

CONTENTS

ビジュアルヘルプ - 注釈	3
注釈	3
注釈のクイックスタート	3
注釈ダイアログ	4
注釈の修正	5
注釈テキストの内容	5
注釈テキストのエスケープコード	5
フォントのエスケープコード	6
フォントサイズのエスケープコード	6
相対フォントサイズのエスケープコード	7
特別なエスケープコード	7
タグ用の動的エスケープコード	8
その他の動的エスケープコード	9
TagVal と TagWaveRef 関数	9
注釈のタブ	9
一般的な注釈プロパティ	10
注釈名	10
注釈フレーム	10
注釈の色	11
注釈の配置	11
グラフにおけるテキストボックス、凡例、カラースケールの配置	12
ページレイアウトにおける注釈の配置	13
凡例	13
凡例テキスト	14
ポイントにおけるシンボルの条件	14
凡例テキストの固定	14
マーカーサイズ	14
トレースシンボルの中央揃え	15
トレースシンボルの幅	15
Color as f(z) を持つシンボル	15
タグ	16
タグテキスト	16
タグウェーブとアタッチメントポイント	16

タグのアタッチメントポイントを変更する.....	17
タグの矢印.....	17
タグの線と矢印のスタンドオフ.....	18
タグのアンカーポイント.....	18
タグの配置.....	19
オフスクリーンポイントに付けられたタグ.....	19
コンターラベルはタグです	19
カラースケール	20
ColorScale Main タブ	20
カラースケールのサイズと向き.....	20
ColorScale Axis Labels タブ.....	21
ColorScale Ticks タブ	21
精巧な注釈	22
テキスト情報変数.....	22
テキスト情報変数の例	22
注釈を使ったプログラミング	23
注釈名の変更.....	23
注釈タイプの変更.....	23
注釈テキストの変更	24
プログラムによるテキストの生成	24
注釈の削除.....	24
注釈のエスケープコード	24
汎用のエスケープコード.....	25
タグのエスケープコード.....	27
テキスト情報変数のエスケープコード	27
動的テキストエスケープコード.....	28
凡例記号のエスケープコード.....	29
軸ラベルのエスケープコード.....	29
注釈のエスケープシーケンスにおけるバックスラッシュ	30
Igor TeX.....	30
Igor Pro は TeX の一部の機能をサポート	31
Igor TeX で使われるフォント.....	32

ビジュアルヘルプ – 注釈

注釈

注釈 (Annotation) とは、グラフ、ページレイアウト、または Gizmo プロットに情報を追加するカスタムオブジェクトのことです。

ほとんどの注釈には、グラフの内容を説明したり、ウェーブの特徴を指摘したり、ウェーブに適用される軸を特定したり、凡例を作成したりするために使用できるテキストが含まれています。

また、注釈には、コンタープロットや画像プロットにおいて、色に対応するデータの範囲を示すカラースケールを含めることもできます。

注釈には、テキストボックス、凡例、カラースケール、タグの4種類があります。

テキストボックスには、1行または複数の行のテキストが含まれており、必要に応じて枠で囲んだり、回転させたり、色を付けたり、配置を調整したりすることができます。

凡例はテキストボックスと似ていますが、グラフ内の1つ以上のウェーブに対応するトレース記号が含まれている点が異なります。

グラフにウェーブが追加されたり削除されたりした場合、あるいはウェーブの表示が変更されたりすると、凡例は自動的に更新されます。

タグもテキストボックスと似ていますが、トレースや画像の特定のポイントに紐付けられ、そのポイントを説明するテキストを動的に更新できる点が異なります。

タグはグラフに追加できますが、ページレイアウトや Gizmo プロットには追加できません。

コンタープロットでは、自動的にタグを生成してコンター線にラベルを付けます。

カラースケールには、データに関連付けられた色の範囲をカバーする軸を持つカラーバーが含まれています。

関連付けられたデータが変更されると、カラースケールは自動的に更新されます。

また、名前付きカラーテーブルと軸の数値範囲を明示的に指定することで、カラースケールをデータから完全に切り離すこともできます。

注釈のクイックスタート

これを行うには

グラフに注釈を追加するには

ページレイアウトに注釈を追加するには

Gizmo プロットに注釈を追加するには

グラフまたは Gizmo 内の注釈を変更するには

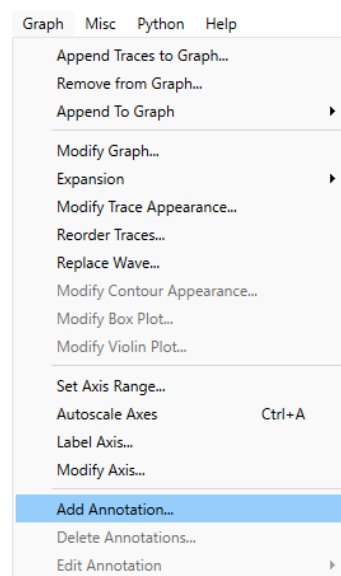
このようにします

Graph メニューから Add Annotation を選択します。

Layout メニューから Add Annotation を選択するか、注釈ツール ("A") をクリックします。

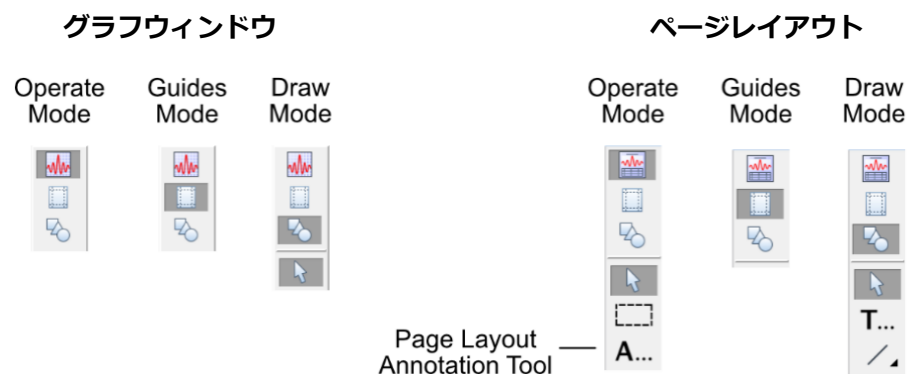
Gizmo メニューから Add Annotation を選択します。

注釈をダブルクリックします。これにより、Modify Annotation ダイアログが表示されます。



ページレイアウト内の注釈を変更するには	注釈ツールで注釈を 1 回クリックします。これにより、Modify Annotation ダイアログが表示されます。
注釈の種類を変更するには	Modify Annotation ダイアログの Annotation ポップアップメニューを使うか、適切な Tag、TextBox、ColorScale、または Legend コマンドを実行してください。
既存の注釈を移動するには	注釈をクリックして、新しい位置までドラッグします。注釈が固定されている場合は、この操作はできません。その場合は、注釈をダブルクリックし、Annotation Position タブで移動可能に設定してください。
	タグの接続ポイントを変更するには、Alt キーを押しながらタグのテキストをウェーブの新しい接続ポイントまでドラッグします。この操作は、タグが固定しているかどうかに関係なく実行できます。
既存の注釈を複製するには	注釈をダブルクリックし、Modify Annotation ダイアログの右上隅にある Duplicate Textbox ボタンをクリックします。注釈がタグの場合は、ボタンのラベルが Duplicate Tag などになります。
注釈を削除するには	注釈をダブルクリックし、Modify Annotation ダイアログで Delete ボタンをクリックします。タグは、その接続ポイントをグラフの外へドラッグすることで削除できます。

マウスを使って注釈を操作する時は、グラフまたはページのレイアウトが「ガイド」モードや「描画」モードではなく、「操作」モードになっていることを確認してください。
ウィンドウがどのモードにあるかは、ツールパレットで確認できます。



注釈ダイアログ

注釈ダイアログを使うと、新しい注釈を作成したり、既存の注釈を編集したりすることができます。
注釈ダイアログは複雑に見えますが、主な機能はわずかしかありません。

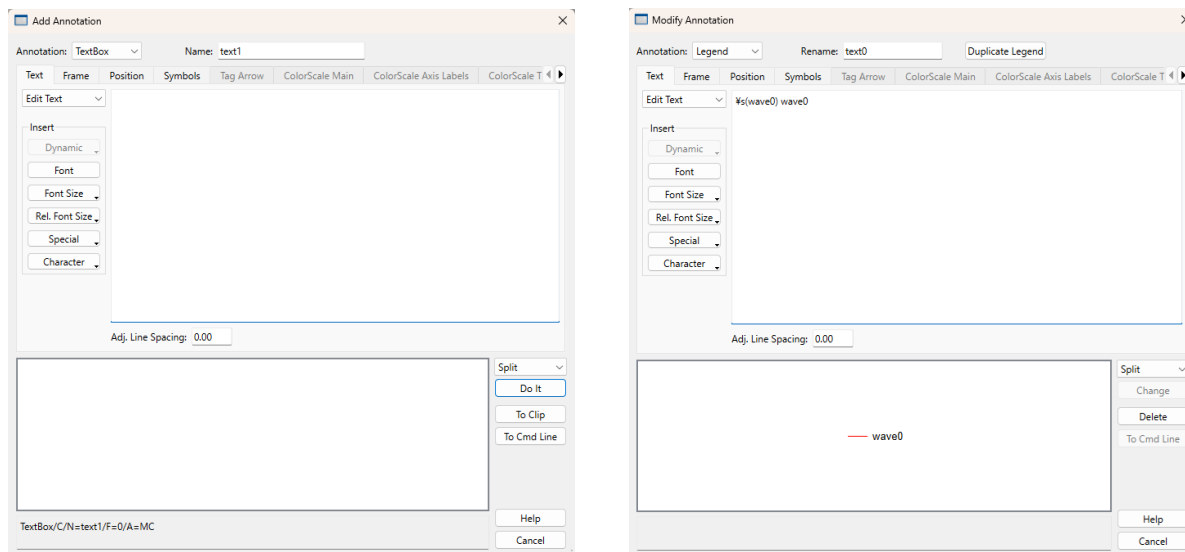
- ・ 注釈の種類を選択するためのポップアップメニュー
- ・ 名前の設定
- ・ 関連する注釈設定をまとめたタブ
- ・ 注釈がどのように表示されるかを確認したり、コマンドを表示したりするためのプレビューボックス
- ・ 通常の Igor のダイアログボタン

注釈の修正

グラフや Gizmo プロットにすでに注釈がある場合、「操作」モードでその注釈をダブルクリックすることで、注釈を変更できます。

これにより、Modify Annotation ダイアログが表示されます。

ページレイアウトで注釈を変更するには、まずレイアウトモードで注釈ツール（"A"）を選択する必要があります。次に、注釈をシングルクリックして、Modify Annotation ダイアログを表示します。



注釈テキストの内容

Text タブにある Annotation テキスト入力エリアにテキストを入力します。

注釈テキストには、プレーンテキストと、上付き文字、フォント、フォントサイズやスタイル、配置、テキストの色などの特殊効果を生み出す「エスケープコード」の両方が含まれる場合があります。

テキストは複数行にわたることもできます。

プレーンテキストを入力している最中ならいつでも、Insert グループ内のポップアップメニューから特殊効果を選択すると、Igor Pro が対応するエスケープコードを挿入します。

ウィザードでは、これらを直接入力することもできます。

注釈テキストを入力すると、プレビューボックスに、その注釈がどのように表示されるかが表示されます。プレビューボックスにはテキストを入力できません。

注釈テキストのエスケープコード

エスケープコードは、バックスラッシュ文字の後に 1 つ以上の文字が続く形式で構成されます。

これは、選択した特殊効果を表しています。

エスケープコードの効果は、後続のエスケープコードによって上書きされるまで持続します。

エスケープコードは複雑ですが、その効果は「プレビュー」ボックスで確認できます。

以下の例では、下付き文字のエスケープコード「\B」が下付き文字の開始を示しており、注釈には表示されません。その後に続く「n」は、下付き文字として表示されるプレーンテキストです。

