

ただし、メッセージが入れ替わるとはいえ、あくまでright-to-left orderのルールに従っていることを理解しておこう。

3-7-14のパッチは、rslider (Range Bar) の設定範囲を計算している。rsliderはマウス・ドラッグで数値の範囲を設定するが、この際、第2インレットから最大値を出力し、次に第1インレットから最小値を出力する。したがって、パッチの右部分のように、単純にアウトレットからの出力をクロス(タスキがけ)して、オブジェクトにつなぐと、正しい計算ができない場合がある。

そのような場合でも、swapオブジェクトによってメッセージを入れ替えれば、常に正しい計算ができるのである。

### 3-8 処理の自動化

パッチが何らかの処理を行うのは、ユーザーがボタンをクリックするといった操作によって、オブジェクトからメッセージが出力されることが契機となる。逆に言えば、メッセージが発生しない限りパッチは何も動作を行わない。一方、目的によって自動的に処理を始めたい場合もある。また、煩雑な操作を避けて、一括して処理を行いたい場合もあるだろう。ここでは、そのような場合に役立つオブジェクトについて説明する。

#### ① loadbangオブジェクトによる開始処理

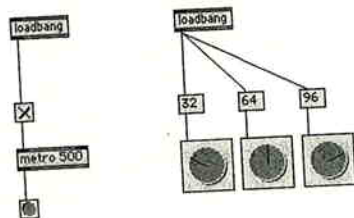
パッチが開くと同時に動作を開始したい場合や、初期設定を行いたい場合には、loadbangオブジェクトが利用できる。loadbangはパッチ・ファイルが開かれたときにbangメッセージを出力するので、このbangメッセージによって処理を行うようにパッチを作ればよい。また、loadbangオブジェクトをダブル・クリックすればbangメッセージが出力される。これで、パッチを開いた際の動作を確認することができる。

3-8-1の左側のパッチでは、loadbangオブジェクトからの出力をtoggleオブジェクトにつないでいるので、パッチが開かれるとtoggleオブジェクトにbangメッセージが送られる。toggleオブジェクトはbangメッセージを受け取ると自らの状態を変えるので、チェック・マークが付いた状態になり、1を出力する。そして、1を受け取ったmetroオブジェクトが動作を開始し、0.5秒ごとにbangを出力する。結果的に、パッチを開くと同時にbuttonオブジェクトが点滅しはじめることになる。

右側のパッチはloadbangからメッセージ・ボックスにつないでいる。そのため、パッチを開くとメッセージ・ボックスのメッセージがdialオブジェクトに送られ、その値が設定される。dialやhslider、rsliderなどのユーザー・インターフェース・オブジェクトの初期状態はパッチに保存されないため、この例のようにloadbangオブジェクトとメッセージ・ボックスを使用するによって、さまざまな初期設定を行う方法が役に立つだろう。

なお、MacOSの場合、システムフォルダの起動項目フォルダにあるファイルは、コンピュータの起動後に、自動的に開かれるようになっている。そこで、loadbangによって自動的に動作を開始するようなパッチを作成し、そのパッチ・ファイルを起動項目フォル

■ 3-8-1 loadbangオブジェクトの利用例



ダに入れておけば、コンピューターの電源を入れるだけで、パッチを自動的に動作させることも可能だ。さらに、省エネルギー設定のコントロール・パネルを使って、コンピューターを設定時刻に起動し、終了するように設定することも考えられる。このようにすれば、必要な時間帯だけ動作するように自動運転することもできるのである。

ただし、loadbangは便利なオブジェクトではあるが、間違った処理を組み込めばパッチを開くと同時に異常終了してしまうことも有り得る。しかも、パッチを開くと同時に異常終了するのだから、パッチを修正することができない。このような場合にはあわてずに、commandキーとshiftキーを押しながらパッチを開こう。こうすることでloadbangオブジェクトの動作を止めることができる。あるいは、FileメニューのOpen As Text...を選んで、テキストとしてパッチを開く方法もある。パッチをテキストとして開いたら、その中からloadbangの文字を探して、これを例えばbangbangに置き換えて保存しよう。これで、パッチを開いても自動的な動作は行われない。

