4*)** *n1*, *n2*, *n3*, *n4* は、各ウェーブの行、列、レイヤー、チャンクの数指定します。  
末尾のゼロは省略できます (例：/N=(*n1*,*n2*,0,0) ⇒ /N=(*n1*,*n2*))。

/O	名前が競合した場合、既存のウェーブを上書きします。 上書き後は、ウェーブの内容を信頼できなくなることがあるため、再初期化するか、適切な値を割り当てる必要があります。
/R	実数値ウェーブを作成します（デフォルト）。
/T	テキストウェーブを作成します。
/T=size	あらかじめ割り当てられたストレージでテキストウェーブを作成します。 size は、各テキストウェーブの各要素にあらかじめ割り当てられたバイト数です。 ウェーブは初期化されていません。 初期化はユーザーの責任で行ってください。  ウェーブに非常に多くのポイントがある場合、ウェーブに割り当てられたすべての文字列があらかじめ割り当てられたサイズとまったく同じサイズである場合に限り、テキストウェーブの割り当てを劇的に高速化することができます。
/U	符号なし整数ウェーブを作成します。
/W	16-bit 符号付き整数ウェーブ、または /U が指定されている場合は、符号なしウェーブを作成します。
/WAVE	ウェーブはウェーブへの参照を保持します。 詳細はマニュアル IV-71 Wave References を参照してください。
/Y=type	次の「ウェーブのデータ形式」を参照してください。

## ウェーブのデータ形式

/B、/C、/D、/I、/L、/R、/T、/U、/W のデータ形式フラグの代わりに、/Y=(numType) を使ってデータ形式を設定することができます。

numType の値については、WaveType 関数を参照してください。

/Y フラグは、他のフラグ形式を上書きします。

同じ関数内の代入文でウェーブを使う場合は、コンパイラによって作成される自動ウェーブ参照をコントロールするために、明示的なデータ形式フラグを使う必要があります。

詳細はマニュアル IV-73 WAVE reference Types を参照してください。

## 詳細

ウェーブ内の要素（行×列×レイヤー×チャンク）の最大許容数は、Igor の 64-bit 版を使っている（最大：214,700,000,000）か、32-bit 版を使っているか（最大：2,147,000,000）によって異なります。

フラグで上書きされない限り、作成されたウェーブは、デフォルトの長さ、形式、精度、単位、スケーリングを持ちます。

出荷時のデフォルトは：

プロパティ	デフォルト
ポイント数	128
精度	単精度浮動小数点
形式	実数

次元	1
x, y, z, t スケーリング	オフセット=0、Delta=1 (ポイントスケーリング)
x, y, z, t 単位	"" (ブランク)
データフルスケール	0, 0
データ単位	"" (ブランク)

---

#### 注記：

Miscellaneous Settings ダイアログで設定した推奨精度は、Make Waves ダイアログのチェックボックスをプリセットし、インポートしたウェーブの精度を決定するだけです。  
これは Make コマンドには影響しません。

## Duplicate

Igor Pro マニュアル : V-185 をもとに編集

#### 文法

```
Duplicate [flags][type flags] srcWaveName, destWaveName [, destWaveName]...
```

Duplicate コマンドは、新しいウェーブを作成します。

その名前は *destWaveName* で指定され、その内容、データ形式、スケーリングは *srcWaveName* と同じです。

#### パラメーター

*srcWaveName* は既存のウェーブの名前でなければなりません。

*destWaveNames* は、既存のウェーブを上書きする /O フラグを使わない限り、現在使用していないウェーブ名である必要があります。

#### フラグ

/FREE[=*nm*]

フリーウェーブを作成します。

関数内でのみ使用可能で、単純な名前またはウェーブ参照構造フィールドが指定されている場合のみ許可されます。

詳細はマニュアル IV-91 Free Waves を参照してください。

*nm* が指定され、ゼロでない場合、*waveName* はフリーウェーブの名前として使われ、デフォルト名「\_free\_」を上書きします。

フリーウェーブの名前を指定する機能は、デバッグ支援として Igor Pro 9.0 で追加されました。

詳細はマニュアル IV-95 Free Wave Names と IV-207 Wave Tracking を参照してください。

/O

*destWaveName* と同じ名前の既存のウェーブを上書きします。

/R=(*startX*,*endX*)