通常のエスケープコード「\M」は下付き文字モードを上書きするため、その後に続くプレーンテキスト「= z」は、「J」に使われた元のサイズと Y 位置（垂直オフセット）で表示されます。

Subscript “escape code” Normal “escape code”



「注釈のエスケープコード」のセクションでは、エスケープコードに関する詳細を説明しています。

フォントのエスケープコード

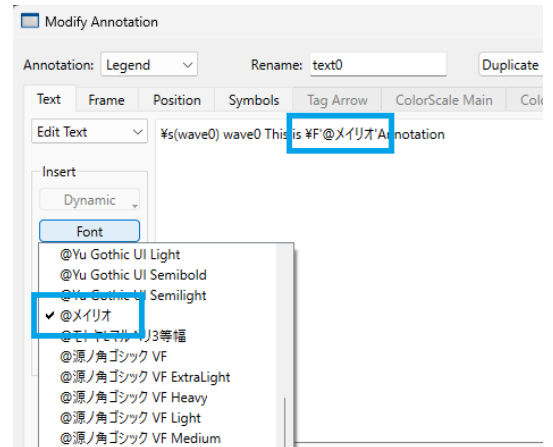
Font ポップアップメニューから項目を選択すると、注釈内のそれ以降の文字のフォントを変更するコードが挿入されます。

選択されたフォントは、注釈のテキスト入力領域内の現在の挿入位置で有効になっているフォントです。

フォントを選択しない場合、Igor Pro はデフォルトのフォント、またはグラフ内の注釈にはグラフ用フォントを使います。

デフォルトのフォントは Misc メニューの Default Font 項目で設定でき、グラフ用フォントは Graph メニューの Modify Graph 項目で設定できます。

Font ポップアップメニューには、Recall font（フォントを呼び出す）項目もあります。



この項目は、複雑な注釈で使われ、「テキスト情報変数のエスケープコード」のセクションで説明しています。

フォントサイズのエスケープコード

Font Size ポップアップメニューから項目を選択すると、注釈内のそれ以降の文字のフォントサイズを変更するコードが挿入されます。

選択されたフォントサイズは、注釈のテキスト入力エリア内の現在の挿入位置で有効になっているサイズです。

表示されていないサイズを挿入するには、表示されているサイズをいずれか選択し、エスケープコードを編集して目的のフォントサイズを指定してください。

注釈のフォントサイズは、03~99 ポイントの範囲で指定できます。

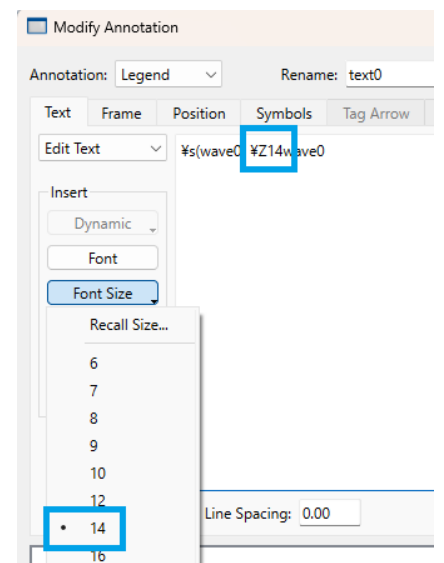
“\Z” エスケープコードの後ろには、2桁の数字を入力する必要があります。

グラフ内の注釈についてフォントサイズのエスケープコードを指定しない場合、Modify Graph ダイアログでグラフのフォントサイズを指定していない限り、グラフのサイズに適したフォントサイズを自動的に選択します。

ページレイアウト内の注釈のデフォルトのフォントサイズは 10 ポイントです。

Font Size ポップアップメニューには、「Recall size」という項目があります。

この項目は、複雑な注釈で使われ、「テキスト情報変数のエスケープコード」のセクションで説明しています。



相対フォントサイズのエスケープコード

Rel. Font Size ポップアップメニューから項目を選択すると、注釈内のその後の文字の相対フォントサイズを変更するコードが挿入されます。

フォントサイズを大きくするには 100 より大きい値を、小さくするには 100 より小さい値を使ってください。

表示されていないサイズを挿入するには、表示されている相対サイズを任意に選択し、エスケープコードを編集して、目的の相対フォントサイズを指定してください。

注釈の相対フォントサイズは 001 から 999 (1% から 999%) まで指定できます。

"\Zr" エスケープコードの後に 3 桁の数字を指定する必要があります。

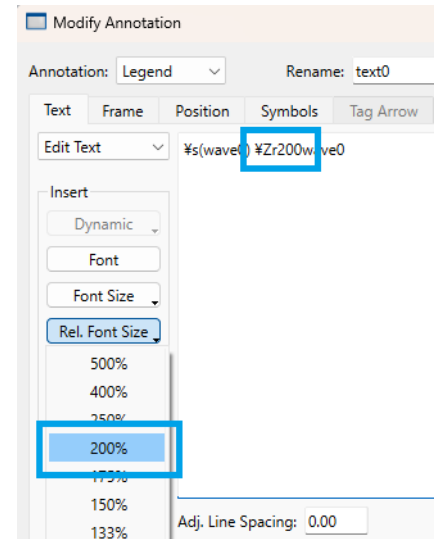
例えば、50% の後に 200% を指定して、元のフォントサイズに正確に戻ると期待してはいけません。

丸め誤差によって期待通りの結果が得られないからです (フォントサイズは整数としてのみ扱われるため)。

例えば、15 ポイントのテキストから始めて \Zr050 (50%) を使った場合、結果は 7 ポイントのテキストになります。

7 ポイントの 200% は、14 ポイントのテキストに過ぎません。

その代わりに、既知のフォントサイズに戻すには、Normal エスケープコード "\M"、絶対フォントサイズ、または呼び出し済みのフォントサイズを使ってください。



特別なエスケープコード

Special ポップアップメニューから項目を選択すると、その後に続く文字を上付き、下付き、または通常表示にするエスケープコードが挿入されたり、後続のテキストのスタイル、位置、色に影響を与えたり、グラフにウェーブを描画するための記号が挿入されたりします。

最初の 4 つの項目、Store Info、Recall Info、Recall X Position、Recall Y Position は、詳細な注釈を作成するために使われ、「テキスト情報変数のエスケープコード」のセクションで説明しています。

Style を選択すると、注釈テキスト入力領域の現在の挿入位置にある注釈のスタイル (太字、斜体など) を変更するためのサブダイアログが表示されます。このサブダイアログには、Recall Style チェックボックスがあり、テキスト情報変数を使った複雑な注釈の作成に使われます。

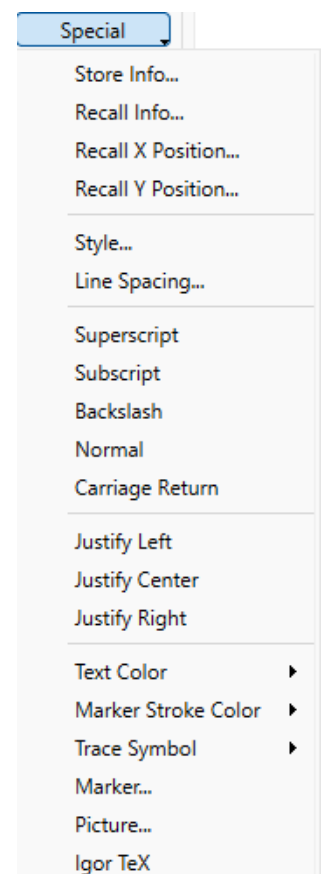
Superscript と Subscript は、その後に続く文字を上付きまたは下付きにするエスケープコードを挿入します。

テキストを元の文字サイズ、Y 位置に戻すには、Normal を使ってください。

Backslash は、エスケープコードとして機能するバックスラッシュではなく、そのまま表示されるバックスラッシュを挿入するためのコードを挿入します。バックスラッシュを表すエスケープコードである 2 つのバックスラッシュを挿入することで、これを実現します。

Normal は、元のフォントサイズとベースラインに戻るためのコードを挿入します。

より正確には、Normal はフォントサイズとベースラインを、テキスト情報変数



0 に格納されている値に設定します（「テキスト情報変数のエスケープコード」のセクションを参照）。
フォントやスタイルには影響しません。

Justify では、現在の行と次の行を揃えるためのコードが挿入されます。

Text Color を選択すると、その後のテキストに色を付けるためのコードが挿入されます。
初期のテキストの色と注釈の背景色は、Frame タブで設定します。

Trace Symbol は、グラフ上でトレースを表示するために使われるシンボル（線、マーカーなど）を出力するコードを挿入します。

このコードは、凡例に自動的に挿入されます。

このメニュー項目を使うと、タグ、テキストボックス、またはカラースケールにシンボルを手動で挿入することができます。

グラフの注釈の場合、サブメニューには上部のグラフにあるすべてのトレース名のインスタンスが一覧表示されます。

レイアウトの注釈の場合、レイアウト内のすべてのグラフにあるすべてのトレース名のインスタンスが一覧表示されます。

Marker は、マーカー記号を描画するためのコードを挿入します。
これらの記号は、グラフ内のトレースとは独立しています。

Igor TeX は、開始テンプレート $\backslash$WMTEX$ xxx \backslash$WMTEX$ を挿入するため、xxx を LaTeX の数式の一部に置き換えることができます。$

詳細については、「Igor TeX」のセクションを参照してください。

タグ用の動的エスケープコード

Dynamic ポップアップメニューは、タグにのみ適用されるエスケープコードを挿入します。

これらのコードは、タグが添付されているウェーブまたはウェーブ上のポイントに関する情報を挿入するように指示します。

ウェーブや添付ポイントが変更されるたびに、この情報を自動的に更新します。

動的項目

効果

Wave name タグが付けられているウェーブの名前を表示します

Trace name and instance ウェーブの名前と同じですが、特定のウェーブ名に関連付けられたグラフ内に複数のトレースが存在する場合、インスタンス番号（例：#1）が末尾に付加されます

Attach point number タグの取り付けポイントの番号を表示します

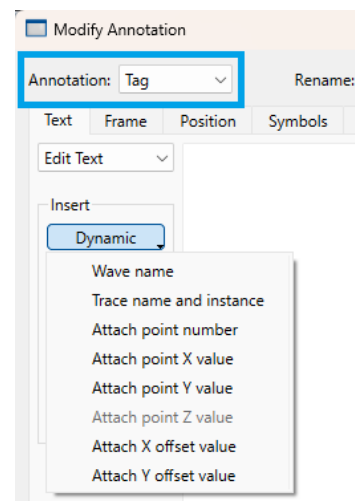
Attach point X value タグの取り付けポイントの X 座標を表示します

Attach point Y value タグの取り付けポイントの Y 座標を表示します

Attach point Z value タグの取り付けポイントの Z 値を表示します。コンタートレース、ウォーターフォールプロット、または画像プロットでのみ利用可能です

Attach X offset value トレースの X 方向のオフセットを表示します

Attach Y offset value トレースの Y 方向のオフセットを表示します



TagVal と TagWaveRef 関数も参照してください。

これらの関数は、Dynamic ポップアップメニュー項目と同じ情報を提供しますが、より柔軟性に富んでいます。

その他の動的エスケープコード

エスケープコードシーケンスを使うと、動的に評価されたテキストをあらゆる種類の注釈に挿入する動的テキストエスケープシーケンスを入力できます。

```
\{dynamicText}
```

ここで、*dynamicText* には数値や文字列の式が含まれる場合があります。

この方法については、「動的テキストのエスケープコード」のセクションで説明しています。

TagVal と TagWaveRef 関数

注釈がタグの場合、TagVal と TagWaveRef 関数を使って、そのタグが紐付けられているデータポイントに関する情報を表示することができます。

例えば、以下のコードは、タグのデータポイントの Y 値を表示します。

```
\{"%g", TagVal(2)}
```

これは、Dynamic ポップアップメニューから「Attach point Y value」を選択して挿入できるエスケープコード "%Y" と、実質的に同じ効果があります。

