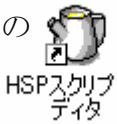
Ｂ－１　授業で用いた教材

スクリプト入門１　文字表示と繰り返し

～HSP 入門編（http://hsp.tv/make/enroll.html）から一部引用～

１．画面上の 

をクリックしてスクリプトを入力する画面を出します。

２．次の一行を入力しましょう。（mes の後ろは半角スペースが必要です。）

```
mes “私の名前は○○です”
```

３．F5 キーを押して実行してみましょう。画面に新しい Window が開き、その中に

```
私の名前は○○です
```

と表示されれば成功です。mes は文字を表示させる命令です。

４．次は、これを 10 回繰り返してみましょう。

```
repeat 10
mes "私の名前は○○です。"
loop
stop
```

スクリプトは repeat～loop で囲まれたプログラムを指定回数繰り返します。
stop はスクリプトの終わりを意味します。

５．次は繰り返し回数を表示してみましょう。４のプログラムをちょっと変えます。

```
kai=1
repeat 10
mes "繰り返し回数は"+kai+"回です"
kai=kai+1
loop
stop
```

kai=1 は初期設定といい、あらかじめ定まった数を入れておきます。
kai=kai+1 が実行されるたびに kai の値が 1 ずつ増えていきます。
"繰り返し回数は"+kai+"回です"　で ＋ は文字列をつなげる役割をしています。
ここで使った kai のようなものは変数とよばれます。

スクリプト入門２　制御と色

～HSP 入門編（http://hsp.tv/make/enroll.html）から一部引用～

１．次のスクリプトを入力して実行してみましょう（実行は F5 キー）

```
mes "どちらかのボタンを押してみよう"
        button "PUSH",*hata1
        button "BYE",*hata2
        stop
*hata1
        mes "「PUSH」を押しましたね"
        stop
*hata2
        mes "「BYE」を押しましたね"
        stop
```

button は画面にボタンを表示し、その後に書かれている命令を実行します。
ここでは *hata1 または *hata2 に書かれてあるスクリプトがボタンをクリックするたびに実行されます。ユーザーの要求によりプログラムの流れが制御されることになります。これが制御の基本です。

命令の書式は、button “表示される文字列”，実行されるスクリプト
これは覚えておきましょう。

TAB キーを利用して上記のように段をつけておけば、後でスクリプトを見直すときにとても楽になります。

２．文字に色をつけるスクリプトを入力して実行しましょう。（１の応用です）

```
        mes "どちらかのボタンを押してみよう"
        button "赤",*hata1
        button "青",*hata2
        stop
*hata1
        color 255,0,0
        mes "「赤」を押しましたね"
        stop
*hata2
        color 0,0,255
        mes "「青」を押しましたね"
        stop
```

スクリプトの制御構造は１とまったく同じです。新しく加わったカラーは、表示される文字の色を制御します。命令は、color 赤の輝度,緑の輝度,青の輝度を 0～255 の間で調整します。緑は 0,255,0、黒は 0,0,0、白は 255,255,255 です。紫は 255,0,255、水色は 0,255,255、黄色は 255,255,0 になります。中間的な色も出せます。

スクリプト入門３　プログラムをスクリプトで実行する

～HSP 入門編（http://hsp.tv/make/enroll.html）から一部引用～

１．Windows のプログラムをスクリプトで実行しよう

```
        pos 100,50
        mes "計算機"
        mes "メモ帳"
        mes "おしまい"
        pos 20,50
        button "CALC",*mkcalc
        button "MEMO",*mkmemo
        button "END",*owari
        stop
*mkcalc
         exec "calc"
        stop
*mkmemo
        exec "notepad"
        stop
*owari
        end
```

