

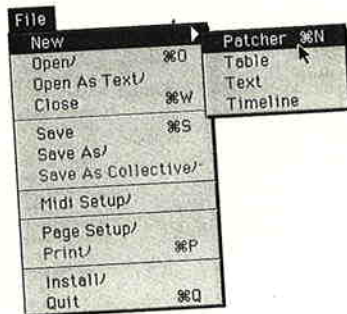
3-2 プログラミングの基本

Maxではどのようにプログラミングを行うのか、その具体的な手順を見ていこう。ここでの説明に従って操作をすれば、プログラミングの手順が理解できるだろう。文章としては長く煩雑に感じられるかもしれないが、実際の操作は難しくない。1度理解すれば、あとはスムーズに作業ができるだろう。

1 バッチ・ウィンドウの作成

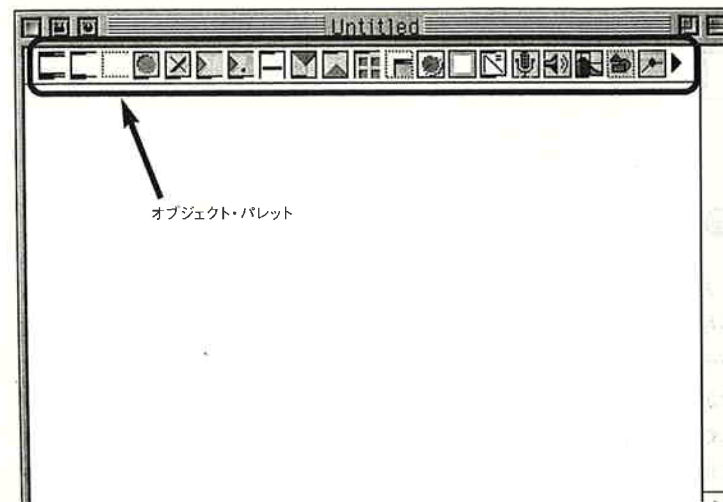
Maxでのプログラミングは、基本的にパッチ・ウィンドウで行う。FileメニューのNewからPatcherを選ぶ(command+N)と、新しいパッチ・ウィンドウが現れる。

■3-2-1 Fileメニューからパッチ・ウィンドウを作成



パッチ・ウィンドウには2つの動作モードがあり、最初はオブジェクトの作成やパッチ・コードの接続ができる編集モードになっている。編集モードでは、タイトル・バーの下にオブジェクト・パレットが表示される。このオブジェクト・パレットを使って、オブジェクトを作成する。

■3-2-2 作成されたパッチ・ウィンドウ



パッチ・ウィンドウのタイトル・バーは、通常のクローズ・ボックスなど以外に、2つの・ボックスを持っている。左側のボックスをクリックするたびに、パッチのロックとアンロック(ロック解除)とが切り替わる。アンロック状態では錠前が開いたマークが表示され、オブジェクト・パレットが表示される。アンロック状態が、いわゆる編集モードであり、パッチの編集ができる状態だ。ロックにすると、錠前が閉じたマークに変わり、オブジェクト・パレットが消える。ロック状態が実行モードであり、パッチの編集はできず、パッチの実行だけが行える。

2つ目のボックスは、MIDIの入出力の有効と無効を切り替える。円形のマークの表示はMIDI端子を表しており、この状態ではMIDIの入出力が可能だ。これをクリックして、×印になるとMIDIの入出力は無効になる。

なお、パッチ・ウィンドウの空白部分でcommandキーを押しながらクリックすれば、パッチのロックとアンロックを切り替えることができる。また、ViewメニューのEditを選ぶことでも(command+E)、切り替えることができる。

04/12/6

ロック — 実行モード (実行時)

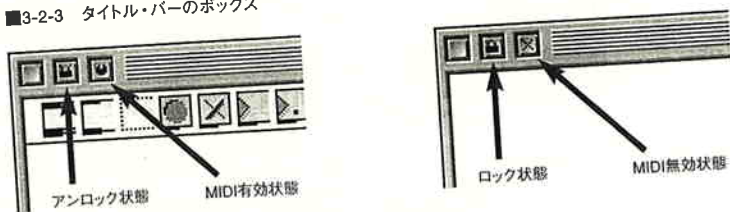
アンロック — 編集モード

Windowsでは

Ctrlキー

04/12/9

■3-2-3 タイトル・バーのボックス

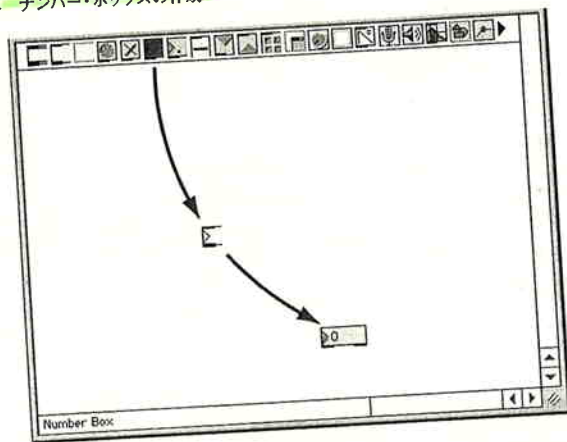


ユーザー・インターフェース・オブジェクトの作成

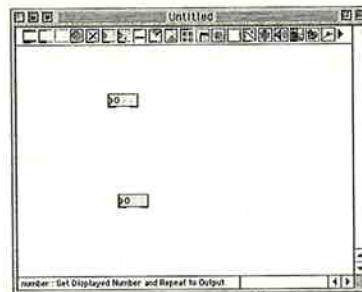
実際に簡単なパッチを作ってみよう。まず、パッチ・ウィンドウをアンロック状態、すなわちパッチが編集できる状態にして、ナンバー・ボックス (number 《Number Box》) というオブジェクトを作成する。オブジェクト・パレットには、いくつものオブジェクトのアイコンが並んでいるが、左から6番目がナンバー・ボックスのアイコンなので、これをクリックして選択する。この状態でウィンドウ内にマウスを移動させると、マウス・ポインターの形状がアイコンと同じ形状に変わるだろう。そして、ウィンドウ内でマウス・ボタンをクリックすれば、ナンバー・ボックスが作成され表示される。

