グラフィック関係のオブジェクト

次の図は**画面の座標**です。画面の中心が原点で、画面の位置はX軸とY軸で表されます。座標の値は画面のピクセルと対応しています。起動したときの大きさは、横800×縦550 程度です。向きは右方向が0度で、左回りに指定します。

タートル

- 画面を歩き回るキャラクタです。動くと軌跡の線が残るので、それを利用して図形を描くことができます（タートルグラフィックス）。
- 描いた線は「図形を作る」でタートルから切り離して**図形**オブジェクトにできます。
- 「**衝突**」というメソッドを定義しておくと、他のオブジェクトと重なったときに実行されます。衝突の判定に使われるのは、タートルの本体です。描いた線を判定に使いたい場合は、図形オブジェクトにしてください。
- 衝突メソッドの1個目のパラメータには、重なった相手のオブジェクトが渡されます。2個目のパラメータには、自分が動いて重なったかどうかが真偽値で渡されます。
- 「衝突」に「**タートル：跳ね返る**」を代入することで、自然な跳ね返りを定義できます。衝突する相手が図形オブジェクトの場合、図形オブジェクトを描き始めたときの方向が衝突する際の壁の角度として扱われます。
    - （例）壁とぶつかった向きとは関係なく、常に斜め後ろを向きます。

```
かめた：衝突＝「自分！150　右回り」。
```

- （例）壁とぶつかった向きに応じて、自然な角度で跳ね返ります。

```
かめた：衝突＝タートル：跳ね返る。
```

- 「**動作**」というメソッドを定義すると、マウスでクリックしたときに実行されます。
- **作る**: 新しいタートルを作ります。
    - （例）タートルを作り「かめた」という名前にします。

```
かめた＝タートル！作る。
```

- **歩く**: 前に進みます。
    - （例）かめたは100歩、歩きます。

```
かめた＝タートル！作る。
かめた！100歩　歩く。
```

- **戻る**: 後ろに戻ります。
    - （例）かめたは100歩、戻ります。

```
かめた＝タートル！作る。
かめた！100歩　戻る。
```

- **右回り**: 右に回ります。
    - （例）かめたは90度、右に回ります。

```
かめた＝タートル！作る。
```

```
かめた！90度　右回り。
```

- **左回り**：左に回ります。
  - （例）かめたは90度、左に回ります。

```
かめた＝タートル！作る。
かめた！90度　左回り。
```

- **移動する**：右にx歩、上にy歩動きます。
  - （例）かめたは「右に0歩、上に100歩」動きます。

```
かめた＝タートル！作る。
かめた！0　100　移動する。
```

- **位置**：指定された座標に移動します。画面の中央が中心（0,0）です。
  - （例）かめたは「（100, 100）」の座標の位置に動きます。

```
かめた＝タートル！作る。
かめた！100　100　位置。
```

- **向き**：向きを指定します。右向きが0度です。角度は左回りに大きくなります。
  - （例）かめたは90度（上方向）を向きます。

```
かめた＝タートル！作る。
かめた！90度　向き。
```

- **ペンなし**：動くときに軌跡が残らないようにします。
  - （例）かめたは線を描かずに歩きます。

```
かめた＝タートル！作る。
かめた！ペンなし。
かめた！100歩　歩く。
```

- **ペンあり**：動くときに軌跡が残るようにします。[1)]
  - （例）かめたは線を描かずに歩いた後、線を描きながら歩きます。

```
かめた＝タートル！作る。
かめた！ペンなし。
かめた！100歩　歩く。
かめた！ペンあり。
かめた！100歩　歩く。
```

- **中心に戻る**：画面の真ん中に戻ります。
  - （例）かめたは画面の中心（「（0, 0）」の座標の位置）に動きます。

```
かめた＝タートル！作る。
かめた！100歩　歩く。
かめた！中心に戻る。
```

- **閉じる**：描き始めの点まで線を引きます。
  - （例）かめたは三角形の2つの辺を描いた後、描き始めた点まで線を引いて戻ります。

```
かめた＝タートル！作る。
かめた！100　歩く　120　右回り　100　歩く　閉じる。
```