TagVal 関数を使う利点は、%g 以外の書式設定手法を利用できることです。

例えば：

```
\{"%5.2f", TagVal(2)}
```

TagVal は、ダイナミックメニューのエスケープコードを介してアクセスできるすべての情報を返すことができます。

テキストの数値形式をコントロールしたい場合に使ってください。

TagWaveRef 関数は、タグが添付されているウェーブの参照を返します。

この参照は、ウェーブ自体の名前を使うのと同じように使用できます。

たとえば、wave0 という名前のウェーブが表示されているグラフがある場合、次のタグテキストを使うと、そのウェーブの平均値が表示されます。

```
\{"%g", mean(wave0)}
```

これで問題ありませんが、タグを別のウェーブに移動しても、依然としてウェーブ 0 の平均値が表示されます。

TagWaveRef を使えば、タグが付けられたウェーブの平均値を表示させることができます。

```
\{"%g", mean(TagWaveRef())}
```

TagVal と TagWaveRef 関数は、注釈テキストの評価を行っている間のみ動作するため、これらの関数は注釈の動的テキスト内、または注釈の動的テキストから呼び出される関数内でのみ使ってください。

注釈のタブ

Text タブの Annotation テキスト領域には、実際には2つの機能があり、これらは左上隅にあるポップアップメニューでコントロールされます。

このポップアップメニューから Set Tabs を選択すると、注釈用のタブストップを表示します。

デフォルトでは、注釈には 1/2 インチ間隔で 10 個のタブストップが設定されています。

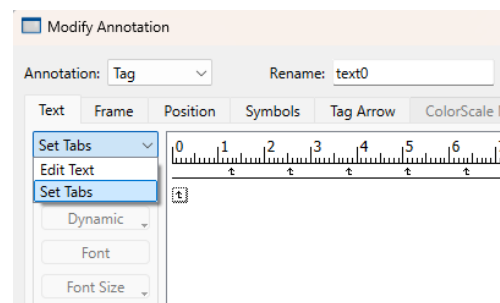
タブストップは、ルーラー上でドラッグすることで変更できます。

タブストップをルーラーの外側へドラッグして削除できます。

左側のタブ保存領域からルーラー上にドラッグすることで、タブを追加できます。

1 つの注釈につき最大 10 個のタブストップを設定でき、これらは常に左揃えのタブとなります。

1 つの注釈につきタブストップは 1 セットのみ設定可能で、その設定は注釈全体に適用されます。



一般的な注釈プロパティ

ほとんどの注釈プロパティは、あらゆる種類の注釈に共通しています。

注釈名

Name 項目を使って、注釈に名前を付けることができます。

Modify Annotation ダイアログでは、これは Rename 項目になります。

この名前は、Tag、TextBox、ColorScale、または Legend コマンドにおいて、注釈を識別するために使われます。

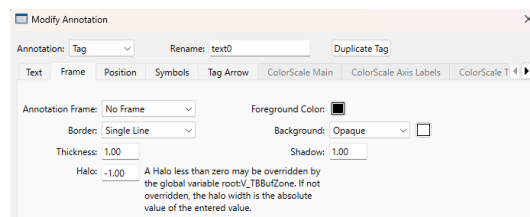
注釈の名前は、そのウィンドウ内で一意である必要があります。

詳細は、「注釈を使ったプログラミング」のセクションを参照してください。

注釈フレーム

Frame タブにある Frame と Border ポップアップメニューを使うと、注釈をボックスやシャドウボックスで囲んだり、テキストボックスに下線を引いたり、フレームを全く表示しないように設定したりできます。

フレームやシャドウの線の太さは、Thickness と Shadow の値で設定されます。



デフォルトでは、フレーム付き注釈には、周囲と区別するために、周囲を取り囲む 1 ポイントの「ハロー」も付いています。

このハローは、注釈の背景色と同じ色になります。

Frame タブの Halo ボックスで希望の太さを設定することで、このハローの幅を 0~10 ポイントの範囲で変更できます。

0.5 などの小数点以下の値も指定可能です。

Halo に負の値を指定すると、ルートデータフォルダ内のグローバル変数 V_TBBufZone によってハローの厚さが上書きされます。

この変数が存在しない場合は、入力された値の絶対値が使われます。

ハローのデフォルト値は -1 です。

IgorStartOrNewHook フック関数内で V_TBBufZone グローバル変数を設定することで、デフォルトのハロー値を上書きすることができます。

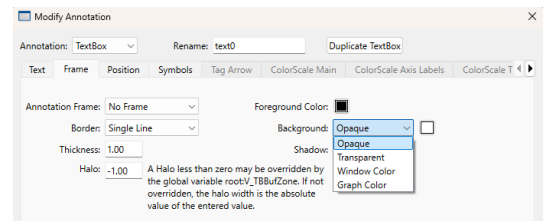
ヘルプ User-Defined Hook Functions の例を参照してください。

注釈の色

Frame タブには、注釈の色設定のほとんどが含まれています。

Foreground Color ポップアップメニューを使って、初期のテキスト色を設定します。

Text タブの Special ポップアップメニューから色エスケープコードを挿入することで、初期の前景色とは異なるテキスト色に変更することができます。



Background ポップアップメニューを使って、背景モードと色を設定します。

背景カラーモード	効果
Opaque	注釈の背景は、その背後にあるオブジェクトを覆い隠します。背景色はポップアップメニューから選択します
Transparent	注釈の背後のオブジェクトが透けて見えます
Graph color	背景は不透明で、グラフの背景色と同じ色になります。ページレイアウトウィンドウに追加された注釈では、この機能は利用できません
Window color	背景は不透明で、ウィンドウの背景色と同じ色になっています

注釈の配置

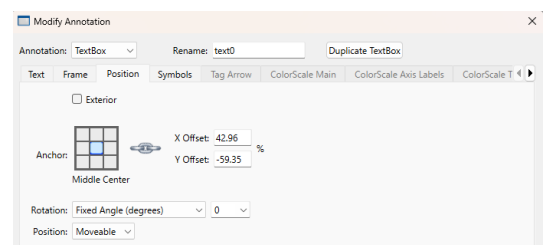
Position タブの Rotation ポップアップメニューを使うと、注釈を4つの主要な向きに回転させることができます。

また、整数度の単位で任意の回転角度を入力することもできます。

コンタートレースやカラースケールに付随するタグには、専用の回転設定があります。

ヘルプ Modifying Contour Labels と「カラースケールのサイズと向きの変更」のセクションを参照してください。

注釈はドラッグすることでウィンドウ内の任意の場所に配置できますが、多くの場合、これを知っておけば十分です。ただし、いくつかの細かい点に注意すれば、ウィンドウのサイズを変更したり、異なるサイズでウィンドウを印刷したりしても、注釈を正しい位置に配置することができます。



これは、グラフをページレイアウトウィンドウに配置する場合に特に重要です。

ページレイアウトウィンドウでは、配置されたグラフのサイズが通常、グラフウィンドウのサイズとは異なるためです。

注釈は、「アンカーポイント」からの X と Y オフセットを使って配置されます。

これらのオフセットやアンカーの意味は、注釈の種類や、ウィンドウがグラフ、レイアウト、あるいは Gizmo プロットのいずれであるかによって異なります。

例えば、タグは、グラフの水平および垂直サイズに対するパーセンテージとして表されるオフセットを使って配置されます。

「タグの配置」のセクションを参照してください。

グラフにおけるテキストボックス、凡例、カースケールの配置

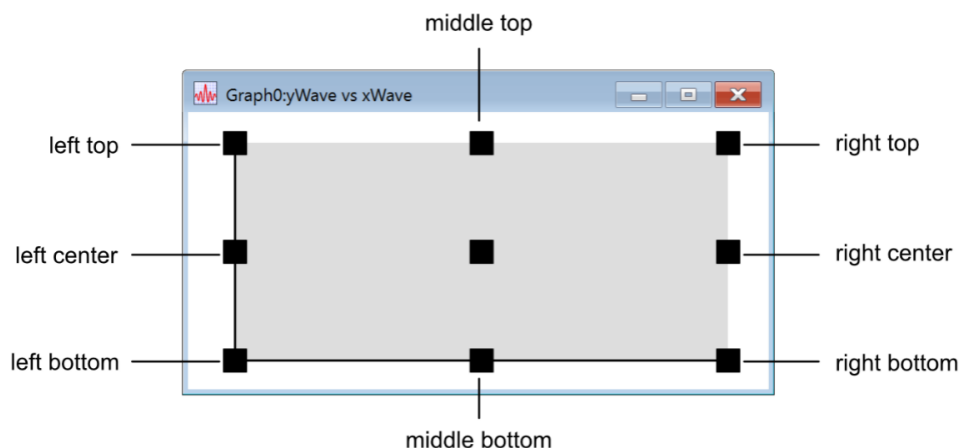
テキストボックス、凡例、カースケールは配置が同じであるため、この説明ではこれらすべてを総称して「テキストボックス」と呼びます。

グラフ内のテキストボックスは、グラフのプロット領域の「内側」または「外側」に配置することができます。この配置オプションは、Exterior チェックボックスを選択することで指定します。

Anchor ウィジェットは、プロット領域またはグラフウィンドウの端における基準点の正確な位置を指定します。また、Igor Pro がテキストボックスの「位置」とみなす、テキストボックス上の位置も指定します。

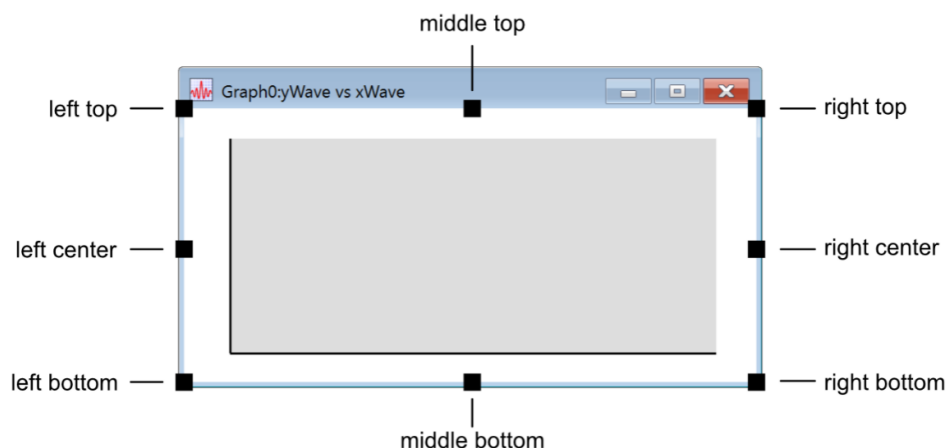
内部テキストボックスは、グラフのプロット領域の端にある基準点に対して相対的に配置されます。

(プロット領域とは、グラフウィンドウ内でトレースがプロットされる中央の長方形です。この長方形は、標準的な左軸、右軸、下軸、上軸に囲まれています。)



内部テキストボックスに対するアンカーポイント

外部テキストボックスは、ウィンドウの端にある基準点に対して相対的に配置され、通常、そのテキストボックスはプロット領域の外側にあります。



外部テキストボックスに対するアンカーポイント

外部テキストボックスの目的は、グラフのプロット領域から離れた場所にテキストボックスを配置できるようにすることです。

例えば、グラフの上部軸の上や、右軸の右側に配置したい場合があります。

グラフをテキストボックスから遠ざけることで、外部テキストボックスがグラフと重ならないようにします。

グラフをどの方向に押し出すかは、テキストボックスのアンカー位置によって決まります。

例えば、テキストボックスが上部にアンカーされている場合、グラフをテキストボックスから離れる方向、つまり下方向に押し出します。

アンカー位置が middle-center の場合、グラフをテキストボックスから離そうとはしません。

したがって、middle-center にアンカーされた外部テキストボックスは、内部テキストボックスと同じように振る舞います。

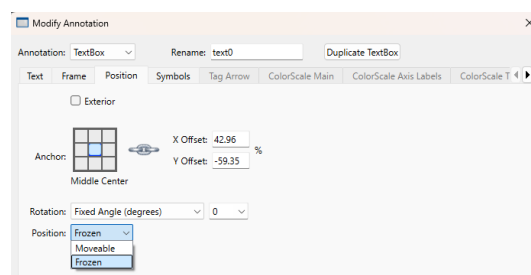
Modify Graph ダイアログで余白を指定すると、外側のテキストボックスの設定が上書きされ、外側のテキストボックスによってグラフが押し出されることはありません。