ソースウェーブの X の範囲を指定し、その範囲から目標のウェーブが作成されます。

<pre>/R=(startX,endX)   (startY,endY)</pre>	<p>X と Y の両方の範囲を指定します。 さらに、次元を同様に作成します。</p>
<pre>/R=[startP,endP]</pre>	<p>ソースウェーブの行の範囲を指定し、そこから目標のウェーブが作成されます。 その他の次元はスケーリングされた次元範囲と同じように構築されます。</p>
<pre>/RMD=[firstRow,lastRow]   [firstColumn,lastColumn]   [firstLayer,lastLayer]   [firstChunk,lastChunk]</pre>	<p>コマンドを適用するソースウェーブの連続したデータの範囲を指定します。 このフラグは Igor Pro 7.0 で追加されました。</p> <p>対応する括弧を省略することで、すべての高次元を含めることができます。 例えば、</p> <pre>/RMD=[firstRow,lastRow]</pre> <p>は、すべての使用可能な列、レイヤー、チャンクを含みます。 空の [] を使うと、指定された次元をすべて含めることができます。 例えば、</p> <pre>/RMD=[] [firstColumn,lastColumn]</pre> <p>は、「A 列から B 列までのすべての行」を意味します。 * を使って、任意の次元の終端を指定することができます。 例えば、</p> <pre>/RMD=[firstRow,*]</pre> <p>は、「最初の行から最後の行まで」を意味します。</p>

### 形式フラグ (関数でのみ使用される)

ユーザー定義関数で使われる場合、Duplicate は、/B、/C、/D、/I、/S、/U、/W、/T、/DF、/WAVE フラグも使うことができます。

これらは、Duplicate コマンドの結果には影響しません。

これらのフラグは、実行時にどのようなウェーブが予想されるかを識別するためにのみ使用されます。

この情報は、関数の後で、複製したウェーブを目標として使うウェーブ代入文を作るときに使われます。

```
Function DupIt(wv)
  Wave/C wv // 複素数ウェーブ
  Duplicate/O/C wv,dupWv // dupWv が複素数であることを Igor に指示
  dupWv[0]=cplx(5.0,1.0) // エラーなし。dupWv は既知の複素数ウェーブであるため
  ...
```

Duplicate に /C フラグが指定されていなかった場合、Igor は cplx 関数の結果への dupWv の代入文をコンパイルしようとすると、「function not available for this number type (この数値型では関数が使用できません)」というエラーメッセージを表示します。

これらの形式フラグは、同じ名前の別のウェーブ参照変数と一致させる必要がある場合、または、ウェーブ代入用にコンパイルする式の種類を識別する必要がある場合を除いて、使う必要はありません。

形式フラグの完全なリストと詳細な説明は、マニュアル IV-73 WAVE Reference Types と IV-74 WAVE Reference Type Flags を参照してください。

### 詳細

/R が省略された場合、ウェーブ全体が複製されます。

/R とともに使われる範囲指定において、末尾に \* を付けると、末尾まで複製することを意味します。

また、単に末尾の指定を省略することもできます。

指定した次元をすべて含めるには、

```
/R=[]
```

を使います。

高次元を省略すると、それらの次元はすべて複製されます。

つまり、2D ウェーブの

```
/R=[1,5]
```

は、

```
/R=[1,5][ ]
```

と同等です。

ソースのウェーブがロックされていた場合でも、目標のウェーブは常にロックが解除されます。

## 警告

ユーザー定義関数のループなど、一部の状況においては、Duplicate が予期しない動作を示す場合があります。

ユーザー定義関数内で、

```
Duplicate/O srcWave, DestWaveName
```

を使うとき、これはコンパイル時に DestWaveName というローカル WAVE 変数を作成します。

実行時に WAVE 変数が NULL の場合、現在のデータフォルダーに同じ名前のウェーブが作成されます。

しかし、ループ内であれば WAVE 変数が NULL ではないため、参照されたウェーブはその場所に関わらず上書きされます。

現在のデータフォルダーにウェーブを作成（または上書き）したい場合は、以下の2つの方法のうちの1つを使用してください。

```
Duplicate/O srcWave, $"DestWaveName"
```

```
WAVE DestWaveName // 目標のウェーブを参照する必要がある場合のみ
```