次に、同じようにして、ナンバー・ボックスをもう1つ作成しよう。

■3-2-4 ナンバー・ボックスの作成

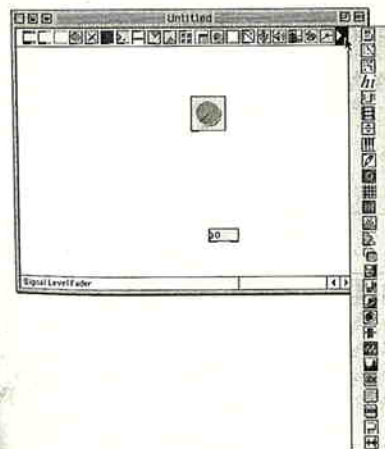


■3-2-5 2つのナンバー・ボックスを作成



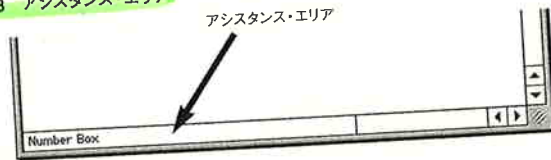
ちなみに、ウィンドウの横幅が十分でないときは、オブジェクト・パレットにすべてのオブジェクトが表示されない。そのような場合には、オブジェクト・パレットの右端の三角形マークをクリックすると、残りのオブジェクトがメニューとして縦に表示される。オブジェクト・パレットに表示されていないオブジェクトは、このメニューから選択して作成することができる。

■3-2-6 オブジェクト・パレットのメニュー表示



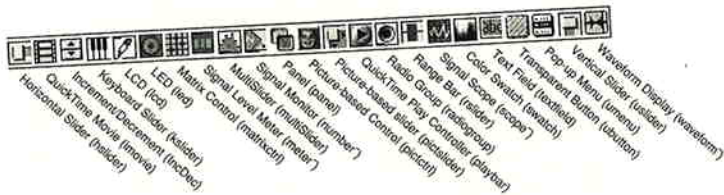
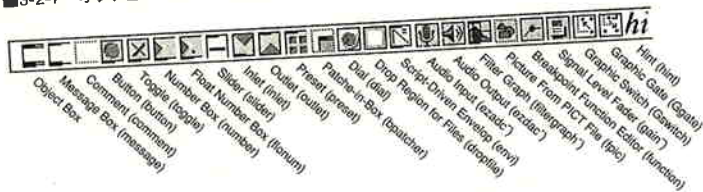
また、パッチ・ウィンドウの左下はアシスタンス・エリアと呼ばれ、オブジェクト・パレット上にマウス・ポインターがある場合は、そのオブジェクトの名称を表示する。オブジェクト・パレットに表示されるアイコンだけでは、そのオブジェクトが何か判別しにくいときは、アシスタンス・エリアの表示を参考にするとよいだろう。

■3-2-8 アシスタンス・エリア



なお、オブジェクト・パレットから作成できるオブジェクトは次の通りだ。標準では47個ものオブジェクトがある。ここでは、オブジェクト作成時にアシスタンス・エリアに表示されるオブジェクトの名称を付記している。カッコ内は実際のオブジェクトの名前だ。オブジェクトを作成すれば、それ以降はアシスタンス・エリアに実際のオブジェクト名が表示される。

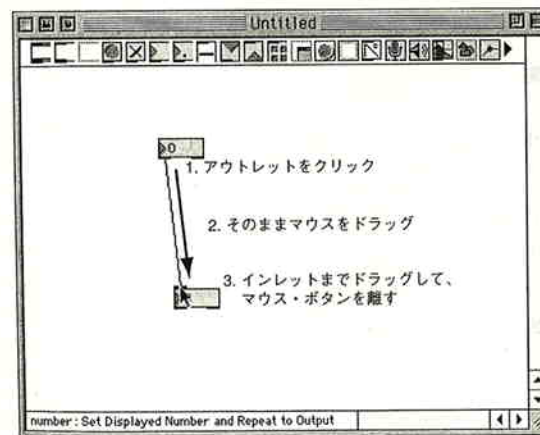
■3-2-7 オブジェクト・パレットのオブジェクト



● パッチ・コードの接続

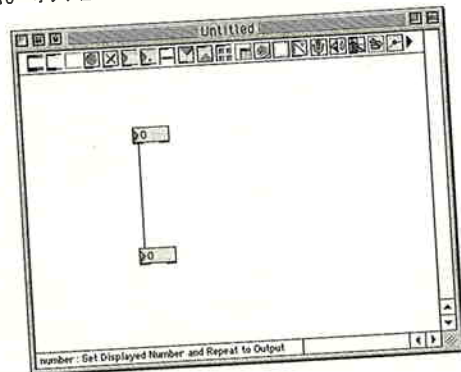
次に、2つのナンバー・ボックスを、パッチ・コードで接続する。まず、上側にあるナンバー・ボックスの左下付近にマウス・ポインターを近づけると、アウトレットが強調表示される。この状態でマウス・ボタンをクリックする。そのまま下へマウスをドラッグすると、マウス・ポインターの動きに連れてパッチ・コードが伸びていく。そして、下側のナンバー・ボックスの左上付近までマウス・ポインターを動かすと、インレットが強調表示されるので、マウス・ボタンを離す。これで、2つのナンバー・ボックスのアウトレットとインレットとが、パッチ・コードで接続される。ちなみに、パッチ・コードを削除するには、パッチ・コードをクリックして選択し、deleteキーを押せばよい。

■3-2-9 パッチ・コードの接続



なお、作成したオブジェクトをドラッグすれば、その位置を変えることができる。接続されているパッチ・コードもオブジェクトの位置に合わせて変化する。適宜、見やすい位置にオブジェクトを移動すればよいだろう。