Position タブの XY Offset は、アンカーからテキストボックスまでの水平と垂直方向のオフセットを、内部テキストボックスの場合はグラフのプロット領域の水平と垂直サイズ、外部テキストボックスの場合はウィンドウサイズを基準としたパーセンテージで示します。

Position ポップアップメニューでは、位置を Moveable（移動可能）または Frozen（固定）に設定できます。

Frozen とは、テキストボックスの位置をマウスで移動できないことを意味します。

これは、テキストボックスを軸の目盛りのラベルとして使っており、誤って移動させてしまわないようにしたい場合に便利です。



ページレイアウトにおける注釈の配置

ページレイアウトウィンドウ内の注釈は、ページの印刷可能領域の端にあるアンカーポイントを基準に配置されます。

アンカーポイントからテキストボックスまでの距離は、印刷可能ページのパーセント単位で指定される X と Y オフセットによって決まります。

ページレイアウト内の注釈は、グラフの場合のように「固定」することはできません。

凡例

凡例はテキストボックスとよく似ています。

グラフやページレイアウト内のトレースの一部またはすべての記号を表示します。

凡例を作成するには、Graph または Layout メニューから Add Annotation を選択します。

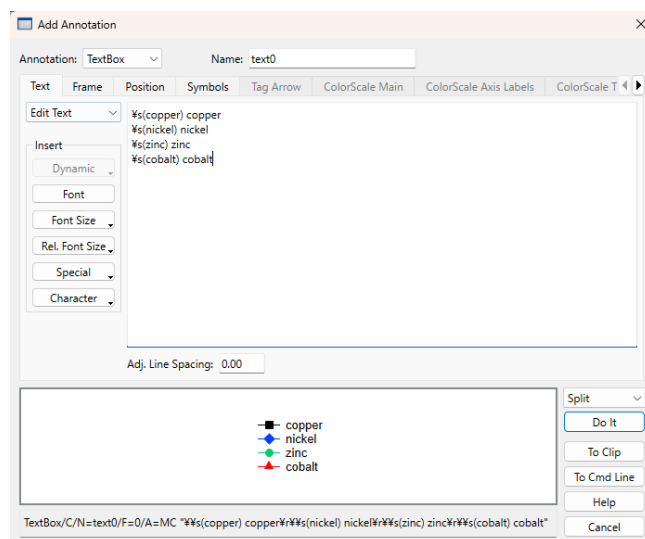
これにより、Add Annotation ダイアログが表示されます。

ダイアログの左上にあるポップアップメニューでは、注釈の種類（TextBox、Tag、Legend、ColorScale）を設定できます。

テキスト入力領域にテキストが入力されていない状態で「Legend」を選択すると、「標準の凡例」に必要なテキストを自動的に生成します。

標準の凡例のままにするには、Do It をクリックするだけです。

ただし、他の種類の注釈と同様に、凡例のテキストを編集することも可能です。



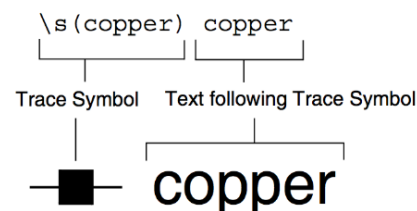
凡例テキスト

凡例テキストは、凡例に表示したいトレースのシンボルを指定するためのエスケープシーケンスと、プレーンテキストで構成されています。

上記のダイアログの例では、`\s(copper)` は、名前が「copper」のトレースのシンボル（線と塗りつぶされた四角形のマーカー）を挿入するエスケープシーケンスです。

このエスケープシーケンスの後に、スペースとウェーブの名前が続きます。

エスケープシーケンスの後の部分はプレーンテキストであり、必要に応じて編集することができます。



凡例シンボルに対してトレース名を指定する代わりに、トレース番号を指定することもできます。

例えば、`"\s(#0)"` と指定すると、トレース番号 0 の凡例が表示されます。

凡例とテキストボックスの違いは、2 つだけです。

まず、テキスト入力領域にテキストが入力されていない状態でポップアップメニューから Legend を選択すると、凡例のテキストが自動的に生成されます。

次に、グラフにウェーブを追加・削除したり、ウェーブの名前を変更したりすると、トレース記号の追加・削除によって凡例が自動的に更新されます。

テキストボックス、タグ、カラースケールでは、これら 2 つの動作はいずれも行われません。

トレース名の詳細は、ヘルプ Trace Names を参照してください。

ポイントにおけるシンボルの条件

トレース名の後に括弧で囲んだポイント番号を付加することで、トレースの特定のポイントにおける状態を示す凡例記号を作成できます。

例：`\s(copper[3])`

この機能は、トレースが f(z) モードを使っている場合や、トレース上の 1 つのポイントがカスタマイズされている場合に役立ちます。

凡例テキストの固定

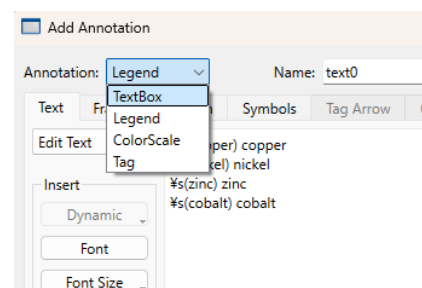
グラフにウェーブを追加した時、凡例が自動的に更新されないようにしたい場合があるかもしれません。

注釈をテキストボックスに変換することで、凡例のテキストを固定することができます。

更新されない凡例を作成するには、Add Annotation ダイアログを開きます。

ポップアップメニューから Legend を選択して Igor に凡例テキストを作成させ、次にポップアップメニューから TextBox を選択します。

これで凡例がテキストボックスに変換され、自動的に更新されなくなります。



マーカーサイズ

トレースにマーカーが設定されている場合、トレース記号にはそのマーカーが含まれます。

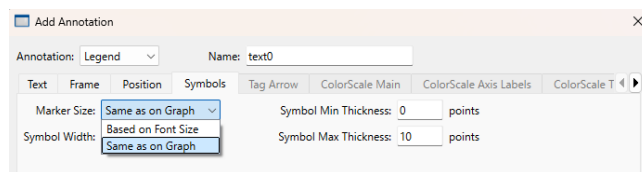
通常、注釈内に描画されるマーカーのサイズは、マーカーが描画される時点で有効なフォントサイズに基づいて決定されます。

トレース記号のエスケープコードの前にフォントサイズを設定すると、マーカーとそれに続くテキストの両方が比例して調整されます。

また、トレース記号の後にフォントサイズを変更することも可能です。

この場合、マーカーのサイズには影響を与えずに、それに続くテキストのサイズのみを設定することになります。

マーカーのサイズを設定する2つ目の方法は、Symbols タブの Marker Size ポップアップメニューから「Same as on Graph」を選択することです。そうすると、注釈のテキストフォントのサイズに関係なく、マーカーのサイズがグラフ内の対応するマーカーのサイズと一致します。



トレースシンボルの中央揃え

トレースシンボルの中には、そのエスケープコードの直前にあるテキスト、あるいは直後に続くテキストに対して垂直方向に中央揃えされるものもあれば、下端がベースラインに合わせて描画されるものもあります。

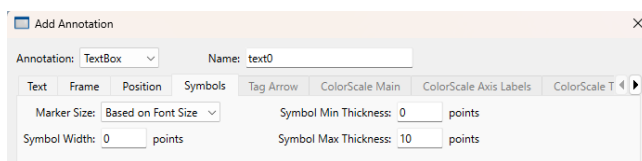
シンボルが中央揃えされるトレーススタイルには、ポイントとポイントをつなぐ線、ドット、マーカー、線とマーカー、シテイスケープなどがあります。

シンボルがベースラインから描画されるトレーススタイルには、ゼロからの線、ヒストグラムの棒、ゼロまでの塗りつぶし、スティックとマーカーなどがあります。

トレースシンボルの幅

トレースシンボルの幅とは、特定の凡例に含まれるすべてのトレースシンボルが描画される幅のことです。

この幅は、トレースシンボルの前にあるテキストのフォントサイズによってコントロールされるか、または Symbols タブの Symbol Width 値を使って、指定したポイント数に明示的に設定されます。



Symbol Width に 0 以外の幅の値を入力することで、シンボルの全体的なサイズを拡大または縮小できます。

小さなテキストに大きなマーカーを使う場合、この設定を使ってトレースシンボルの幅を縮小する必要がある場合があります。

長いダッシュ/ギャップパターンを持つ一部の線種については、そのパターンが表示されるように十分な大きさの値を明示的に入力することを推奨します。

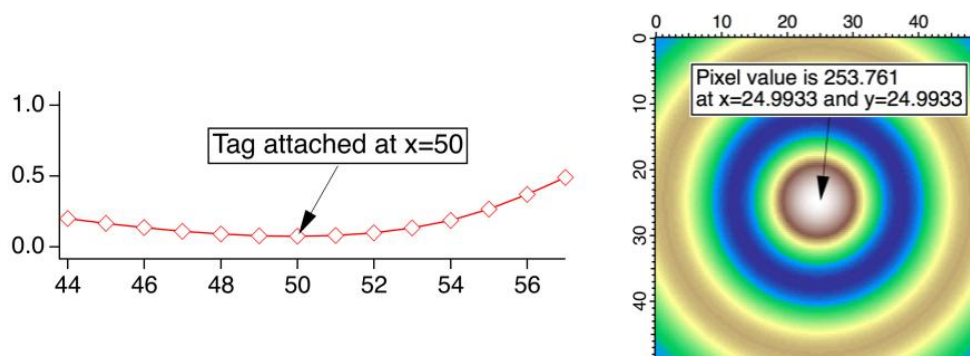
Color as f(z) を持つシンボル

Color as f(z) を使ったグラフを作成する場合、凡例を作成したいかもしれません。その方法は、ヘルプ Color as f(z) Legend Example を参照してください。

タグ

タグはテキストボックスに似ていますが、いくつかの機能が追加されています。

タグは、グラフ内の特定のトレース、画像、またはウォーターフォールプロット上の特定のポイントに付けられます。



ページレイアウトや Gizmo プロットにはタグを追加できません。

ただし、タグを含むグラフをページレイアウトに追加することは可能です。

Igor Pro は、コンタープロットにラベルを付けるためのタグを自動的に生成します。

タグテキスト

タグ内のテキストには、テキストボックスや凡例で扱える内容に加え、さらに多くの情報を含めることができます。

Text タブの Dynamic ポップアップメニューには、タグにのみ適用されるエスケープコードが登録されています。これらのコードは、タグが添付されているウェーブに関する情報、またはタグが添付されているウェーブ上のポイントに関する情報を挿入するように指示します。