ここで pos は画面上での位置を表します。プログラムを実行するとボタンが window から少し離れた位置にきたはずです。

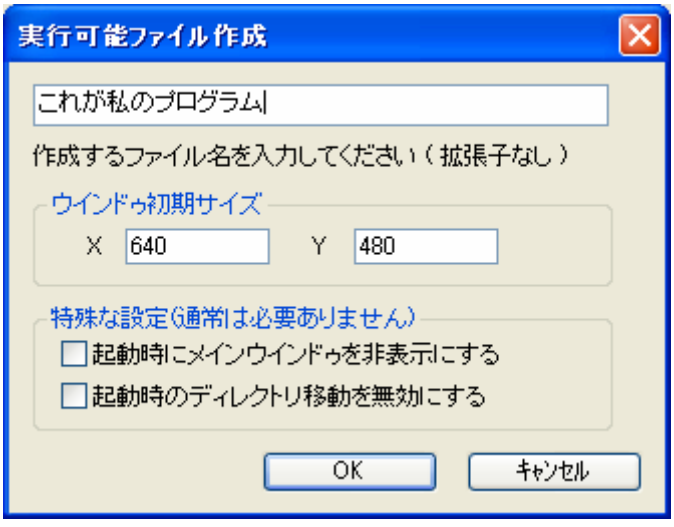
新しい命令語は、exec と end です。

exec の後ろに書いたプログラムが実行されます。exec “プログラム名”

end

２．実行可能形式のプログラムを作ってみよう（まず F9 を押してから始めましょう。）

メニューバーの ツール をクリックし、 EXE ファイル作成(M) をクリックします。



←この四角の中にプログラム名を入力します。ここでは「これが私のプログラム」と入力しました。この名前で実行形式ファイルができます。

ウィンドウ初期サイズはこのままで大丈夫です。OK をクリックすれば、MyDocuments の中に実行形式ファイルが作成されます。

スクリプト入門４　変数と条件判断

～HSP 入門編（http://hsp.tv/make/enroll.html）から一部引用～

１．変数を用いて条件判断をするスクリプトを動かそう

```
                kai=0
                mes "ボタンを押してね"
                button "PUSH",*gopush
                button "END",*goend
                stop
*gopush
                kai=kai+1
                stop
*goend
                if kai > 0：goto *owari
                mes "せめて１回は「PUSH」ボタンを押してよ"
                stop
*owari
                cls
                mes "あなたは、"+kai+"回「PUSH」ボタンを押しましたね"
                stop
```

最初に kai=0 で、kai の値を決めます。そして、PUSH ボタンが押されるたびに *gopush に書いてあるスクリプトが実行され､kai の値が１増えます｡したがって、PUSH ボタンを押さなければ、kai の値は 0 のままです。

END ボタンを押すと､*goend に書いてあるスクリプトが実行されます｡この時、kai の値が 0 であれば､「せめて１回は「PUSH」ボタンを押してよ」と表示され、kai の値が 0 以上であれば、*owari に書いてあるスクリプトが実行されます。

通常のプログラムは、このように変数によって何らかの制御を行うことが多く、エアコンや冷蔵庫などの温度制御は必ずこのようなプログラムを内蔵しています。

【発展１】ボタンを３回以上押さないとプログラムが終了しないようにスクリプトとメッセージを書き変えてみよう。

【発展２】四角形の４つの頂点の座標を x1,y1,x2,y2 としたとき四角形を書く命令は、boxf x1,y1,x2,y2 である。このとき、事前に color で指定された四角形が書かれる。これを応用して、ボタンを押すと