### ● closebangオブジェクトによる終了処理

パッチが閉じられるときに何らかの処理を行うにはclosebangオブジェクトが利用できる。closebangはパッチが閉じられようとするときにbangメッセージを出力する。このbangメッセージを引き金にして、必要な処理を行えばよい。3-8-2の左側のパッチでは、closebangによってメッセージ・ボックスのメッセージを出力するので、パッチを閉じればsee you again!という挨拶をMaxウィンドウに表示する。

右側のパッチでは、closebangによってwrite mydata.textというメッセージをTextオブジェクトに送る。Textはテキストを扱うオブジェクトだが、その内容を自動保存する機能はない。そこで、パッチが閉じられる際に、アーギュメントでファイル名を指定してwriteメッセージを送れば、ファイルが自動的に作られて保存できるわけだ。

■ 3-8-2 closebangオブジェクトの利用例



### ● presetオブジェクトによる設定記憶

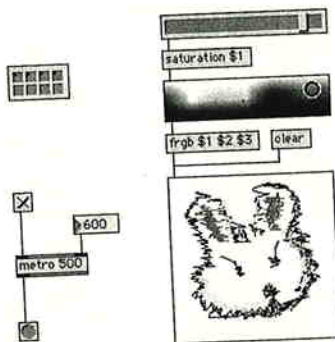
ナンバー・ボックスなどのユーザー・インターフェース・オブジェクトの値を記憶したり、呼び出したりしたい場合には、オブジェクト・パレットの11番目にあるpreset (Preset) オブジェクトが役立つ。presetオブジェクトを作成すると、6個のボタンを持ったオブジェクトが現れる。このボタンがユーザー・インターフェース・オブジェクトの値を記憶し呼び出すプリセット・ボタンになる。presetオブジェクトの大きさを変えることで、プリセット・ボタンの数を増やすことも減らすこともできる。

パッチをロックした状態で、いずれかのプリセット・ボタンをshiftキーを押しながらクリックすると、同じパッチ・ウィンドウにあるすべてのユーザー・インターフェース・オブジェクトの値を記憶する。記憶されたプリセット・ボタンは、左上に小さなドット・マークが付く。そして、プリセット・ボタンをクリックすれば、記憶されたユーザー・インターフェース・オブジェクトの値が再現される。このときにユーザー・インターフェース・オブジェクトは、その値をメッセージとして出力するので、関連するオブジェクトの動作も変更される。つまり、presetオブジェクトによって、パッチの動作状態を記憶し、再現することができるわけだ。3-8-3のパッチには、ナンバー・ボックス、toggle、swatch (Swatch)、sliderの各オブジェクトがあるが、これらの設定をpresetに記憶させて、呼び出すことができる。ユーザー・インターフェース・オブジェクトの設定を変えながら数個のプリ

セット・ボタンに記憶させたあと、プリセット・ボタンで呼び出すと分かりやすい。

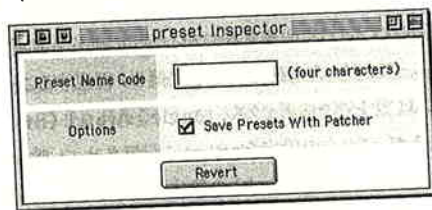
このパッチではmetroオブジェクトによってボタンが点滅する速さや、lcdオブジェクトをマウス・ドラッグして線を描く色を、presetオブジェクトを使って一瞬で変更できるだろう。ただし、lcdオブジェクトに描いた絵はプリセットできない。

### ■3-8-3 presetオブジェクトの利用例



presetオブジェクトのインスペクターでは、Save Prests With Patcherがチェックされていれば、presetの内容をパッチ・ファイルとして保存するため、パッチを再び開いたときも、同じ設定を利用することができる。この項目はデフォルトでチェックされているので、多くの場合はそのままよいだろう。Preset Name Codeはpresetの内容をプリセット・ファイルとして保存する際のファイル・タイプを4文字で指定する。パッチ内でpresetを利用するだけなら、この項目の設定は不要だ。