この情報は、ウェーブや添付ポイントが変更されるたびに「動的に」更新されます。

「タグ用の動的エスケープコード」のセクションを参照してください。

TagVal 関数と TagWaveRef 関数は、動的なテキストを含むタグを作成する時にも役立ちます。

「TagVal 関数と TagWaveRef 関数」のセクションを参照してください。

タグウェーブとアタッチメントポイント

Position タブで、Tag on ポップアップメニューからウェーブを選択することで、タグをどのウェーブに付けるかを指定します。

タグをどのポイントに配置するかは、At p= の欄にポイント番号を入力するか、At x= の欄に X 値を入力することで指定します。

X 値は、タグを添付するウェーブの X スケールに基づいています。

ウェーブが XY モードで表示されている場合、これは必ずしもそのポイントの X 軸上の位置と同じではありません。

これは、タグが添付される Y ウェーブのポイントの X 値です。

この違いが分かりにくい場合は、ヘルプ The Waveform Model of Data を参照してください。

(2D) 画像やウォーターフォールプロットにおけるタグの取り付けポイントは、1D ウェーブの場合とは少し異なる扱いになります。

画像では、これは行列配列内で線形にインデックス付けされた順序番号です。

このダイアログでは、At p= 設定として入力されたこのポイント番号を X 座標と Y 座標に変換し、その逆の変換も行います。

実際のアタッチメントポイントはポイント番号によって決まるため、四捨五入の影響により、タグは入力された At x= と At y= の値の正確な位置に取り付けられるとは限りません。

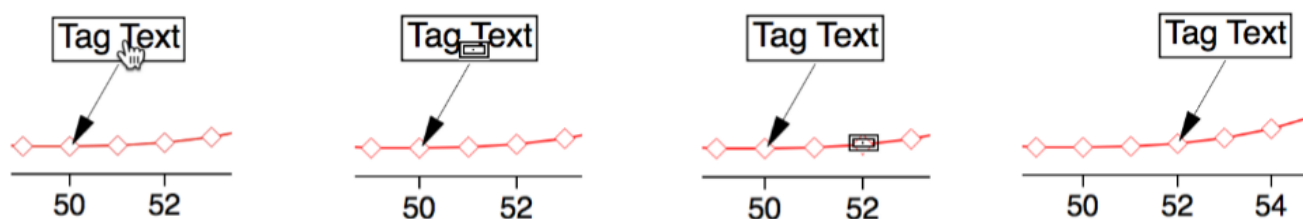
次のセクションで説明するように、タグをドラッグして手動で配置することができます。

タグのアタッチメントポイントを変更する

タグがグラフ上に配置された後、Alt キーを押しながらタグをクリックし、特別なタグカーソルをトレース上の新しいアタッチメントポイントまでドラッグすることで、別のポイントにタグを配置し直すことができます。

タグカーソルは、タグのテキストを表示させたい画面上の位置ではなく、トレース上のタグを配置したいポイントまでドラッグする必要があります。

タグカーソルの中央にあるポイントが、タグが配置される位置を示しています。



1. カーソルをタグテキストに移動

2. Alt キーを押す

3. 新しいアタッチメントポイントまでドラッグ

4. タグが新しいポイントにアタッチされる

タグのカーソルをグラフの外にドラッグすると、そのタグがグラフから削除されます。

タグの矢印

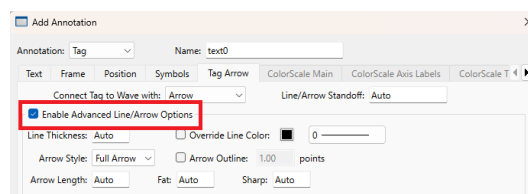
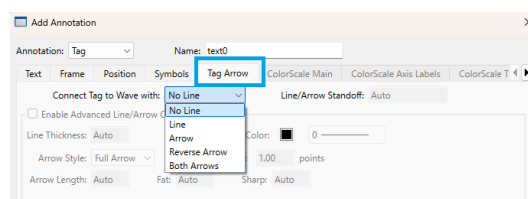
Tag Arrows タブにある Connect Tag to Wave with: ポップアップメニューを使うと、タグのアンカーポイントから矢印や線を引いて、タグのアタッチメントポイントを指定することができます。

Line/Arrow Standoff をポイント単位で設定することで、矢印や線がデータポイントにどれだけ近づくかを調整できます。

Advanced Line/Arrow オプションを使うと、線や矢印の特性をさらに細かくコントロールできます。

Line Thickness の値が 0 の場合、デフォルトの線幅である 0.5 ポイントになります。

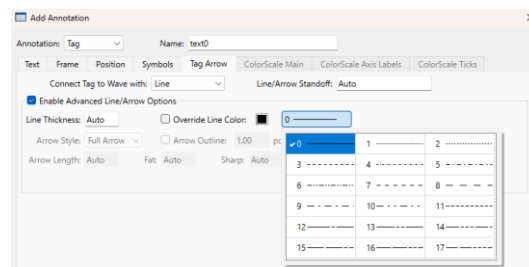
それ以外の場合は、10.0 ポイントまでの値を入力してください。



線を非表示にするには、Connect Tag to Wave with: ポップアップメニューから No Line を選択してください。

線の色は通常、注釈枠の色（Frame によって設定されます）。Override Line Color チェックボックスをオンにし、ポップアップメニューから色を選択することで、この設定を上書きすることができます。

線種のポップアップメニューを使って、添付線のスタイルをデフォルトの実線から変更します。



Connect Tag to Wave with: のポップアップが「Arrow」に設定されている場合、残りのコントロールを使って矢印の形状を調整できます。

オプションには、矢印の頭部を「フル」または「ハーフ」にするか、塗りつぶしまたは輪郭線にするか、矢印の長さ、太さ、鋭さなどが含まれます。

Arrow Length の設定はポイント単位で指定され、「0」または「自動」を選択するとデフォルトの長さが適用されます。

Sharp の設定は、-1.0 から 1.0 までの間の小さな値です。

0 または Auto に設定すると、デフォルトのシャープネスが適用されます。

Fat 設定は、幅と長さの比率を指定します。

0 または Auto に設定すると、デフォルトの比率である 0.5 が適用されます。

数値が大きいほど、矢印は太くなります。

数値が小さい場合（たとえば 0.1）、矢印の長さを長くしない限り、矢印が見えなくなってしまうことがあります。印刷された矢印は、画面上に表示される矢印よりも細く見える場合があります。

タグの線と矢印のスタンドオフ

Line/Arrow Standoff 設定を使うと、そのトレースに対して線や矢印をどの程度近づけるかを指定できます。

ポイント単位で具体的な距離を入力することもできます。

Auto または 0 を入力すると、出力デバイスの解像度とグラフのサイズに応じて距離を調整します。

値を 1 に設定すると、線はトレースに可能な限り近づきます。

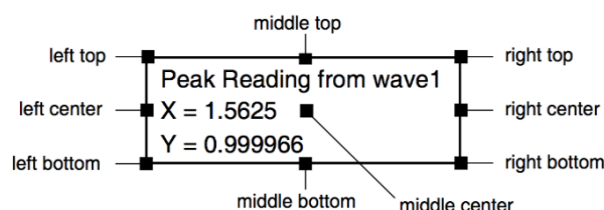
ウェーブがマーカー付きでグラフ化されている場合、線や矢印がマーカーと交差しないように、スタンドオフをマーカーのサイズよりも大きい値に設定することを推奨します。

タグのアンカーポイント

タグには、タグ自体上にアンカーポイントがあります。

矢印や線がある場合、それらはタグ上のアンカーポイントからトレース上のアタッチメントポイントへと描画されます。

また、アンカーの設定によって、タグの位置を表すタグ上の正確な位置も決まります。

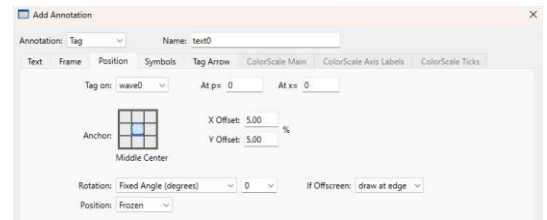


線は常にタグの後ろに引かれるため、アンカーポイントが middle center にある場合でも、線がテキストと重なることはありません。

タグの配置

タグの位置は、タグが取り付けられているポイントの位置と、Position タブの設定にある XY オフセットによって決まります。

XY オフセットは、アタッチメントポイントからタグのアンカーまでの水平方向および垂直方向の距離を、グラフのプロット領域の水平および垂直のサイズのパーセンテージで表したものです。



タグがグラフ上に配置されたら、ドラッグするだけでその XY オフセット、位置を変更することができます。Position タブの Position ポップアップメニューから「frozen」を選択することで、タグがドラッグされないようにすることができます。

コンターラベル用のタグを作成する時は、そのタグを自動的に固定します。

テキストボックスで使われる「内部/外部」の設定は、タグには適用されません。

オフスクリーンポイントに付けられたタグ

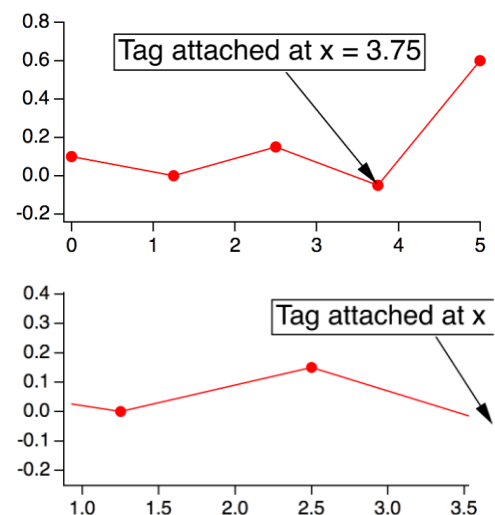
グラフにウェーブの一部しか表示されていない場合、タグのアタッチメントポイントがグラフ上に表示されないことがあります。

つまり、「画面外」または「表示範囲外」にあるということです。

これは通常、グラフが手動で拡大されたか、軸が自動スケール設定になっていないために発生します。

Igor Pro は、画面外にあるアタッチメントポイントに向かって接続線を描画します。

右のグラフでは、 $x=3.75$ にあるアタッチメントポイントは、表示されている X 軸の値の範囲内に収まっています。



グラフの X 軸の範囲を拡大して、 $x=3.75$ のタグのアタッチメントポイントを除外すると、タグのアタッチメントポイントは画面外になりますが、タグ自体は依然として描画されます。

Position タブの Offscreen ポップアップメニューから「hide the tag」を選択すると、アタッチメントポイントが画面外にあるタグの描画を非表示にできます。

非表示になっているタグを表示または変更したい場合は、グラフを自動拡大縮小して、そのタグが非表示にならないようにしてください。

コンターラベルはタグです

Igor Pro では、コンタープロットの数値ラベルを作成するために専用のタグを使います。

この特化機能により、「接線」機能が追加され、コンタープロットの経路に沿ってタグの向きが自動的に調整されま

す。
詳細は、ヘルプ Contour Labels を参照してください。

カラスケール

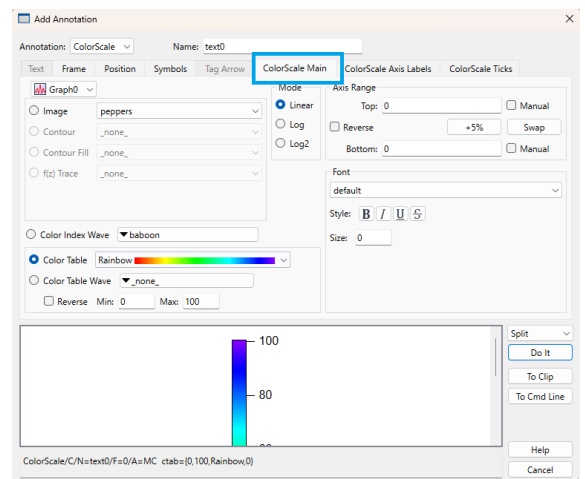
カラスケールによる注釈は、カラーバーと 1 つ以上の軸を使ってデータの範囲を要約するものです。

カラスケールの作成や変更は、Add Annotation ダイアログと Modify Annotation ダイアログから呼び出せる ColorScale コマンドを使って行います。

ColorScale Main タブ

カラスケールは、グラフ内の $f(z)$ トレース、画像プロット、コンタープロット、Gizmo プロット内の最初のサーフェスオブジェクト、あるいは任意のカラーインデックスウェーブやカラーテーブルに関連付けられています。この関連付けは、ColorScale Main タブで変更できます。