■3-2-10 オブジェクトの位置を整えたパッチ・コード



● パッチの実行

パッチを実行するには、錠前マークのボックスをクリックして、錠前が閉じた表示にする。これでオブジェクトの操作だけができるロック状態になる。

この状態で、上のナンバー・ボックスをドラッグすれば、下のナンバー・ボックスの数値も変化する。これは、ユーザーがナンバー・ボックスを操作することで、その値がメッセージとして出力され、パッチ・コードを伝わって下のナンバー・ボックスがメッセージを受け取り、その数値が表示されたことを意味する。また、下のナンバー・ボックスをドラッグして数値を変えることもできる。しかし、この場合はアウトレットの出力がどこにもつながっていないので、他のオブジェクトは動作しない。

■3-2-11 パッチの実行



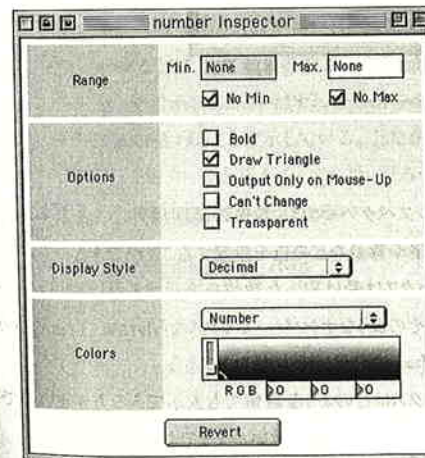
ちなみに、パッチがアンロック状態であっても、実際にはパッチは動作可能な状態になっている。commandキーを押しながらナンバー・ボックスをドラッグすれば、パッチが動作することを確かめられるはずである。また、MIDIなど外部機器からのデータを受け取れば、それに対応したオブジェクトが動作する。したがって、commandキーを使い分けることで、パッチを編集しながらパッチを動作させることができる。Maxに慣れてくれば、このような方法が便利だろう。

● オブジェクトの設定

ナンバー・ボックスなどのユーザー・インターフェース・オブジェクトは、その動作や表示を変更することが可能だ。例えば、ナンバー・ボックスを動かすと、そのままでは無制限に値が変化するが、これを一定の範囲に限定することができる。

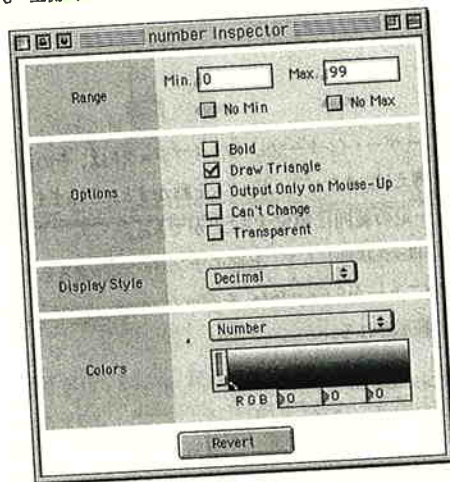
このためには、パッチをアンロック状態にして、上部のナンバー・ボックスをクリックして選択する。そして、ObjectメニューからGet Info...を選べば(command+I)、ナンバー・ボックスのインスペクターが表示される。

■3-2-12 ナンバー・ボックスのインスペクター



このインスペクターで、No MinとNo Maxのチェック・ボックスのチェックを外し、Min.を0、Max.を99と設定する。設定が終われば、クローズ・ボックスをクリックしてインスペクターを閉じる。こうしてから、パッチをロックしてナンバー・ボックスをドラッグすると、0から99までの範囲で値が変化し、それ以外の数値にはならない。

■3-2-13 上側のナンバー・ボックスの範囲を設定

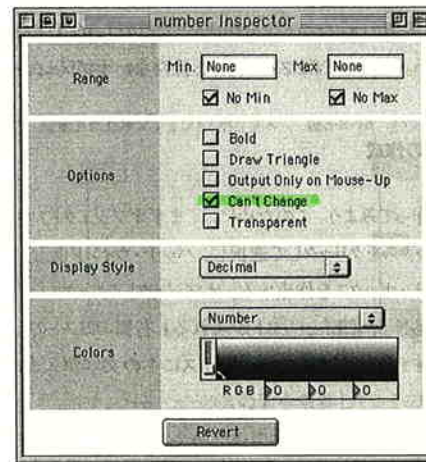


このナンバー・ボックスのインスペクターでは、数値の範囲以外に、表示や操作のオプション、表示形式、そして数字や背景などの色を設定することができる。

例えば下側のナンバー・ボックスは受け取った数値を表示するだけで、ユーザーに操作を求めるわけではない。そのようなナンバー・ボックスならDraw Triangleのチェックを外し、Can't Changeにチェックを付ける方がよいだろう。

また、下側のナンバー・ボックスはどのような数値でも表示できる方がよいので、No MinとNo Maxのチェック・ボックスはチェックが付いたままでよい。

■3-2-14 下側のナンバー・ボックスの設定を変更



これで下側のナンバー・ボックスの左側には三角形マークが描かれなくなり、ナンバー・ボックスのドラッグなどによって数値を変更することができなくなる。

三角形マークは、そのオブジェクトがナンバー・ボックスであることを明示し、上下にドラッグすれば数値が変わることを示している。したがって、表示専用のナンバー・ボックスであれば、三角形マークを描かなくてもよいだろう。このようにしてオブジェクトの設定を変更したパッチは、3-2-15のようになる。