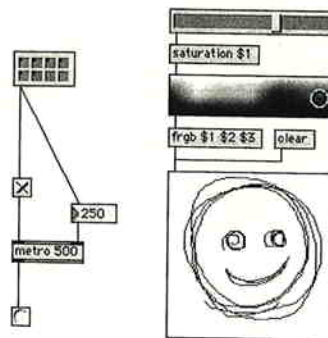
### ■3-8-4 presetオブジェクトのインスペクター



presetオブジェクトはパッチ・ウィンドウ内のすべてのユーザー・インターフェース・オブジェクトの設定を記憶するが、特定のユーザー・インターフェース・オブジェクトの設定だけを記憶させたいことがある。このような場合は、presetオブジェクトの第1アウトレットから、記憶させたいオブジェクトの第1インレットにパッチ・コードをつなぐことでパッチ・コードをつないだオブジェクトだけをプリセットの対象にできる。

逆に、特定のユーザー・インターフェース・オブジェクトの設定を記憶させたくない場合は、presetオブジェクトの第3アウトレットから、除外したいオブジェクトの第1インレットにパッチ・コードをつなげばよい。次のパッチでは、toggleオブジェクトとナンバー・ボックスの状態だけがpresetオブジェクトに記憶される。そのため、プリセットを切り替えてもhsliderやswatchは影響を受けない。

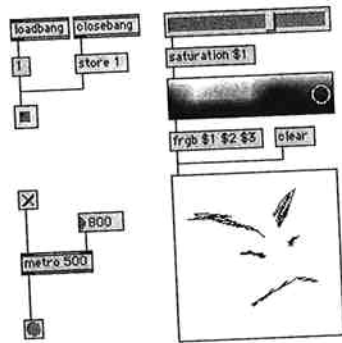
### ■3-8-5 特定のオブジェクトだけを記憶するpresetオブジェクトの対象にする



先に説明したloadbangとclosebangと組み合わせ、presetオブジェクトによってパッチの状態を自動保存すれば、再びパッチを開いたときに元の状態を再現することができる。このためには、closebangオブジェクトを使って、store 1というメッセージをpresetオブジェクトに送る。パッチが閉じられる際に自動的にstoreメッセージがpresetオブジェクトに送られるので、presetオブジェクトはユーザー・インターフェース・オブジェクトの設定を記憶する。記憶するプリセット・ボタンはstoreのアーギュメントで指定した番号になり、プリセットの内容はパッチ・ファイルとして保存される。

パッチを開くと、loadbangオブジェクトによって1というメッセージがpresetオブジェクトに送られる。整数を受け取ったpresetオブジェクトは、その番号のプリセットを呼び出す。このようにして、パッチを閉じたときの状態が再現されることになる。このようなパッチはHide On Lockを指定して隠蔽し、ユーザーが意識しないようにしてもよい。

■3-8-6 パッチの状態を自動保存し、再現する

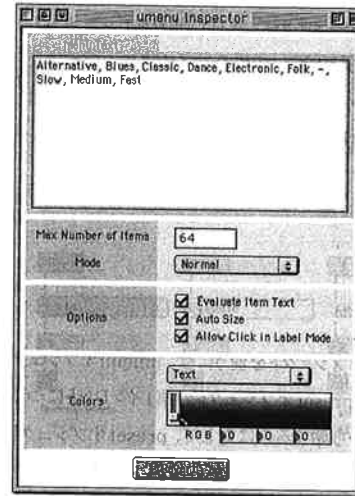


● umenuオブジェクトによるメニュー選択

presetオブジェクトを使って数多くのプリセットを記憶させると、どれがどのような設定であるかが分からなくなってしまいます。そこで活用したいのがポップアップ・メニューを実現するumenuオブジェクトだ。各プリセットに対応した名前を付けておけば、ポップアップ・メニューから簡単にプリセットを選べるようになる。もともと、umenuはプリセット専用のオブジェクトではなく、それ以外の用途もいろいろと考えられるだろう。