ポップアップメニューからグラフまたは Gizmo のプロットを選択すると、別のグラフ内の画像、コンタープロット、または $f(z)$ トレース、あるいは別の Gizmo ウィンドウ内のサーフェスプロットに基づいて、カラスケールを作成することができます。



カラスケールのサイズと向き

カラスケールのサイズと向きは、ダイアログの Position タブで設定します。

カラスケールのサイズは、注釈内の「カラーバー」のサイズと向き、さまざまな軸や目盛ラベルのパラメーターによって間接的にコントロールされます。

注釈は、カラーバー、目盛ラベル、軸のラベルを収めるように自動的に調整されます。

Auto または 0 に設定すると、Width と Height の設定により、カラスケールの軸方向に沿って、ウィンドウに合わせてカラスケールのサイズが自動的に調整されます。

水平方向のカラスケールは水平方向に自動的にサイズ調整されますが、垂直方向には調整されません。

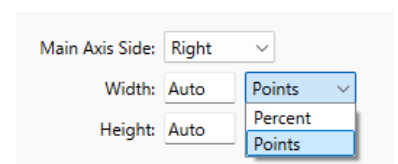
その逆も同様です。

カラーバーの長い辺は、グラフのプロット領域または Gizmo のプロットサイズの 75% に自動的に維持されます。短い辺は 15 ポイントに設定されます。

どちらのスケール軸についても、パーセントまたはポイント単位でカスタム設定を入力できます。

メニューから Percent を選択すると、グラフのサイズが変更された時に、対応する軸のサイズを自動的に調整します。

Points を選択すると、軸のサイズが固定され、変更されることはありません。



ColorScale Axis Labels タブ

ColorScale Axis Label タブで、主軸と（ある場合は）副軸の軸ラベルを設定します。

軸ラベルのテキストは 1 行に制限されます。

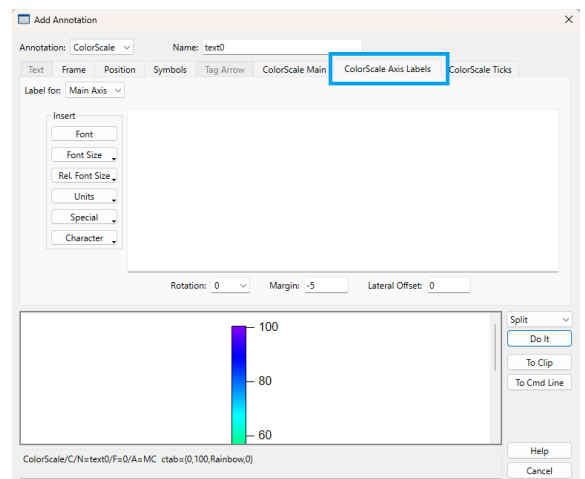
このテキストは、Text タブのテキストボックス、凡例、タグで使われるものと同じですが、Annotation ポップアップメニューが ColorScale に変更されると、1 行に切り詰められます。

Units ポップアップメニューでは、カラースケールが関連付けられている項目のデータ単位に関連するエスケープコードが挿入されます。

画像、コンタープロット、または表面プロットの場合、これらのコードは、画像、コンタープロット、または表面行列のデータ単位、あるいは XYZ コンターの Z データウェーブのデータ単位に関連しています。

Rotation、Margin、Lateral Offset は、色軸に対する軸ラベルの向きと位置を調整します。

2 番目の軸ラベルは、Color Scale Ticks タブで、ユーザーが指定した目盛りの値とラベルのウェーブに基づいて 2 番目の軸が作成された場合にのみ有効になります。



ColorScale Ticks タブ

カラースケールの軸目盛りの設定は、グラフの軸の設定と似ています。

Automatic Ticks にチェックを入れると、主軸の目盛りが自動的に計算されます。

また、User Ticks from Waves にチェックを入れると、目盛りを手動でコントロールすることができます。

後者の場合、2 つのウェーブを指定する必要があります。

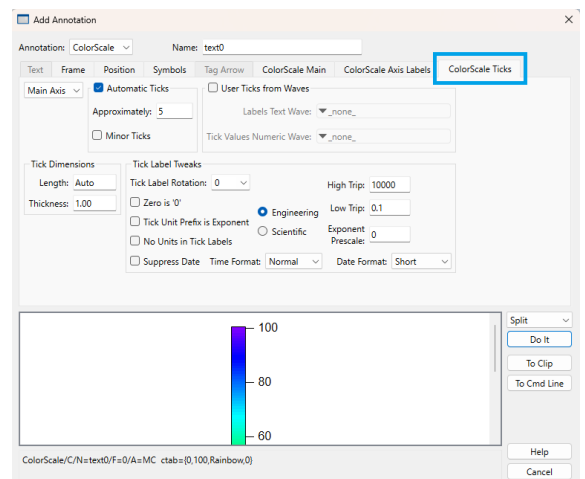
1 つは目盛りの位置を指定する数値ウェーブ、もう 1 つはそれに対応する目盛ラベルを指定するテキストウェーブです。

また、ユーザー定義の目盛ラベルや値を指定して、2 つ目の軸を作成することもできます。

これは、例えば、温度範囲を華氏と摂氏の両方で表示する場合などに便利です。

これを行うには、通常 Main Axis と表示されているポップアップメニューから Axis 2 を選択します。

2 つ目の軸は、カラーバーを挟んで主軸の反対側に描画されます。



Tick Dimensions 設定は、両方の軸に適用されます。

長さが -1 の場合は Auto となります。

それ以外の場合は、次元はポイント単位で表示されます。

目盛を非表示にするには、Thickness を 0 に設定してください。

精巧な注釈

Igor Pro では、下付き文字、上付き文字、数式記号などを使った精巧な注釈を作成することができます。

これを行うには、エスケープコードを入力する必要があります。

Add Annotation ダイアログには、必要となる可能性のある多くのエスケープコードを挿入するためのメニューが用意されています。

これは、軸ラベルやグラフのラベルなどに使うような、比較的単純な数式表記であれば可能です。

複雑な数式の場合は、Igor TeX 数式を使うか、本格的な数式エディタを使って、数式を表す画像を作成し、それをグラフやページレイアウトに貼り付ける必要があります。

テキスト情報変数

テキスト情報変数とは、単なるフォント、フォントサイズ、スタイルの変更よりも高度な「知能」を備えたエスケープコードを挿入できるようにする仕組みです。

テキスト情報変数を使えば、根気さえあれば、かなり凝った注釈を作成することができます。

これらは凝ったことを行う場合にのみ知っておく必要があるため、単純な注釈で満足できるのであれば、この「テキスト情報変数」に関するセクションの残りの部分は読み飛ばしてください。

テキスト情報変数は、注釈内の特定の「位置」（テキストの挿入位置）に関する情報を保存します。

具体的には、その位置のフォント、フォントサイズ、書体（太字、斜体など）、水平・垂直位置を保存します。

各注釈には、0 から 9 までの番号が付けられた 10 個のテキスト情報変数があります。

注釈のテキストにエスケープシーケンスを埋め込むことで、挿入位置に関する情報を特定の変数に保存することができます。

後で、エスケープシーケンスを埋め込むことで、その情報の一部またはすべてを呼び出すことができます。

Label Axis と Add Annotation ダイアログの Font、Font Size、Special ポップアップメニューには、この操作を行うための項目があります。

使用可能なエスケープシーケンスの一覧については、「テキスト情報変数のエスケープコード」のセクションを参照してください。

テキスト情報変数の例

これを理解するために、簡単な例を見てみます。

化学化合物「エタノール」の化学式 $\text{CH}_3\text{CH}_2\text{OH}$ を表示するテキストボックスを作成したいとします。

この数式を 24 ポイントのフォントで表示するテキストボックスを作成するには、Add Annotation ダイアログの Text タブに、通常のテキストとエスケープコード（赤で表示）を組み合わせた以下の内容を入力する必要があります：

$\backslash z24 \backslash [0\text{CH} \backslash \text{B3} \backslash \text{MCH} \backslash \text{B2} \backslash \text{MOH}$

エスケープコードは、直接入力するか、ダイアログの Insert セクションにあるポップアップメニューから選択することで入力できます。

例えば、フォントサイズのエスケープコード「 $\backslash z24$ 」は Font Size ポップアップメニューを使って生成され、その他のエスケープコードは Special ポップアップメニューを使って生成できます。

エスケープコードの意味は以下の通りです：

$\backslash z24$ フォントサイズを 24 ポイントにする

\[0	現在の状態をテキスト情報変数 0 として取得する（テキスト情報変数 0 には「通常」の状態が格納されている）
\B	下付き文字
\M	通常の状態に戻す（テキスト情報変数 0 に保存されている状態）
\B	下付き文字
\M	通常の状態に戻す（テキスト情報変数 0 に保存されている状態）

これを入力する方法の1つは、まず通常のテキストを入力し、その後でエスケープコードを追加することです。このプロセスの各ステップで、注釈のプレビューには次のように表示されます。

CH3CH2OH	CH3CH2OH
\z24 CH3CH2OH	CH3CH2OH （ただし、24 ポイント）
\z24\[0 CH3CH2OH	CH3CH2OH （見かけ上の変化なし）
\z24\[0CH\B3 CH2OH	CH ₃ CH2OH
\z24\[0CH\B3\MCH2OH	CH ₃ CH2OH
\z24\[0CH\B3\MCH\B2OH	CH ₃ CH ₂ OH
\z24\[0CH\B3\MCH\B2\MOH	CH ₃ CH ₂ OH

注釈を使ったプログラミング

TextBox、Tag、Legend、ColorScale コマンドを使って、注釈の作成、変更、削除を行うことができます。AnnotationInfo 関数は、既存の注釈 1 つに関する情報を返します。AnnotationList 関数は、既存の注釈の名称のリストを返します。

注釈名の変更

各注釈には、それが含まれているウィンドウ内で一意の名前が付けられています。TextBox、Tag、Legend、ColorScale、AnnotationInfo ルーチンの各々にこの名前を指定することで、変更したい注釈を特定します。

コマンドを実行する時に、/C/N=旧名/R=新名 という構文を使うことで、注釈の名前を変更できます。例えば：

```
TextBox/C/N=oldTextBoxName/R=newTextBoxName
```

注釈タイプの変更

注釈のタイプを変更するには、その注釈に適切なコマンドを適用します。例えば、タグや凡例をテキストボックスに変更するには、次のようにします。

```
TextBox/C/N=annotationName
```

注釈テキストの変更

既存の注釈のテキストを変更するには、「/N=注釈名」を使ってその注釈を指定し、新しいテキストを指定します。例えば、text0 という名前のテキストボックスに新しいテキストを指定するには、次のように入力します。

```
TextBox/C/N=text0 "This is the new text"
```

注釈にテキストを追加するには、AppendText コマンドを使います。

```
AppendText/N=text0 "and this text appears on a new line"
```

/NOCR フラグを使うと、改行を入れずにテキストを追加することができます。

プログラムによるテキストの生成

プロシージャを作成し、解析や計算の結果から生成されたテキストを使って、注釈を作成または更新することができます。

例えば、以下の関数は、上部のグラフまたはレイアウトウィンドウ内のテキストボックスを作成または更新するものです。

このテキストボックスの名前は「FitResults」です。

```
Function CreateOrUpdateFitResults(slope, intercept)
    Variable slope, intercept
    String fitText
    sprintf fitText, "Fit results: Slope=%g, Intercept=%g", slope, intercept
    TextBox/C/N=FitResults fitText
End
```

この関数は、直線へのフィッティングを行う CurveFit コマンドを実行した後、CurveFit コマンドによって返された係数を引数として渡して呼び出します（別の関数内から呼び出す場合もあります）。