■3-2-15 オブジェクトの設定を変更したパッチ



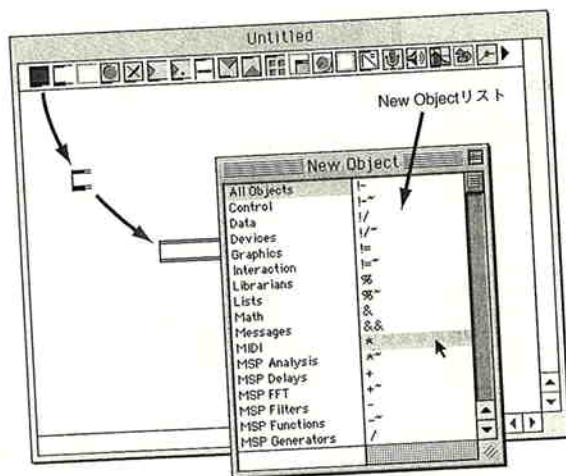
なお、インスペクターの下部にあるRevertボタンをクリックすれば、インスペクターを開いた時点の設定に戻すことができる。またインスペクターで設定できる内容は、オブジェクトによって異なる。他の種類のオブジェクトを作成して、どのようなインスペクターが表示されるか確かめてみよう。ただし、インスペクターを持たないオブジェクトもある。

04/11/22

3-2-16 ノーマル・オブジェクトの作成

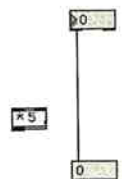
次に、乗算を行うパッチを作ってみよう。このためには、まずオブジェクト・パレットの一番左にあるオブジェクト・ボックスをクリックして選択し、パッチ・ウィンドウ上の適当な場所をクリックして、オブジェクト・ボックスを作成する。オブジェクト・ボックスを作成すると同時に、2列のリストから成るNew Objectリストが現れる。右側のリストの中から乗算を表す*をダブル・クリックすれば、オブジェクト・ボックスに*の文字が入力される。

3-2-16 オブジェクト・ボックスの作成



オブジェクト・ボックスでは文字入力ポインターが点滅しているので、スペース・バーを押して空白文字を入れ、さらに数字の5を入力する。そしてenterキーを押すか、パッチ・ウィンドウ上の空白部分をクリックする。これでオブジェクト・ボックスの内容が確定され、インレットやアウトレットが現れる(3-2-17)。

3-2-17 確定されたオブジェクト・ボックス



3-2-18 パッチ・コードの接続を変更したパッチ



ここで、上のナンバー・ボックスの値を乗算して、その結果を下のナンバー・ボックスに表示するようにしたい。そのためには、2つのナンバー・ボックスの間に*5オブジェクトを挟み込むようにパッチ・コードを接続すればよい。そこで、まず、2つのナンバー・ボックスをつなぐパッチ・コードをクリックして選択する。そして、deleteキーを押せばパッチ・コードが削除される。次に、上のナンバー・ボックスから*5オブジェクトにパッチ・コードをつなぎ、さらに*5オブジェクトから下のナンバー・ボックスにパッチ・コードをつなぐ。接続後は、オブジェクトの位置も整えておこう(3-2-18)。

これで、乗算を行うパッチ3-2-19ができた。ウィンドウをロックし、上のナンバー・ボックスをドラッグして数値を確認しよう。アンロック状態のまま、commandキーを押しながらナンバー・ボックスをドラッグしてもよい。上のナンバー・ボックスの値が0から99までの範囲に設定しているので、出力された数値は*5オブジェクトによって5倍され、下のナンバー・ボックスには0から495までの数値が表示されるはずである。

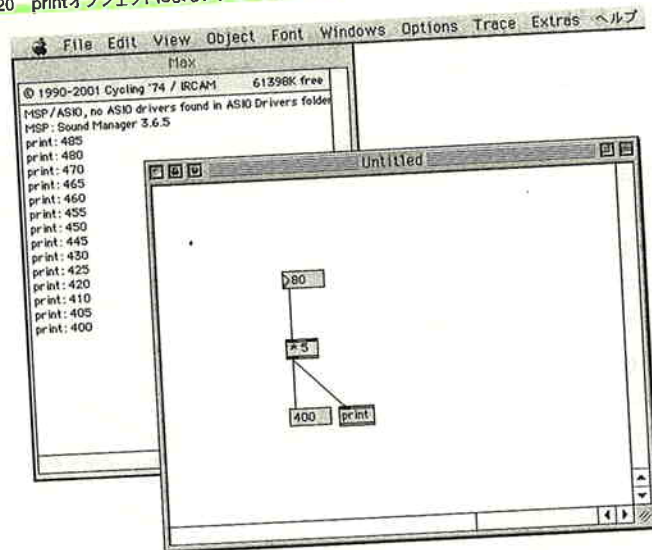
3-2-19 乗算パッチの実行



さらに、オブジェクト・ボックスを使ってprintオブジェクトを作っておこう。printオブジェクトは、受け取ったメッセージをそのままMaxウィンドウに表示するオブジェクトで、接続は * 5オブジェクトのアウトレットからprintオブジェクトのインレットへパッチ・コードをつなぐ。これで、上のナンバー・ボックスの数値を動かせば、次々と乗算の結果がMaxウィンドウに表示されるだろう。

ナンバー・ボックスの表示では、最新の数値しか表示されないが、printオブジェクトを用いれば、それまでの数値もMaxウィンドウで確認することができる。

■3-2-20 printオブジェクトによるメッセージの表示

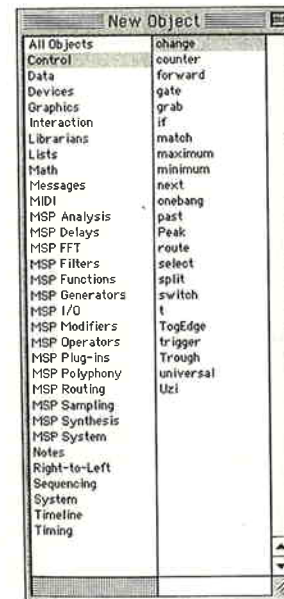


● New Objectリストの利用

オブジェクト・ボックスによって作られるノーマル・オブジェクトは、極めて多くの種類がある。そこで、オブジェクトを効率よく選択する手段として、New Objectリストが用意さ

れている。左側のリストには、オブジェクトの種類を表す項目が並んでおり、All Objectsが選択されているときは、右側のリストにはすべてのノーマル・オブジェクトが表示される。同じようにControlの場合は、メッセージの流れを制御するオブジェクトが表示され、Dataならデータ処理のオブジェクトが表示される、といった具合だ。

■3-2-21 オブジェクトの種類ごとのリスト表示



1文字だけ?