（１）徐々に大きさが変わる四角形

（２）徐々に色が変わる四角形

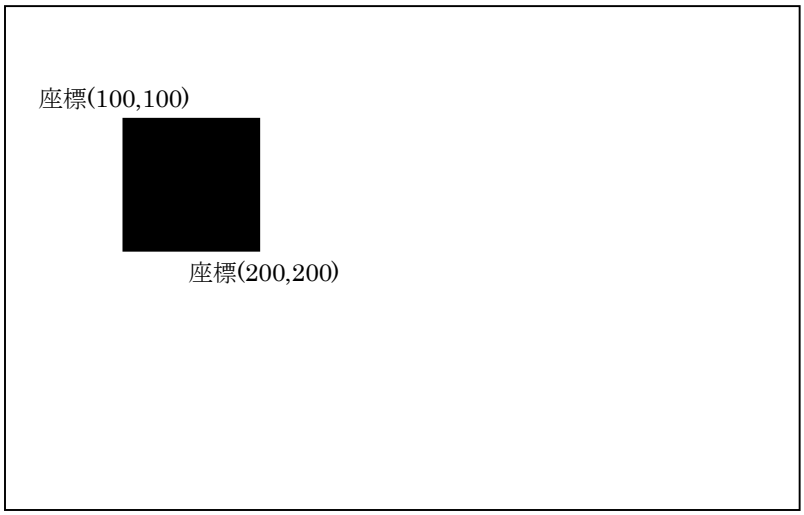
（３）徐々に大きさと色が変わる四角形

が描画されるようなプログラムを作りなさい。画面を消す命令は cls，動きが速すぎる場合は、スクリプトに１行 wait=1 を加えなさい。

スクリプト入門５　図を描いてみよう

１．　四角を描いてみよう

boxf 100,100,200,200



２．　色付き四角形を描こう

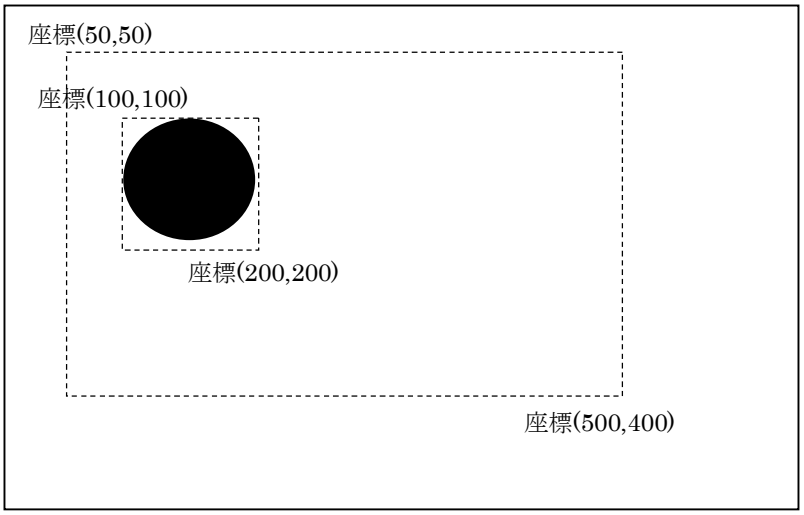
color 255,0,0
boxf 200,200,300,300
color 0,255,0
boxf 300,300,400,400
color 0,0,255
boxf 400,200,400,300

３．　円を描いてみよう

circle 100,100,200,200

座標で指定された四角形に内接する円が描かれます。
同じ要領で線も描くことができます。

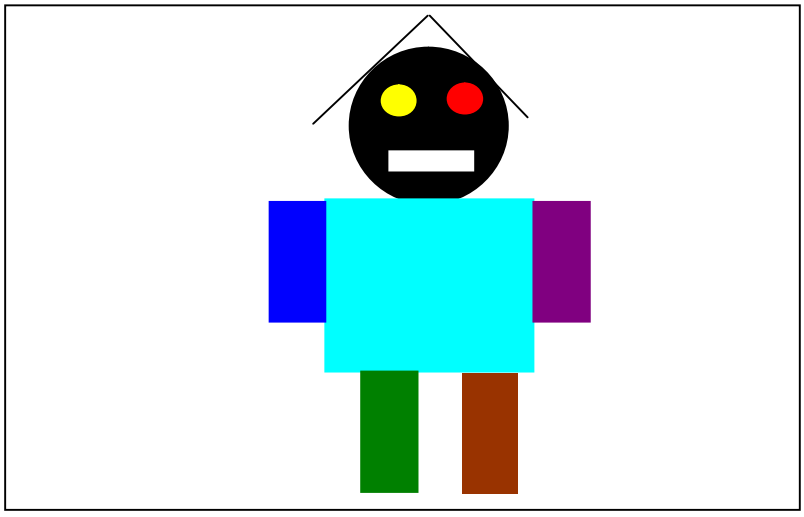
line 50,50,500,400



４．　ロボットを描いてみよう

紙の上で下書きをして、それぞれの命令に入れる座標を決めてからスクリプトを書き始めるとよい。

黒　color 0,0,0
青　color 0,0,255
赤　color 255,0,0
紫　color 255,0,255
緑　color 0,255,0
水　color 0,255,255
黄　color 255,255,0
白　color 255,255,255