- **図形を作る**：軌跡の線を自分から切り離して図形オブジェクトにします。
  - （例）かめたが描いた線を図形オブジェクトにして「三角」という名前にします。

```
かめた＝タートル！作る。
かめた！100　歩く　120　右回り　100　歩く。
三角＝かめた！図形を作る。
```

色を指定するとその色で塗られます。

- （例）かめたが描いた線を黄色の色を塗った図形オブジェクトにして「三角」という名前にします。

```
かめた＝タートル！作る。
かめた！100　歩く　120　右回り　100　歩く。
三角＝かめた！（黄）図形を作る。
```

- **拡大する**：タートルを拡大します。

□n 拡大する」でn倍に拡大します□nは正の整数です。

- （例）かめたを「縦横2倍」に拡大します。

```
かめた＝タートル！作る。
かめた！2　拡大する。
```

□m n 拡大する」で、「横にm倍、縦にn倍」に拡大します□m, nは正の整数です。

- （例）かめたを横に3倍、縦に2倍に拡大します。

```
かめた＝タートル！作る。
かめた！3　2　拡大する。
```

- **線の色**：描く線の色を変えます。はじめの色は黒です。
  - （例）かめたが描く線の色を「緑」に設定します。

```
かめた＝タートル！作る。
かめた！（緑）線の色　100　歩く。
```

- **線の太さ**：描く線の太さを変えます。はじめの太さは3です。
  - （例）かめたが描く線の太さを「5」に設定します。

```
かめた＝タートル！作る。
かめた！5　線の太さ。
かめた！100　歩く。
```

- **消える**：姿を消します。描いている線も消えます。
  - （例）かめたを画面から消します。

```
かめた＝タートル！作る。
かめた！消える。
```

- **現れる**：画面に現れます。消えた姿を戻すときに使います。
  - （例）消えているかめたを表示します。「消える」の例に続いて実行してください。

```
かめた！現れる。
```

- **手前に表示**：他のタートルや図形オブジェクトより手前に表示します□(Java版のみ)
  - （例）かめたを他のオブジェクトより手前に表示します。

かめた！手前に表示。

- **円**：指定した半径の円を描きます[2)]。大きさが正の場合はタートルの右側に、負の場合は左側に描かれます。
  - （例）かめたが半径100の円を描きます。

```
かめた＝タートル！作る。
かめた！100　円。
```

- **角形**：正n角形を描きます□□m n 角形」で、1辺が長さmの図形（正三角形、正方形、正五角形など）を描きます。図形はタートルの右側に描かれますが、辺の長さmを負にすると左側に描かれます。
  - （例）かめたが右側に1辺が100の三角形と、左側に1辺が100の五角形を描きます。

```
かめた＝タートル！作る。
かめた！100　3　角形。
かめた！-100　5　角形。
```

- **向き？**：タートルの向きを調べます。右向きが0度です。角度は左回りに大きくなります。
  - （例）上を向いたかめたの向きを表示します。「90」が表示されます。

```
かめた＝タートル！作る。
かめた！90　左回り。
ラベル！（かめた！向き？）作る。
```

- **横の位置？**：タートルのX座標を調べます。
  - （例）かめたのX座標を表示します。「50」が表示されます。

```
かめた＝タートル！作る。
かめた！60　左回り。
かめた！100　歩く。
ラベル！（かめた！横の位置？）作る。
```

- **縦の位置？**：タートルのY座標を調べます。
  - （例）かめたのY座標を表示します。「50」が表示されます。

```
かめた＝タートル！作る。
かめた！30　左回り。
かめた！100　歩く。
ラベル！（かめた！縦の位置？）作る。
```

- **変身する**：タートルの姿を変えます。
  - 以下の画像はあらかじめ用意されています。下の一覧の画像名に「.png□をつけて利用してください□□tulip□は「tulip.png□がファイル名です）。
  - 独自の画像を利用する場合は、画像ファイル□png,jpg,gifのいずれか）をプログラムファイル（現在編集中のdtlファイル）のあるフォルダ、またはドリトルをインストールしたdolittle.jarのあるフォルダに置いてください。
  - （例）かめたの姿を「tulip.png□という画像に変更します。

```
かめた＝タートル！作る。
かめた□"tulip.png"□変身する。
```