New Objectリストが表示されているときに、英数字をタイプすると、該当する項目が選択される。これは下側のエリアがドット模様になっているリストが対象であり、下側のエリアをクリックすることで、対象とするリストを切り替えることができる。

なお、作成したいオブジェクトの名前を覚えているなら、オブジェクト・ボックスに直接オブジェクトの名前を入力すればよい。オブジェクト・ボックスをクリックすれば、New Objectリストが閉じるので、その後オブジェクト名をタイプすることになる。

ちなみに、OptionsメニューのNew Object Listのチェックを外すと、New Objectリストは表示されないようになる。また、New Objectリストが表示されていない場合でも、空のオブジェクト・ボックスをoptionキーを押しながらクリックすると、New Objectリストが表示される。

■3-2-22 Optionsメニュー



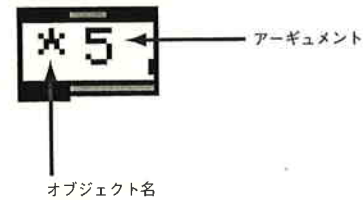
③ アーギュメントの指定

ここで、乗算を行う * 5 オブジェクトを、詳しく見ておこう。* 5 オブジェクトは2つの部分から成っており、その1つがオブジェクトの名前を示す * だ。オブジェクトの名前によって、そのオブジェクトが何を行うかが決まる。* なら乗算であり、+ なら加算ということになる。

オブジェクト名以降の部分はアーギュメントと呼ばれ、オブジェクトの動作方法や処理内容の初期値を指定するために用いる。アーギュメントは、他の言語では引数やパラメーターとも呼ばれる。* 5 オブジェクトなら5がアーギュメントに当たり、掛ける数値、すなわち乗数として5を指定したことになる。したがって、アーギュメントを10として、* 10 というオブジェクトを作れば、10倍の乗算を行うオブジェクトができる。

アーギュメントは省略することも可能で、その場合はデフォルト値が用いられる。* オブジェクトのアーギュメントのデフォルト値は0だ。デフォルト値はオブジェクトによって異なるが、数値のアーギュメントのデフォルト値は0であることが多い。

■3-2-23 オブジェクトの名前とアーギュメント



注意点としてはオブジェクトの名前とアーギュメントとの間には、スペース(半角の空白文字)を入れなければならないということが挙げられる。アーギュメントは数値の場合もあれば、シンボル(文字列)の場合もある。また、2つ以上のアーギュメントを指定できるオブジェクトもあり、アーギュメントとアーギュメントもスペースで区切って記述する。なお、アーギュメントを持たないオブジェクトもある。

■3-2-24 オブジェクトとアーギュメントの例

```
counter 0 0 15
```

```
graphic canvas
```

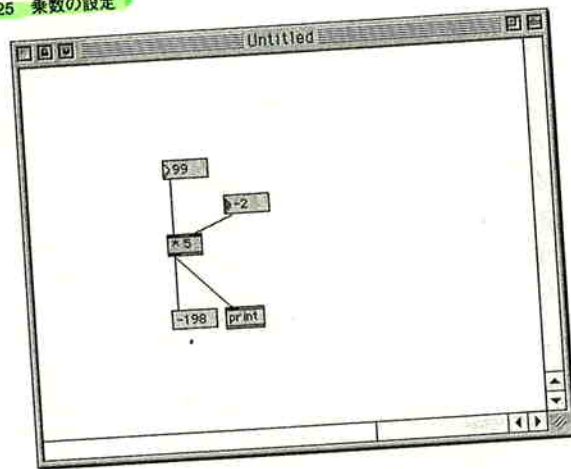
```
if ($i1 < 0) then 0 else $i1
```

```
key
```

ほとんどのオブジェクトは、アーギュメントで指定した内容を変更することができる。* オブジェクトなら、2番目のインレットに数値を受け取れば、それ以降、その数値を乗数として処理が行われる。ナンバー・ボックスを作り、* オブジェクトの2番目のインレットにつないでみよう。そして、ナンバー・ボックスをドラッグして数値を変えた上で、上側のナンバー・ボックスを動かせば、その変化が分かるだろう。

例えば、乗数のナンバー・ボックスを-2とすれば、* オブジェクトの乗数は5ではなく-2になる。したがって、上側のナンバー・ボックスを動かすにつれて、* オブジェクトの計算結果は0から-198の範囲で変化するはずだ。もっとも、オブジェクトの見え方としては* 5のまま、表示が変化するわけではない。

■ 3-2-25 乗数の設定

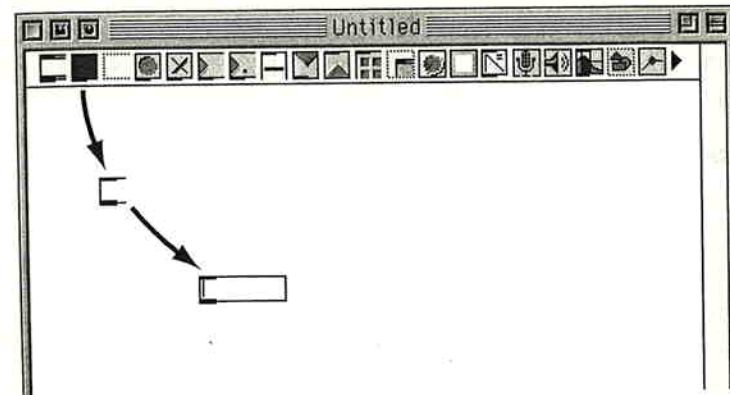


ところで、ノーマル・オブジェクトはアークギュメントを指定できるが、ユーザー・インターフェイス・オブジェクトはアークギュメントを記述することはできない。インスペクターを使って設定する内容がアークギュメントに相当すると思えばよいだろう。