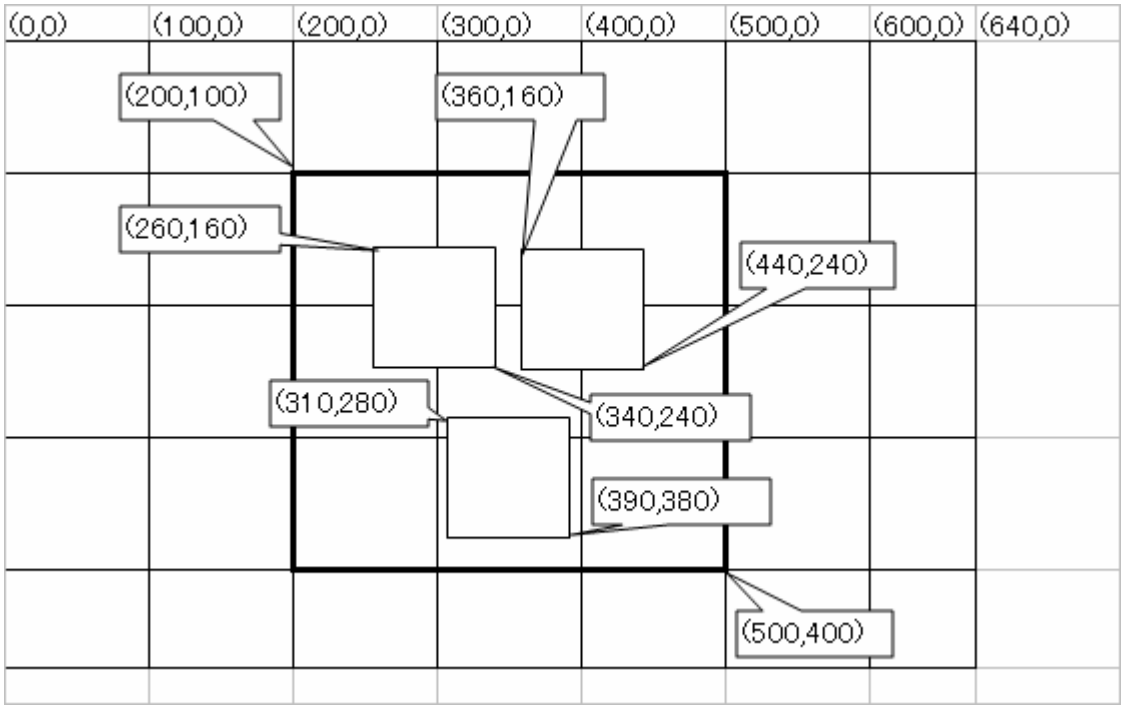
図形を描画するための準備作業

(0,0)	(100,0)	(200,0)	(300,0)	(400,0)	(500,0)	(600,0)	(640,0)

(0,0)	(100,0)	(200,0)	(300,0)	(400,0)	(500,0)	(600,0)	(640,0)