または

```
Duplicate/O srcWave, DestWaveName
```

```
// DestWaveName の使用が完了したら
```

```
WAVE DestWaveName=""
```

## Redimension

[Igor Pro マニュアル : V-788 をもとに編集](#)

### 文法

```
Redimension [flags] waveName [, waveName]'''
```

Redimension コマンドは、指定されたウェーブを再作成し、その内容を可能な限り維持します。

### フラグ

/B ウェーブを 8-bit 符号付き整数に変換し、/U が指定されている場合は、符号なし整数に変換します。

/C 実数ウェーブを複素数ウェーブに変換します。

/D	単精度ウェーブを倍精度ウェーブに変換します。
/E=e	次元の変換モードをコントロールします。 e=0 特別な処理なし (デフォルト) e=1 データを変換または移動せずに次元変更を強制する e=2 エンディアンを入れ替えます。エンディアンバイト並びの説明は FBinRead コマンドを参照してください。
/I	ウェーブを 32-bit 符号付き整数に変換し、/U が指定されている場合は、符号なし整数に変換します。
/L	ウェーブを 64-bit 符号付き整数に変換し、/U が指定されている場合は、符号なし整数に変換します。 Igor Pro 7.0 以降が必要です。
/N=n	n は各ウェーブの新しいポイント数です。 多次元ウェーブは 1 次元に変換されます。 n=1 の場合、ウェーブは元の列数を持つ 1 次元のウェーブに変換されます。
/N=(n1, n2, n3, n4)	n1、n2、n3、n4 は、各ウェーブの行、列、レイヤー、チャンクの数を指定します。 末尾のゼロは省略できます (例: /N=(n1, n2, 0, 0) ⇒ /N=(n1, n2))。 次元のサイズを変更しない場合は、その次元に -1 を指定します。
/R	虚数部分を捨てて、複素数ウェーブを実数に変換します。
/S	倍精度ウェーブを単精度ウェーブに変換します。
/U	整数ウェーブを符号なしに変換します。
/W	ウェーブを 16-bit 整数に変換します (/U が指定されている場合は符号なし)。
/Y=type	ウェーブのデータ形式を指定します。 次の「ウェーブのデータ形式」を参照してください。

## ウェーブのデータ形式

上記の数値形式のフラグの代わりに、/Y=numType を使って数値形式を整数コードとして設定することができます。

コードの値は WaveType コマンドを参照してください。

/Y を他の形式フラグと組み合わせて使用しないでください。

このコマンドでは、実数/複素数の設定を変更せずに数値の形式を変更することはできません。

## 詳細

ウェーブは既に存在している必要があります。

ウェーブの拡張された新しいポイントはゼロに設定されます。

一般的に、Redimension は、データのある次元から別の次元に移動させることはありません。

例えば、6x6 の行列ウェーブがあり、それを 3x12 にしたい場合、行が短くなり、最後の 3 行分のデータが失われます。

特別なケースとして、1D ウェーブへの変換、または 1D ウェーブからの変換の場合、Redimension はデータの場所を変更せずに、ウェーブの次元を変更します。

例えば、Redimension を使って 36 要素の 1D ウェーブを 6x6 行列に変換することができます。

この行列では、最初の列（列 0）の要素は 1D ウェーブの最初の 6 要素、2 番目の列の要素は次の 6 要素、というように続きます。

1D ウェーブから次元変更するときには、まず列が埋められ、次にレイヤー、そしてチャンクが埋められます。

## 例

4つの要素を持つ 1D ウェーブを 2x2 の行列に変形する：

```
Make/N=4 vector=x
```

```
Redimension/N=(2,2) vector
```