注釈の削除

アノテーションをプログラムで削除するには、次のようにします。

```
TextBox/K/N=text0
```

注釈のエスケープコード

注釈エスケープコードを使うと、テキストボックス、タグ、凡例、カラースケールなどの注釈において、書式設定やその他の機能をコントロールできます。

また、軸ラベル、コントロールのタイトル、styledText キーワードを使った SetVariable の値、ListBox コントロールの内容、DrawUserShape コマンドでも使用できます。

これらのエスケープコードを使うと、テキストのフォント、サイズ、書式、色をコントロールしたり、上付き文字や下付き文字を作成したり、動的に更新されるテキストを作成したり、凡例記号を挿入したり、その他の効果を適用したりすることができます。

汎用のエスケープコード

これらのエスケープコードは、注釈エスケープコードに対応しているあらゆるテキストで使用できます。

<code>\B</code>	下付き文字
<code>\F'fontName'</code>	指定されたフォントを使う（例： <code>\F'Helvetica'</code> ）
<code>\fdd</code>	<code>dd</code> はビット単位のパラメーターで、各ビットは次のようにフォントスタイルの1つの側面をコントロールする Bit 0： 太字 Bit 1： 斜体 Bit 2： 下線 Bit 4： 取り消し線 ビット設定の詳細は、ヘルプ <code>Setting Bit Parameters</code> を参照してください。
<code>\JR</code>	右揃えテキスト
<code>\JC</code>	中央揃えテキスト
<code>\JL</code>	左揃えテキスト
<code>\K(r,g,b)</code>	テキストには指定された色を使う。<code>r</code>、<code>g</code>、<code>b</code> は 0 から 65535 までの整数 必要に応じて、0 から 65535 の範囲で不透明度を指定する4番目のアルファパラメーターを含めることができる <code>\K</code> は、<code>\W</code> によって追加されるマーカーの塗りつぶし色も設定する。マーカーの線色を設定するには、<code>\k</code> を使う
<code>\KB(r,g,b)</code>	テキストの背景色として指定された色を使う。<code>r</code>、<code>g</code>、<code>b</code> は 0 から 65535 までの整数 必要に応じて、0 から 65535 の範囲で不透明度を指定する4番目のアルファパラメーターを含めることができる <code>\KB0;</code> を使うと、背景色がオフになる <code>\KB</code> は Igor Pro 9.0 で追加された
<code>\k(r, g, b)</code>	マーカーのストローク（線の色）に指定した色を使う。<code>r</code>、<code>g</code>、<code>b</code> は 0 から 65535 までの整数。マーカーのストローク色をデフォルトの黒（0,0,0）から変更するには、<code>\Wtdd</code> の前にこの指定を記述する 必要に応じて、0 から 65535 の範囲で不透明度を指定する4番目のアルファパラメーターを含めることができる <code>\W</code> で追加されたマーカーの塗りつぶし色を設定するには、<code>\K</code> を使う
<code>\Ldtss</code>	テキスト情報変数 <code>d</code> で指定された <code>x</code> 位置から、現在の <code>x</code> 位置まで線を引く。現在のテキスト色を使う。線の太さは 4、5、6、7 の値を持つ数字 <code>t</code> で指定され、それぞれ 0.25、0.5、1.0、1.5 pt になる。線のスタイルは2桁の数字 <code>ss</code> で指定される
<code>\M</code>	通常の（メインの）文字体を使う（メインの行とサイズに戻る）

`\$PICT$name=pictName$/PICT$`

指定した画像を挿入する。`pictName` には、`ProcPict` またはメニュー `Misc→Pictures` ダイアログに表示されている画像名を指定できる

`\$WMTEX$ formula \$/WMTEX$`

LaTeX のサブセットを使って数式を挿入する。詳細は、ヘルプ `Igor TeX` を参照

`\S` 上付き文字

`\sa+dd` 行の上部に余白を追加する。`dd` は、ハーフポイント (1/144 インチ) 単位の 2 桁の数値。行内のどこにでも指定できる

`\sa-dd` 行の上部の余白を縮小する。`dd` は、ハーフポイント (1/144 インチ) 単位の 2 桁の数値。行内のどこにでも指定できる

`\sb+dd` 行の下部に余白を追加する。`dd` は、ハーフポイント (1/144 インチ) 単位の 2 桁の数値。行内のどこにでも指定できる

`\sb-dd` 行の下部の余白を縮小する。`dd` は、ハーフポイント (1/144 インチ) 単位の 2 桁の数値。行内のどこにでも指定できる

`\Wtdd` 現在のフォントサイズと色を使って、マーカー記号を描画する

`\Wtddd` 現在のフォントサイズと色を使って、マーカー記号 `ddd` を描画する

マーカーの輪郭線の太さは、1 桁の数字 `t` で指定されり。1、4、5、6、7、8 は、それぞれ 0.0、0.25、0.5、1.0、1.25、1.5 ポイントに相当する。`t` の値が 1 の場合、輪郭線の太さは 0 になるが、これは塗りつぶされたマーカーにのみ有用であり、塗りつぶされていないマーカーは表示されなくなる

マーカー記号番号は、2 桁の数字 `dd` または 3 桁の数字 `ddd` で指定される。新しいコードでは、3 桁のマーカーを使う

3 桁の構文は、カスタムマーカーをサポートするために Igor Pro 6.30 で追加された。この追加により、2 桁の `dd` エスケープシーケンスの直後に数字が続くと、誤って解釈されるようになった。例えば、次のようなシーケンス：

`\W718500 m`

は本来、マーカー 18 の後に “500 m” を表示するはず。しかし現在は、3 桁のマーカー番号 (185) の後に “00 m” が続くものとして解釈される。これを修正するには、2 桁のマーカー番号の先頭に 0 を追加して、3 桁のマーカー番号に変更する。例えば、次のようにする：

`\W7018500 m`

マーカーの線色を設定するには、`\k` を使う。マーカーの塗りつぶし色を設定するには、`\K` を使う

`\x+dd` 現在の `x` 座標を、現在のフォントの最大幅の $2*dd$ パーセント分だけ右に移動する

`\x-dd` 現在の `x` 座標を、現在のフォントの最大幅の $2*dd$ パーセント分だけ左に移動する

`\y+dd` 現在の `y` 座標を、現在のフォント高さの $2*dd$ パーセント分だけ上に移動する

`\y-dd` 現在の `y` 座標を、現在のフォント高さの $2*dd$ パーセント分だけ下に移動する

`\Znn` フォントサイズ `nn` を使う。`nn` は必ず 2 桁の数字でなければならない

`\Zrnnnn` `nnn` は、現在のフォントサイズを変更する割合 (%) を表す 3 桁の数値。`nnn` は必ず 3 桁でなければならない

タグのエスケープコード

これらのエスケープコードは、タグ内でのみサポートされています。

\ON	タグが付けられているウェーブの名前を挿入する
\On	トレース名と、0 より大きい場合はそのインスタンス番号を挿入する
\OP	タグが添付されているポイント番号を挿入する
\OX	タグが添付されているポイントの X 値を挿入する
\OY	タグが添付されているポイントの Y 値を挿入する
\OZ	コンタートレースの場合、タグが添付されているポイントの Z 値を挿入する。その他のトレースの場合は NaN を挿入する

テキスト情報変数のエスケープコード

テキスト情報変数は、Igor Pro によって作成される Igor Pro 独自の内部構造です。

このセクションで説明するエスケープコードを使うことで、テキスト情報変数を操作し、洗練された注釈を作成することができます。

これらのエスケープコードは、注釈用エスケープコードをサポートするあらゆるテキストで使用できます。

各注釈は、0 から 9 までの番号が付けられた 10 個のテキスト情報変数とともに作成されます。

各テキスト情報変数は、その注釈を含むグラフまたはページレイアウトのデフォルトのフォント、フォントサイズ、スタイルで初期化されます。

X 座標と Y 座標は、初期状態では未定義です。

背景情報や具体例については、「テキスト情報変数」のセクションを参照してください。

以下は、テキスト情報変数でサポートされているエスケープコードです。

<digit> は 0、1、2、…、9 のいずれかを意味します。

\[<digit>	フォント、サイズ、スタイル、現在の X 座標と Y 座標をテキスト情報変数に保存する
\]<digit>	テキスト情報変数から X 座標と Y 座標を除くすべての値を復元する
\X<digit>	テキスト情報変数から X 座標を復元する
\Y<digit>	テキスト情報変数から Y 座標を復元する
\F]<digit>	テキスト情報変数からフォントを復元する
\Z]<digit>	テキスト情報変数からフォントサイズを復元する
\f]<digit>	テキスト情報変数からスタイルを復元する

テキスト情報変数 0 には特別な性質があります。

これは「メイン」のフォントサイズを定義しており、\M エスケープシーケンスを使うことでこのサイズが復元されます。

また、\M はベースラインの Y 方向のオフセットを 0 に設定します。

\M によって、テキスト情報変数 0 のその他の設定（フォント、スタイル、オフセット）が設定されることはありません。

動的テキストエスケープコード

動的テキストを使うと、自動的に更新される注釈を作成できます。
動的テキストを挿入するための構文は次のとおりです。

```
\{dynamicText}
```

dynamicText には、グローバル変数やウェーブを参照し、ユーザー定義関数を呼び出す数値式や文字列式を含めることができます。

dynamicText 内で直接、あるいは関数呼び出しを通じて間接的に参照されているグローバル変数やウェーブが変更されると、自動的に dynamicText を再評価します。

また、注釈が更新されるたびに、dynamicText を再評価します。

注意

動的テキストを使うと、不安定で分かりにくい依存関係が生じることがあります。ほとんどの場合、「プログラムによるテキストの生成」のセクションで説明しているように、プログラムで静的テキストを生成するほうが望ましいです。

注記

dynamicText 内でローカル変数を参照することはできません。これは、ローカル変数はプロシージャの実行中にのみ存在し、それ以外の時点ではその変数を再評価する必要があるためです。

数値式および文字列式は、ルートデータフォルダーのコンテキストで評価されます。

式内では、ルート以外のオブジェクトについては、データフォルダーのフルパスを使ってください。

dynamicText には2つの形式があります。

1つは1つの数値式用の簡単な形式、もう1つは結果の書式設定を細かくコントロールできる、より複雑な形式です。

簡単な形式は次のようになります。

```
\{numeric-expression}
```

これは式を評価し、汎用の ("%g") 形式で出力します。

例えば：

```
TextBox "Two times PI is \{2*PI}"
```

は、次のテキストを持つ textbox を作成します。

```
Two times PI is 6.28319
```

TextBox のリテラル文字列パラメーターで2つのバックスラッシュを使う方法については、「アノテーションのエスケープシーケンスにおけるバックスラッシュ」のセクションで説明しています。

完全な形式は次のようになります。

```
\{formatStr, list-of-numeric-or-string-expressions}
```

formatStr と list-of-numeric-or-string-expressions は、printf コマンドの場合と同様に扱われます。
例えば、次の例には、フォーマット文字列、数値式、文字列式が含まれています。

```
TextBox "\{\\"Two times PI is %1.2f, and today is %s\\", 2*PI, date()}"
```

これは次の結果を生成します。

```
Two times PI is 6.28, and today is Thu, April 9, 2015
```

フォーマット文字列や数値式、文字列式では、その他の注釈エスケープコードを使うことはできません。
これらは `\{ ... }` のコンテキスト内では機能しません。

また、フォーマット文字列と文字列式では、複数行のテキストはサポートされていません。
複数行のテキストを使う必要がある場合は、「プログラムによるテキストの生成」のセクションで説明している手法を使ってください。

式を入力する時の補助として、中括弧の間の改行をスペースと同等とみなします。
Add Annotation ダイアログでは、次のように入力するのではなく：

```
\{"Two times PI is %1.2f, and today is %s",2*PI,date() }
```

次のように入力することができます：

```
\{
    "Two times PI is %1.2f, and today is %s",
    2*PI,
    date()
}
```