umenuオブジェクトはオブジェクト・パレットの後ろから3番目にある。パッチ・ウィンドウにumenuオブジェクトを作成すると、最初はメニュー項目を持たない空のポップアップ・メニューが作られる。そこで、umenuオブジェクトのインスペクターを開いて、Menu Text欄にメニュー項目を入力する。各メニュー項目は、(カンマ)で区切って入力すればよい。メニュー項目にはスペース(空白文字)を含めてもよい。また、-(マイナス記号)は、メニュー記号に区切り線を入れることになる。

■3-8-7 umenuオブジェクトのインスペクター



このようにして作成したumenuオブジェクトをクリックすれば、ポップアップ・メニューが表示され、いずれかの項目を選ぶことができる。ちなみに、umenuオブジェクトは表示フォントやサイズを設定可能だ。ただし、日本語フォントを指定しても一部の2バイト文字は文字化けするので、注意が必要である。

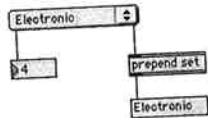
■3-8-8 umenuオブジェクトの表示





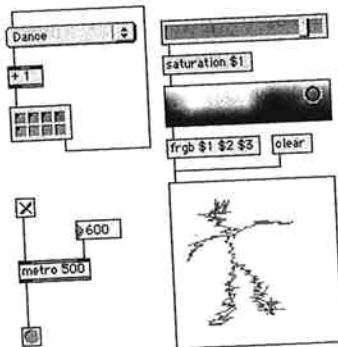
ポップアップ・メニューを選択すると、umenuオブジェクトは第1アウトレットからメニュー項目番号を出力する。この番号は0から始まるので、一番上のメニュー項目が選ばれたときは0が出力され、2番目のメニュー項目なら1が出力される。また、第2インレットからは選択されたメニュー項目の文字列がメッセージとして出力される。これはインスペクターのEvaluate Item Textがチェックされている場合に有効だ。

### ■3-8-9 umenuオブジェクトの出力



presetオブジェクトとともにumenuオブジェクトを使うには、umenuオブジェクトの第1アウトレットからの出力に1を加算してpresetオブジェクトの第1インレットにつなげばよい。umenuのメニュー項目番号が0から始まるのに対して、presetオブジェクトのプリセット番号は1から始まるからだ。また、プリセットから除外するオブジェクトとしてumenuを指定しておくべきだ。このようにしなければ、umenuオブジェクトで選ばれているメニュー項目自体もpresetオブジェクトが記憶してしまい混乱が生じる。具体的には、presetオブジェクトの第3アウトレットからumenuオブジェクトのインレットにつなぐ。

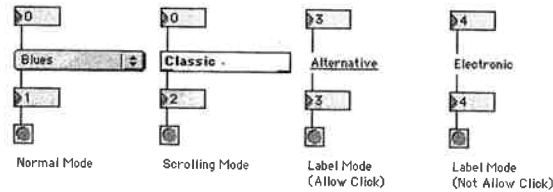
### ■3-8-10 umenuによるpresetの利用



## Chapter 3 ● プログラミングの手法

ところで、umenuオブジェクトのインスペクターでは、ModeとしてNormal、Scrolling、Labelが選べるようになっている。Normalは、これまで説明したポップアップ・メニューとして動作する。Scrollingではポップアップ・メニューは表示されず、マウス・ドラッグによって項目を選択するモードになる。そしてLabelではumenuの操作はできなくなり、インレットに受け取った数値に応じて表示を行うだけになる。ただし、インスペクターでAllow Click in Label Modeがチェックされている場合は、表示文字に下線が付き、それをクリックすれば項目番号やメッセージが出力される。

### ■3-8-11 umenuのモードによる表示の違い



なお、menubarオブジェクトを用いれば、メニュー・バーに独自のメニューとメニュー項目を表示し、その選択によって何らかの処理を行うこともできる。