## SetScale

Igor Pro マニュアル : V-853 をもとに編集

### 文法

```
SetScale [/I/P] dim, num1, num2 [, unitsStr], waveName [, waveName]...
```

```
SetScale d, num1, num2 [, unitsStr], waveName [, waveName]...
```

SetScale コマンドは、指定されたウェーブの次元スケールまたはデータのフルスケールを設定します。

### パラメーター

最初のパラメーター dim は次のいずれかでなければなりません：

文字	意味
d	データのフルスケール
t	チャンクの次元のスケールリング (t スケールリング)
x	行の次元のスケールリング (x スケールリング)
y	列の次元のスケールリング (y スケールリング)
z	レイヤーの次元のスケールリング (z スケールリング)

任意の次元 (x, y, z, t) のスケールリングを設定する場合、num1 は開始インデックス値であり、次元内の最初のポイントのスケールリングされたインデックスです。

num2 の意味は、/I と /P フラグによって異なります。

/P を使う場合は、num2 はデルタ値、つまり、1つのポイントから次のポイントへのスケールリングされたインデックスの差分となります。

/I を使う場合は、num2 は「終了値」、つまり、次元の最後の要素のインデックス値です。

どちらのフラグも使わない場合、num2 は「右の値 (Right)」、つまり、次元の最後の要素の次の要素を持つインデックス値です。

これら3つの方法は、各ウェーブの各次元に保存される2つのスケールリング値、つまり開始値とデルタ値を指定する3つの異なる方法があるということにすぎません。

データフルスケール (d) を設定する場合、num1 はウェーブの名目最小値、num2 はウェーブの名目最大値です。

データの実測値は使われていません。

これは、ウェーブが到達すると予想される最小値と最大値を記録する目的のみに使われています。

データフルスケールを設定するときにフラグは使われません。

unitsStr パラメーターは、指定したウェーブの x、y、z、t、またはデータ値の自然単位を識別する文字列です。

Igor はこれを使ってグラフの軸に自動的にラベルを付けます。

この文字列は、メートル (m)、グラム (g)、秒 (s) のように、1~49 バイトでなければなりません。

ウェーブに自然単位がない場合は、このパラメーターに "" (空白) を入力できます。

unitsStr を「dat」(大文字と小文字は区別される) に設定すると、ウェーブが Igor の日付/時刻フォーマット (1904 年 1 月 1 日午前 0 時からの秒数) でデータを持つ日付/時刻ウェーブであることを指定することになります。

日付/時刻のウェーブは倍精度でなければなりません。

## フラグ

最大 1 つのフラグのみ許可されていて、次元スケーリング (データの実寸ではない) が、設定されている場合のみ許されます。

/I 包括的なスケーリングです。  
num2 は終了インデックス、つまり、次元内の最後の要素のインデックス値です。

/P ポイントごとのスケーリングです。  
num2 はデルタスケーリング値、つまり、ある要素から次の要素へのスケーリングされたインデックス値の差分です。

## 詳細

SetScale では、デルタスケーリング値をゼロにすることはできません。

デルタ値をゼロにして SetScale コマンドを実行すると、自動的にデルタ値が 1.0 に設定されます。

/P フラグを使わない場合、SetScale は、num1 と num2 を開始インデックス値とデルタインデックス値に変換します。

要素数が、2 つ未満の次元に対して SetScale を呼び出すと、その次元に要素が 2 つあるかのように変換されます。

数値ウェーブの次元スケーリングを設定するときには、unitsStr パラメーターを省略することができます。

Igor はウェーブのスケーリングを自動的に設定しますが、単位は変更しません。

ただし、テキストウェーブの次元スケーリングを設定するときには、unitsStr パラメーターを指定する必要があります (ウェーブに単位がない場合は "" を使う)。