2) メッセージ・ボックスの作成

数値を含めて、特定のメッセージを出力するには、メッセージ・ボックス (message 《Message Box》) というオブジェクトを利用する。メッセージ・ボックスは、オブジェクト・パレットの2番目にあるので、これをクリックして選択し、ウィンドウ内をクリックすれば、メッセージ・ボックスが作成される。

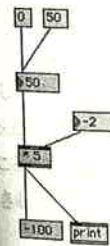
■ 3-2-26 メッセージ・ボックスの作成



メッセージ・ボックスには、数値や文字列などの任意のメッセージを入力すればよい。そして、メッセージ・ボックスのアウトレットから、他のオブジェクトのインレットにパッチ・コードをつなぐ。これで、パッチ・ウィンドウをロックして、メッセージ・ボックスをクリックすれば、その内容が出力される。

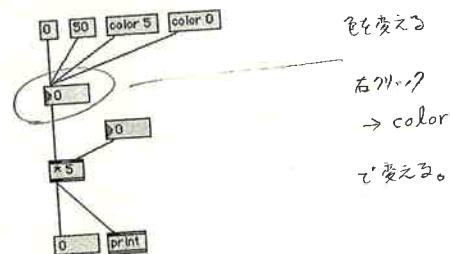
例えば、0や50といった数値のメッセージ・ボックスを作り、ナンバー・ボックスにつないでみよう。このような数値のメッセージを送れば、ナンバー・ボックスはその数値に設定され、同時にその数値を出力する。したがって、乗算も行われて下のナンバー・ボックスに結果が表示されることになる。

■ 3-2-27 ナンバー・ボックスへの数値メッセージ



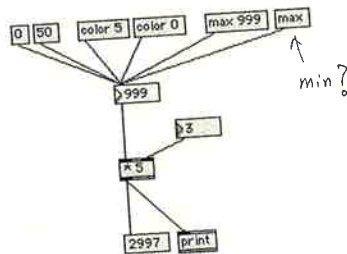
ではcolor 10とcolor 0というメッセージ・ボックスを作り、ナンバー・ボックスにつないでみよう。color 10というメッセージを送れば、ナンバー・ボックスが黄色に変化する。color 0なら元の薄い茶色に戻るはずだ。colorは、ナンバー・ボックスなどの表示色を設定するメッセージで、続く0から15までの数値で色を指定するようになっている。この場合の数値を、colorメッセージのアーギュメントと呼ぶ。

3-2-28 ナンバー・ボックスへのcolorメッセージ



同じように、ナンバー・ボックスには数値の範囲を設定するminメッセージやmaxメッセージがあり、いずれも続く数値で最小値や最大値を指定する。数値を省略した場合は、最小値や最大値の制限がなくなり、インスペクターでNo MinやNo Maxをチェックした状態になる。

3-2-29 ナンバー・ボックスへのmaxメッセージ



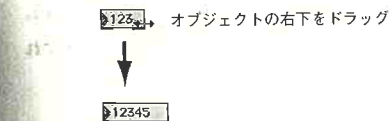
このように、オブジェクトはさまざまなメッセージを受け取り、メッセージによって動作や表示を変えることができる。どのようなメッセージを受け取るかは、オブジェクトによって異なる。詳しくは、リファレンス・マニュアルを参照してほしい。

なお、オブジェクト・ボックスとメッセージ・ボックスは見かけが似ているため、混同しやすい。オブジェクト・ボックスは上下の辺が二重線で描かれるところが、メッセージ・ボックスと異なる。オブジェクト・ボックスのつもりで、メッセージ・ボックスを作り、パッチが動作しないという単純なミスに陥らないように気を付けよう。

3 オブジェクトのリサイズ

乗算の結果が桁数の多い数値になれば、ナンバー・ボックスに表示しきれなくなる。通常のナンバー・ボックスは4桁まで表示できるが、それ以上の桁数になると数字の最後に...と表示され、それ以降に数字があることを示すようになる。より多くの桁数を表示するには、ナンバー・ボックスの大きさを変えればよい。このためには、ナンバー・ボックスの右下付近にマウス・ポインターを近づければ、左右向きの矢印にマウス・ポインターの形状が変わるので、この状態でマウス・ボタンをクリックし、右または左にドラッグすればよい。これでナンバー・ボックスを大きくすることも、小さくすることもできる。

3-2-30 ナンバー・ボックスのリサイズ



同じ方法で、他のオブジェクトも大きさを変えることができる。オブジェクト・ボックスやメッセージ・ボックスもリサイズ可能だ。ナンバー・ボックスなどは横幅のみ大きさを変えることができるが、button《Button》オブジェクトなどは、全体の大きさを変えることができる。buttonオブジェクトはオブジェクト・パレットの4番目にある。

■3-2-31 buttonオブジェクトのリサイズ

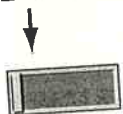
☐ オブジェクトの右下をドラッグ



さらに、hsliderオブジェクトなどは、横幅と縦幅を変更することができる。hsliderオブジェクトはオブジェクト・パレットの25番目にある。

■3-2-32 hsliderオブジェクトのリサイズ

☐ オブジェクトの右下をドラッグ



なお、dialオブジェクトのように大きさが固定で変更できないオブジェクトもある。どのようにリサイズできるかはオブジェクトによって異なるが、オブジェクトの右下にマウスを近づけたときのポインターの形状で、リサイズの可否やリサイズの方向が分かるだろう。また、複数のオブジェクトを選択している際に、いずれかのオブジェクトをリサイズすれば、すべてのオブジェクトを同じ比率でリサイズすることができる。