スクリプト入門６　ロボットにウィンクさせてみよう

１．用紙に図を描いて、それぞれの四角の左上の座標と右下の座標を記入する。



２．絵を描くスクリプトを書いてみよう。
（スクリプト入門５を参考に！）

応用課題１　ボタンを押したら絵が描かれるようにしてください。

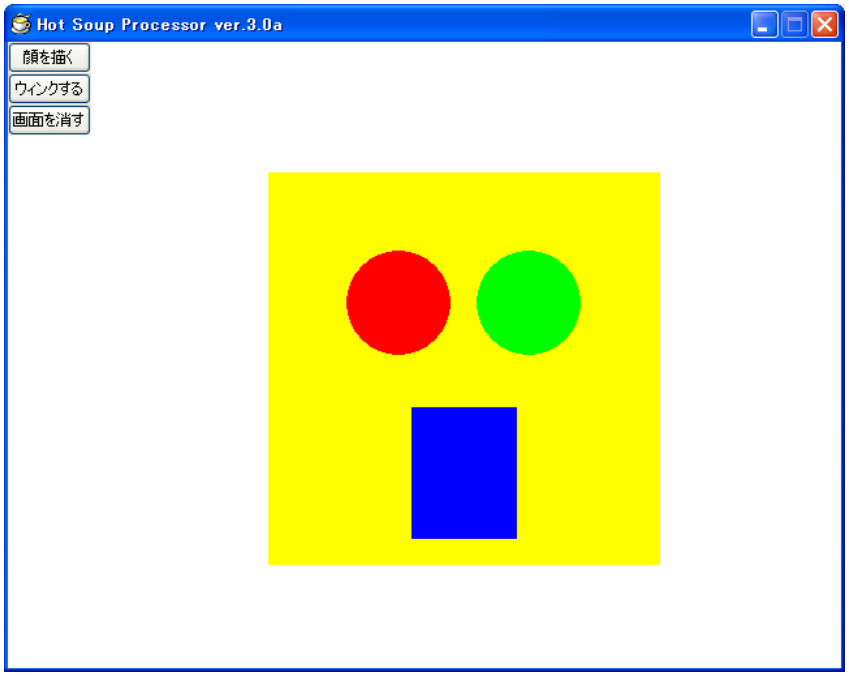
応用課題２　ボタンを押したら絵が動くようにしてください。
ヒント：背景と同じ色で塗りつぶしてから、書けばよい。
ヒント：しばらく待つ命令は　wait 100（数字が大きい方が長く待つ）

応用課題３　ボタンを押したら絵が消えるようにしてください。
ヒント：画面を消す命令は cls

応用課題４　絵が消えてもボタンが消えないようにしてください。
ヒント：画面を消してから、もう一度ボタンを書けばよい。

ロボットにウィンクさせてみよう（サンプルプログラム）

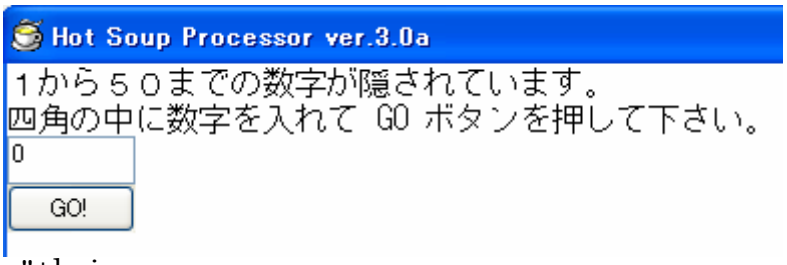
```
*sousa
        button "顔を描く",*kao
        button "ウィンクする",*wink
        button "画面を消す",*kesu
        stop
*kao
        ;顔
        color 255,255,0
        boxf 200,100,500,400
        ;左目
        color 255,0,0
        circle 260,160,340,240
        ;右目
        color 0,255,0
        circle 360,160,440,240
        ;口
        color 0,0,255
        boxf 310,280,390,380
        stop
*wink
        ;左目を消す
        color 255,255,0
        circle 260,160,340,240
        ;ちょっと待つ
        wait 100
        ;左目を描く
        color 255,0,0
        circle 260,160,340,240
        stop
*kesu
        cls
        goto *sousa
        stop
```



スクリプト入門７　数当てゲームで遊ぼう

１．次のスクリプトを入力して実行してください。

```
randomize
kotae=rnd(51)
mes "１から５０までの数字が隠されています。"
mes "四角の中に数字を入れて GO ボタンを押して下さい。"
kai=1
input kazu
button "GO!",*go
stop
*go
mes "Number of challenge="+kai
if kazu=kotae {mes "当たり"}
if kazu>kotae {mes "数字が大きい"}
if kazu<kotae {mes "数字が小さい"}
kai=kai+1
stop
```