そうしないと、Igor はテキストウェーブを文字列の開始とみなし、それを unitsStr として処理しようとします。

## KillWaves

Igor Pro マニュアル : V-471 をもとに編集

### 文法

```
KillWaves [flags] waveName [, waveName]'''
```

KillWaves コマンドは、指定されたウェーブを Kill (削除、破壊) します。

### フラグ

/A 現在のデータフォルダー内のすべてのウェーブを Kill します。  
/A を使う場合は、waveNames を省略します。

/F waveName がロードされた Igor バイナリウェーブファイルを Kill します。

/Z 使用中のウェーブや存在しないウェーブを Kill しようとしてもエラーを表示しません。

## 詳細

ウェーブが占有していたメモリは、他の用途に使うことができるようになります。  
グラフやテーブルで使われているウェーブや、XOP によって予約されているウェーブは強制削除できません。

XOP は OBJINUSE (Object in use) メッセージを送信することでウェーブを確保します。

現在のバージョンでは廃止された rtGlobals=0 設定でコンパイルされた関数については、ユーザー定義関数から参照されたウェーブも削除することもできません。

## 例

```
KillWaves/A/Z // 現在のデータフォルダー内で使われていないウェーブを Kill (削除) する
```

## Rename

[Igor Pro マニュアル : V-796 をもとに編集](#)

## 文法

```
Rename oldName, newName
```

Rename コマンドは、ウェーブ、文字列、数値変数を oldName から newName に名前を変更します。

## パラメーター

oldName は、単純なオブジェクト名、またはデータフォルダーのパス+名前です。

newName は、単純なオブジェクト名でなければなりません。

## 詳細

すでに存在する名前を使ってオブジェクト名を変更することはできません。

次のコマンドはエラーとなります：

```
Make wave0, wave1  
// wave0 の名前を変更して wave1 を上書きする  
Rename wave0, wave1 // これは動作しない
```

しかし、次のようにすれば、望む結果を得ることができます：

```
Make wave0, wave1  
Duplicate/O wave0, wave1; KillWaves wave0
```

## InsertPoints

Igor Pro マニュアル : V-443 をもとに編集

### 文法

```
InsertPoints [ /M=dim /V=value ] beforePoint, numPoints, waveName [, waveName]...
```

InsertPoints コマンドは、各 waveName において、beforePoint ポイントの前に numPoints 個のポイントを挿入します。

新しいポイントの値はゼロです。

### フラグ

<i>/M=dim</i>	要素が挿入される次元を指定します。 値は、行の場合は 0、列の場合は 1、レイヤーの場合は 2、チャンクの場合は 3 です。 <i>/M</i> が省略された場合、InsertPoints は行次元に挿入します。
<i>/V=value</i>	数値ウェーブと併用する場合、value には新しい要素の値を指定します。 <i>/V</i> を省略すると、新しい要素はゼロに設定されます。 <i>/V</i> フラグは Igor Pro 8.0 で追加されました。

### 詳細

ゼロポイント（ポイント無し）の行以外に点を挿入しようとする、ゼロポイントウェーブが生成されます。何らかの効果を得る前に、まず行数をゼロ以外にしなければなりません。

## DeletePoints

Igor Pro マニュアル : V-157 をもとに編集

### 文法

```
DeletePoints [/M=dim] startElement, numElements, waveName[, waveName]...
```

DeletePoints コマンドは、指定されたウェーブから、要素 startElement から始めて numElements 個の要素を削除します。

### フラグ

<i>/M=dim</i>	dim は、要素が削除される次元を指定します。 次の値をとります： 0 行 1 列 2 レイヤー 3 チャンク <i>/M</i> が省略された場合、DeletePoints は行の次元から削除します。
---------------	---

### 詳細

ウェーブは、ゼロポイントを含む任意の数のポイントを持つことができます。

任意の次元からすべての要素を削除すると、ウェーブからすべてのポイントが削除され、ポイント数がゼロの 1D ウェーブが残ります。

すべての要素を削除する場合を除き、DeletePoints はウェーブの次元を変更しません。  
その場合は Redimension を使用してください。