② オブジェクトのフォント指定

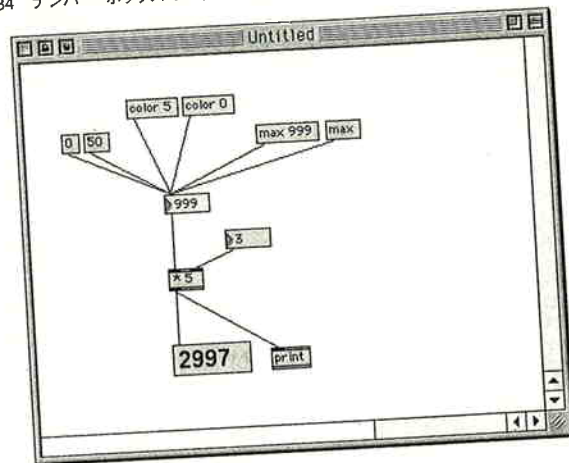
オブジェクト・ボックスやメッセージ・ボックス、ナンバー・ボックスなどは、その内部に文字や数字を表示するが、これらの文字のフォントを指定することができる。これは、単にオブジェクトをクリックして選択し、Fontメニューからフォントのサイズや種類を選ぶだけでよい。重要な操作や結果を表すオブジェクトは、大きなサイズや目立つフォントを使って強調すればよいだろう。

■3-2-33 Fontメニュー



ここでは、計算結果を表示するナンバー・ボックスについて、太目のフォントを指定し、フォント・サイズも大きくしている。また、これに伴い、ナンバー・ボックス自体の大きさも変えている。

■3-2-34 ナンバー・ボックスのフォントを設定



なお、デフォルトではGenevaの9ポイントのフォントが選ばれている。オブジェクトを選択していない状態で、Fontメニューでフォントのサイズと種類を選べば、それ以降に作成するオブジェクトは、その指定に従ったフォントが用いられる。特定のサイズやフォントでパッチを作りたいときは、最初にフォントを選択しておくとういだろう。

また、日本語フォントを指定してもよいが、後述するcomment《Comment》オブジェクトを除いて、一部の2バイト文字(全角文字)は文字化けする。原則的に、日本語フォントや2バイト文字は使用しない方がよい。さらに、システムに標準ではないフォントも避けた方がよいだろう。他のコンピューターでパッチを開くと、インストールされていないフォントは意図通りに表示されないからだ。

commentオブジェクトの活用

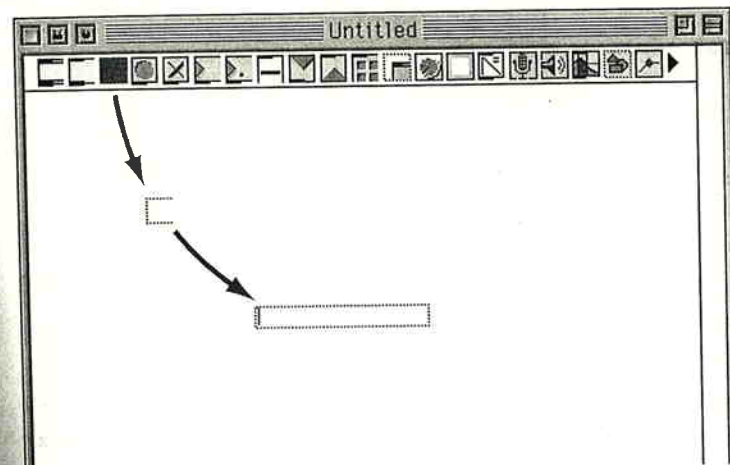
簡単なパッチであれば、それをどのように操作するかを直ちに理解できるかもしれない。しかし、少し込み入ったパッチとなれば、何が何だか分からなくなってしまう。そこで、活用したいのがこのcommentオブジェクトだ。

ユーザー・インターフェース・オブジェクトの役割を示すラベルとしてコメントを添えれば、ユーザーがパッチを利用しやすくなる。コメントによって操作手順を表示することも考えられる。また、所要所に処理の説明をコメントとして入れておけば、のちほどパッチを改造するときに役立つだろう。

実際、Maxでは思いのままパッチを拡張していきけるのだが、それゆえに内容を読み取ることが難しくなりがちだ。したがって、適切なコメントを付けることが重要になる。

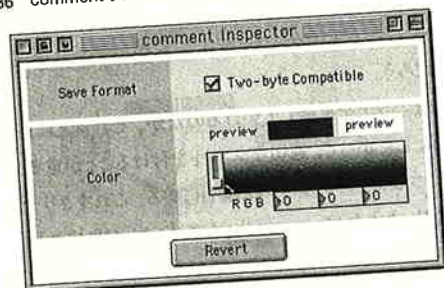
commentオブジェクトは、オブジェクト・パレットの3番目にある。これをクリックして選択し、パッチ・ウィンドウ内をクリックすれば、commentオブジェクトが作成され、エリア内には任意の文字を入力することができる。また、Fontメニューによって、コメントに用いるフォントとフォント・サイズを指定することができる。

■3-2-35 commentオブジェクトの作成



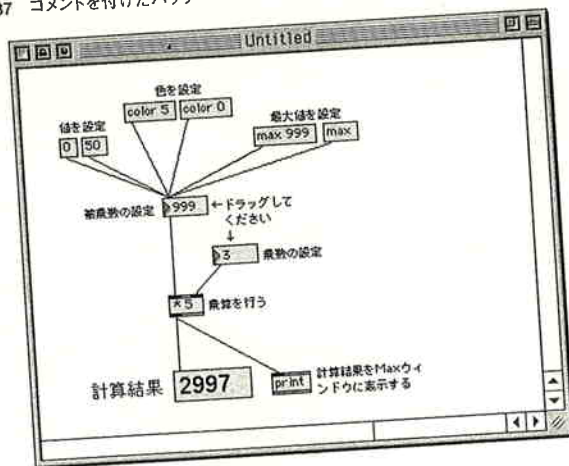
コメントには日本語フォントを選択して、漢字など2バイト文字を使用することが可能だが、そのためにはcommentオブジェクトのインスペクターを開いて、Two-byte Compatibleをチェックすることが必要になる。このチェックを行わないと、一部の2バイト文字は文字化けしてしまう。