２．何回かチャレンジして少ない回数で数が当てられるように練習してください。

３．別の用紙に自分が数を当てるために行っている動作や判断を、フローチャートの形で書いて提出してください。フローチャートについては、教科書の「アルゴリズム」のページを読んで理解して下さい。

【命令解説】

randomaize	ランダムな数を発生させるための準備です。
kotae=rnd(51)	kotae に 1～50 までのランダムな数を入力します。
input kazu	kazu に数字を入力してもらいます
if kazu=kotae {mes”当たり”}	kazu と kotae が一致したら「当たり」と表示します。
kai=kai+1	kai を 1 増やします。

数当てゲームについての機械側と人間側のアルゴリズム

１９H　　番　名前

教科書をよく読み、できるだけ詳しく書きましょう。

機械側のフローチャート	人間側のフローチャート

スクリプト入門８　音声を録音して再生しよう

１．マイクの接続
　ヘッドホンマイクまたはマイクをコンピュータに接続します。
　ディスプレイのどこかに端子があります。
　コンピュータ本体まで途中の線がつながっているか、よく確認してください。

２．サウンドレコーダーの起動。
　「スタート」→「プログラム(P)」→「アクセサリ」→「マルチメディア」
　→「サウンドレコーダー」

３．録音・再生・保存の使い方

録音ボタン「●」をクリックします。
位置や長さの下の秒数が動いたら、マイクに向かってしゃべります。(10 秒くらい)
「■」の停止ボタンを押して、停止します。
再生ボタンを押すと再生します。
ファイルは MyDocuments に保存しましょう。
やり直す場合は、ファイル→新規作成。
※ボリュームなどの調整が必要になる場合もあります。
※エフェクタを使えば自分の声を高くしたり、低くしたり逆転再生したりできます。

４．次の音声ファイルを作成して、MyDocuments に保存してください。
　「私の名前は」　　ファイル名　my.wav
　「○○」　　ファイル名　name.wav　　ここで　.wav は勝手につきます。
　「です」　　ファイル名　desu.wav　　自分では入力しないでください。

５．HSP で音声を扱う場合は、次のようにします。mmload で音声ファイルを読み込み、mmplay で音声ファイルを再生します。読み込むファイルには番号をつけます。
　「私の名前は」を再生するには
　　mmload “my.wav”,1　　“my.wav”を１番に読み込む。
　　mmplay 1　　１番の音声ファイルを再生する。

課題１　「私の名前は」、「○○」、「です」の３つのボタンを作り、それをクリックした時に、それぞれの音声が再生されるようなスクリプトを作りなさい。

課題２　数当てゲームで「数字が大きい」「数字が小さい」「当たり」に相当する音声ファイルを作り、メッセージが表示されると同時に音声が再生されるようにしなさい。

音声の読み込みと再生サンプルスクリプト

【私の名前は○○です】

```
;音声ファイルを読み込む
    mmload "my.wav",1
    mmload "name.wav",2
    mmload "desu.wav",3
;ボタンを表示
    button "私の名前は",*my
    button "○○",*name
    button "です",*desu
    stop
;音声ファイルを再生
*my
    mmplay 1
    stop
*name
    mmplay 2
    stop
*desu
    mmplay 3
    stop
```

【数当てゲーム音声つき】

```
*jyunbi
    mmload "madamada.wav",1
    mmload "yatta.wav",2
*main
    randomize
    kotae=rnd(51)
    mes "１から５０までの数字が隠されています。"
    mes "四角の中に数字を入れて GO ボタンを押して下さい。"
    kai=1
    input kazu
    button "GO!",*go
    stop
*go
    mes "Number of challenge="+kai
    if kazu=kotae {mes "当たり":mmplay 2}
    if kazu>kotae {mes "数字が大きい":mmplay 1}
    if kazu<kotae {mes "数字が小さい":mmplay 1}
    kai=kai+1
    stop
```

※ １行に複数の命令を書く場合、その間をコロンで区切ります。
例：mes “○○”:mmplay 1