凡例記号のエスケープコード

注釈に凡例記号を挿入することができます。

グラフに記号を挿入するための構文は次のとおりです：

```
\s(traceName)
```

ページレイアウトにシンボルを挿入するための構文は次のとおりです：

```
\s(graphName.traceName)
```

`\s` は通常、記号が自動的に作成・削除される凡例で使われますが、記号は更新されるものの、自動的に追加・削除されないタグ、テキストボックス、軸ラベルでも使用できます。
「凡例」のセクションを参照してください。

軸ラベルのエスケープコード

以下のエスケープコードが、軸ラベルおよび軸タグでサポートされています。

- | | |
|-----------------|--|
| <code>\c</code> | その軸をコントロールするウェーブの名前を挿入する。これは、その軸に対してプロットされた最初のウェーブ |
| <code>\E</code> | 先頭に「x」を付けて、10 の累乗によるスケール変換を挿入する。これは曖昧になる可能性があるため、 <code>\U</code> または <code>\u</code> のいずれかを使うことを推奨 |
| <code>\e</code> | <code>\E</code> と同様だが、指数符号を反転させる。これは曖昧さを生じる可能性があるため、 <code>\U</code> または <code>\u</code> のいずれかを使うことを推奨 |
| <code>\U</code> | 接頭辞を自動的に付加して単位を挿入する |
| <code>\u</code> | 10 進スケール変換を挿入するが、 <code>\E</code> で使われるような先頭の「x」は付かない。軸がスケール変換されていない場合は何も行わない。カスタム単位や複合単位の文字列の前に使う。
ラベルの例：「Field Strength (<code>\u</code> Volts/Meter)」とすると、「Field Strength (10 ⁶ Volts/Meter)」のような表示になる |

- \u#1 これは \u の変種であり、\u の逆数（例： 10^6 の代わりに 10^{-6} ）を挿入する
- \u#2 単位やスケーリングの自動挿入を防止する。通常、Change Wave Scaling ダイアログや SetScale コマンドを使ってウェーブの単位やスケーリングを設定し、軸ラベルを明示的に指定しない場合、単位とスケーリングに基づいて軸ラベルを自動的に生成する。\\u#2 を使うと、この動作が邪魔になる場合に、それを抑制することができる

エスケープコードは大文字と小文字が区別される点に注意してください。

\\u と \\U では、挿入される部分文字列が異なります。

注釈のエスケープシーケンスにおけるバックスラッシュ

注釈のエスケープコードは、バックスラッシュ文字で始まります。

Igor Pro の文字列では、バックスラッシュ自体がエスケープ文字であるため、リテラル文字列にエスケープコードを入力する時は、1つのエスケープコードを表すために2つのバックスラッシュを入力する必要があります。

例えば：

```
TextBox/C/N=text0 "\\Z14Bigger"
```

Igor Pro コマンドパーサーは、「\\」を1つのバックスラッシュに変換します。

TextBox コマンドでは、この1つのバックスラッシュを検出し、\\Z をフォントサイズを変更したいという指示として解釈します。

次の例は、二重バックスラッシュを使いたくない場合を示しています。

```
TextBox/C/N=text0 "First line\rSecond line"
```

Igor Pro コマンドパーサーは、「\\r」をキャリッジリターン文字に変換します。

TextBox コマンドは、このキャリッジリターン文字を検出し、改行を開始したいという意図があると解釈します。この例には注釈用エスケープコードは含まれておらず、通常のエスケープコードのみが使われているため、二重バックスラッシュは使わないでください。

Add Annotation ダイアログは、注釈を作成するためのコマンドを生成します。

このダイアログは、バックスラッシュのルールを知っています。

したがって、Add Annotation ダイアログを使って注釈を作成する場合、生成されたコマンドを確認することで、バックスラッシュの正しい使用法を確認することができます。

Igor TeX

Igor Pro 8 以降では、LaTeX 構文の一部を使うことで、従来なら面倒なエスケープコードや、他のプログラムで生成した画像の挿入が必要だった数式を簡単に挿入できます。

LaTeX 式は、以下に示すように「\\\$WMTEX\$」と「\\\$/WMTEX\$」で囲みます。

```
Display/W=(35,45,339,154)
```

```
TextBox/C/N=TeXTest/A=MC "\\Z18" + "\\$WMTEX$ \\frac{3x}{2} \\$/WMTEX$"
```

これは次を生成します。

$$\frac{3x}{2}$$

この例が示すように、エスケープシーケンスの開始と終了には二重バックスラッシュ（「\\\$WMTEX\$」と「\\\$/WMTEX\$」）が必要です。

これらのエスケープシーケンスの内側には、二重バックスラッシュが使われていない TeX マークアップテキスト（「`\frac{3x}{2}`」）が含まれています。

Add Annotation ダイアログと Modify Annotation ダイアログでは、Special ポップアップメニューから「Igor TeX」を選択することで、開始および終了のエスケープシーケンスを挿入できます。

これらのダイアログを使う時は、二重バックスラッシュを入力しないでください。

Igor Pro が必要に応じて自動的に処理します。

上記の例では、開始エスケープシーケンスの後と終了エスケープシーケンスの前に空白が含まれています。

このような空白は省略可能です。

ローカルの文字列変数を使って、注釈のテキストを作成することができます。

例えば：

```
static StrConstant kTeXOpen = "\\$WMTEX$  
static StrConstant kTeXClose = "\\$/WMTEX$  
static StrConstant kTeXFontSize = "\\Z18"  
  
Function TeXTest()  
    DoWindow/F TextTestGraph  
    if (V_Flag == 0)  
        Display /N=TextTestGraph /W=(35,45,339,154)  
    endif  
  
    String TeXFormula = "\\frac{3x}{2}"  
    String annotationText = kTeXFontSize + kTeXOpen + TeXFormula + kTeXClose  
    TextBox/C/N=TeXTest/A=MC/F=0 annotationText  
  
End
```

その他の例については、Igor TeX Demo のエクスペリメントをご覧ください。

メニュー File→Example Experiments→Graphing Techniques→Igor TeX demo

デフォルトでは、数式にはインラインスタイルが適用されます。

TeX 数式の先頭に「`\displaystyle`」を追加することで、数式を別行で表示する時に一般的に使われる、より大きなスタイルに切り替えることができます。

Igor Pro は TeX の一部の機能をサポート

Igor Pro には完全な LaTeX インタープリタは含まれていません。

その代わりに、Knuth の TeX.web を模したコードを使っており、ユーザーが使う可能性が最も高い一般的な数式構文をサポートするサブセットを備えています。

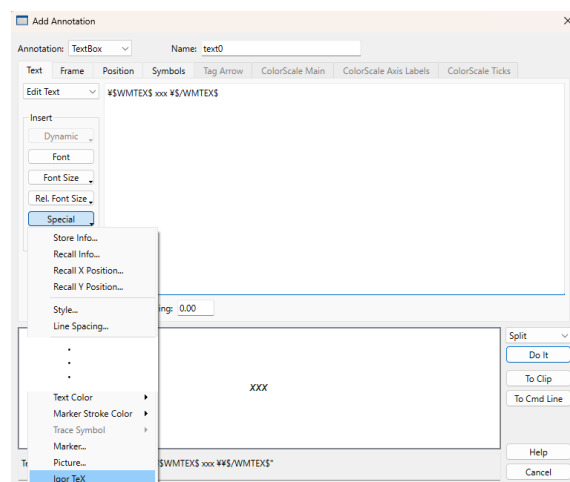
数式は、Igor Pro 通常のテキストと線描画コードを使って直接描画されます。

あらかじめ画像や .dvi ファイルを作成することはありません。

Igor Pro のサブセットは、LaTeX の `\frac` をサポートしていますが、標準 TeX の `\over` やマクロはサポートしていません。

`\rm` を使うと、文字をイタリック体ではなく、通常体（直立体）に強制的に表示させることができます。

これは、エタノールなどの化学式で役立ちます：`\rm CH_3CH_2OH`



Igor TeX で使われるフォント

どのフォントが使われているかを判断するために、TeX 数式の各構成要素は、以下のいずれかに分類されます。

- `\alpha`、`\beta`、`\gamma` といった TeX コードで指定されるギリシャ文字
- `\neg`、`\prod`、`\sum`、`\int` といった TeX コードで指定される数学記号
- その他の文字（英字、数字、関数名、上記で定義されたギリシャ文字や数式記号以外のあらゆる文字）

デフォルトでは、ギリシャ文字や数式記号に分類されるコンポーネントに対して、このフォントを使います。

- 代替フォントとして Cambria Math を使う記号

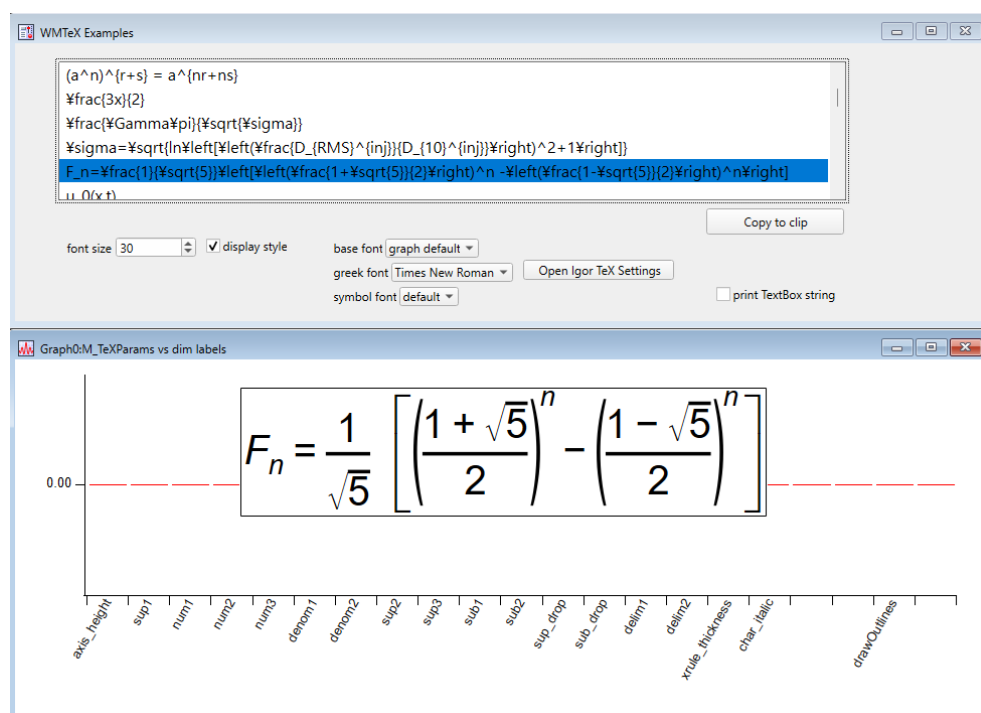
それ以外のテキストはすべて、TeX 数式の直前にある注釈で適用されているフォントを使って表示されます。

デフォルトのギリシャ文字と数式記号フォントは、外観や、角括弧の拡張記号（背の高い角括弧を作成する時に使われる）などの特殊文字への対応を基準に選定されました。

特定の注釈内で、ギリシャ文字についてはテキスト情報変数 8 に、数式記号についてはテキスト情報変数 9 にフォント名を指定することで、これらのデフォルト設定を上書きすることができます。

「テキスト情報変数」のセクションを参照してください。

Igor TeX Demo エクスペリメントを使って、さまざまなフォントを試すことができます。



α （アルファ）などのギリシャ文字を直接使うことも可能ですが、ギリシャ文字の表示用に選択された特定のフォントを使うためには、TeX の `\alpha` という構文を使うべきです。

数式で使われるフォントは、文字の視覚的な外観だけでなく、Igor PDF や EPSF といったグラフィック形式へエクスポートする時のフォント埋め込み状況にも影響を与えます。

ヘルプ Font Embedding (Windows) を参照してください。

標準フォントの埋め込みを無効にして Igor PDF や EPSF をエクスポートする場合は、出力結果を注意深く確認し、問題が見つかった場合は完全なフォント埋め込みを使ってください。