■3-2-36 commentオブジェクトのインスペクター



これまで作成したパッチにコメントを付けると、以下ようになる。この例はコメントの付け過ぎという気もするが、適切なコメントを付けるように心がけるのがよいだろう。

■3-2-37 コメントを付けたパッチ

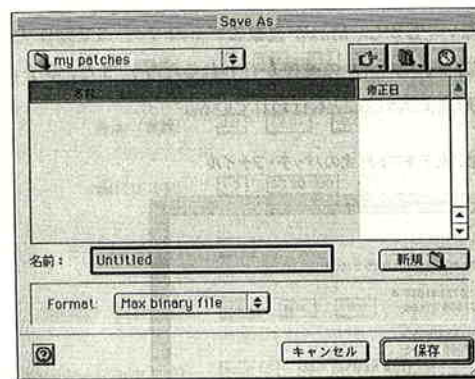


なお、コメントは単に注釈を表示するだけで、パッチの動作には関係がない。また、コメントにはインレットもアウトレットもないので、他のオブジェクトとメッセージをやり取りすることもない。

● パッチの保存

このようにして作成したパッチは、ファイルとしてハード・ディスクなどに保存することができる。パッチを保存するには、FileメニューのSaveを選べばよい。パッチを初めて保存する際や、FileメニューのSave As...を選んで別の名前でも保存するときは、ファイル保存ダイアログが開く。ファイル保存ダイアログでは、ファイルを保存する場所を指定し、ファイルの名前を入力して保存ボタンをクリックする。

■3-2-38 ファイル保存ダイアログ



Maxのファイル保存ダイアログでは、Formatポップアップ・メニューによってファイル・フォーマットを選択することができる。ファイル・フォーマットにはMax binary fileとMax text fileとがある。

■3-2-39 2つのファイル・フォーマットのファイル・アイコン

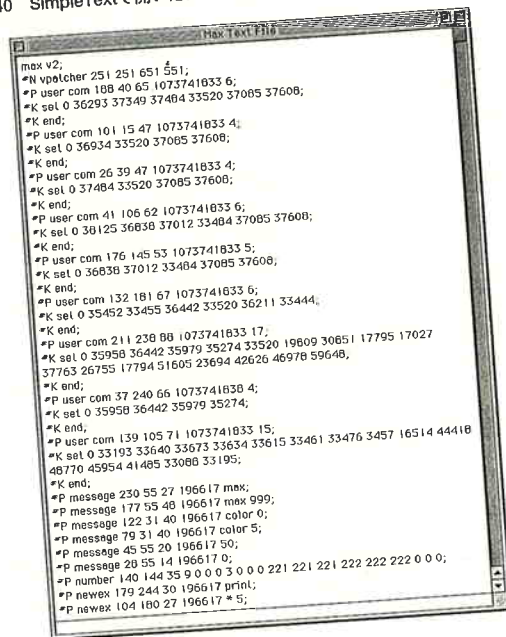


Max binary fileはMaxの独自フォーマットによるバイナリー形式で、Max text fileは一般的なテキスト形式となっている。Max binary fileはファイル容量が少なくて済むため、通常はMax binary fileとしてファイルを保存すればよいだろう。どちらの形式で保存しても、Maxで開けば同じパッチとして表示される。

テキスト形式で保存したファイルをSimpleTextなどのテキスト・エディターで開くと、記号や数字が並んでいることが分かる。また、MaxのFileメニューからOpen As Text File...を選べば、ファイル・フォーマットに関わらず、パッチをテキストとして開くことができる。このテキストはオブジェクトやパッチ・コードの接続などを表している。

現時点で、他のアプリケーションでMax text fileとして保存したファイルを利用する機会は少ないかもしれない。しかし、Maxが複数のプラットフォームで動作するようになれば、テキスト形式でパッチ・ファイルの交換が行われるだろう。また、電子メールの本文にパッチをテキスト形式で入れることも行われている。

■3-2-40 SimpleTextで開いたテキスト形式のパッチ・ファイル



全部は何種類か
それらはどんな
メッセージか?
05/1/20

3-3 メッセージの種類

Maxではオブジェクトにメッセージを送ることで処理が行われる。したがって、どのような種類のメッセージがあり、それらをどのように使うかが重要になる。Maxで使用する基本的なメッセージのタイプは5種類だが、これらを組み合わせたメッセージも利用できる。ここでは、メッセージの種類と使用方法について説明する。

■3-3-1 メッセージのタイプと例

int (整数)	<input type="text" value="0"/>	<input type="text" value="538"/>	<input type="text" value="-17"/>
float (実数)	<input type="text" value="0.0"/>	<input type="text" value="12.01"/>	<input type="text" value="-0.05"/>
list (リスト)	<input type="text" value="1 5"/>	<input type="text" value="60 80 500"/>	<input type="text" value="0 -0.1"/>
bang (バン)	<input type="text" value="bang"/>		
symbol (シンボル)	<input type="text" value="start"/>	<input type="text" value="open"/>	<input type="text" value="clear"/>
message (メッセージ)	<input type="text" value="color 15"/>	<input type="text" value="open portrait.pict"/>	

● bangメッセージ

bangメッセージは、オブジェクトに何らかの動作のきっかけを与える特別なメッセージだ。オブジェクト・パレットの4番目にあるbuttonオブジェクトは、それをクリックするとbangメッセージを出力する。また、メッセージ・ボックスにbangと入力してクリックしても、bangメッセージが出力される。

bangメッセージを受け取ったオブジェクトは、オブジェクトによって異なるが、何らかの動作を行うことになる。例えば、randomオブジェクトはbangメッセージを受け取ると、0からアーギュメントで指定した数値より1小さい数値の範囲で、ランダムに選ばれた整数を出力する。