- 画像をURLで指定します。
- （例□URLを指定して画像を変更します。

```
かめた＝タートル！作る。
```

```
かめた□"https://dolittle.eplang.jp/_media/logo.png"□変身する。
```

- 画像をTwitterのアイコンで指定します。[3]□Java版のみ）
- （例）かめたの姿を「@watayan□というユーザーの画像に変更します。

```
かめた＝タートル！作る。
かめた□"@watayan"□変身する。
```

図形

- タートルグラフィックスで描かれた図形を独立した図形オブジェクトにできます。
- 図形を画面の上で移動したり回転させることができます。
- 回転の中心は図形を描いたときの始点です。
- 他のオブジェクトと重なると「衝突」というメソッドが実行されます。
- □動作」というメソッドを定義すると、マウスでクリックしたときに実行されます。

- **作る**：自分を複製して新しい図形を作ります。
  - （例）「三角形」を複製して「三角形2」を作り、画面上で移動します。

```
かめた＝タートル！作る。
三角形＝「かめた！100　歩く　120　左回り」!3　繰り返す（赤）図形を作る。
三角形2＝三角形！作る。
三角形2！150　0　移動する。
```

- **右回り**：右に回ります。
  - （例）三角形は10度、右に回ります。

```
かめた＝タートル！作る。
三角形＝「かめた！100　歩く　120　左回り」!3　繰り返す（赤）図形を作る。
三角形！10度　右回り。
```

- **左回り**：左に回ります。
  - （例）三角形は10度、左に回ります。

```
かめた＝タートル！作る。
三角形＝「かめた！100　歩く　120　左回り」!3　繰り返す（赤）図形を作る。
三角形！10度　左回り。
```

- **移動する**：右にx歩、上にy歩動きます。
  - （例）三角形は「右に0歩、上に100歩」動きます。

```
かめた＝タートル！作る。
三角形＝「かめた！100　歩く　120　左回り」!3　繰り返す（赤）図形を作る。
三角形！0　50　移動する。
```

- **位置**：指定された座標に移動します。画面の中央が中心（0,0）です。
  - （例）三角形は「（100, 100）」の座標の位置に動きます。

```
かめた＝タートル！作る。
三角形＝「かめた！100　歩く　120　左回り」!3　繰り返す（赤）図形を作る。
三角形！100 100 位置。
```

- **塗る**: 自分の中を色で塗ります。色は色オブジェクトを括弧で囲んで書きます。色を省略すると線の色で塗られます。
    - （例）三角形を「青」で塗ります。

```
かめた＝タートル！作る。
三角形＝「かめた！100　歩く　120　左回り」!3　繰り返す（赤）図形を作る。
三角形！（青）　塗る。
```

- **拡大する**: 自分をn倍します。または縦横にx倍y倍します。
    - （例）三角形を「縦横2倍」に拡大します。

```
かめた＝タートル！作る。
三角形＝「かめた！100　歩く　120　左回り」!3　繰り返す（赤）図形を作る。
三角形！2　拡大する。
```

- （例）三角形を横に3倍、縦に2倍に拡大します。

```
かめた＝タートル！作る。
三角形＝「かめた！100　歩く　120　左回り」!3　繰り返す（赤）図形を作る。
三角形！3　2　拡大する。
```

- **消える**: 画面から消えます。
    - （例）三角形を画面から消します。

```
かめた＝タートル！作る。
三角形＝「かめた！100　歩く　120　左回り」!3　繰り返す（赤）図形を作る。
三角形！消える。
```

- **現れる**: 画面に現れます。消えた姿を戻すときに使います。
    - （例）消えている三角形を表示します。「消える」の例に続けて実行してください。

```
三角形！現れる。
```

- **手前に表示**: 他のタートルや図形オブジェクトより手前に表示します。
    - （例）三角形を他のオブジェクトより手前に表示します。

```
三角形！手前に表示。
```

- **向き？**: 向きを調べます。右向きが0度です。角度は左回りに大きくなります。
    - （例）三角形の向きを表示します。「90」が表示されます。

```
かめた＝タートル！作る。
三角形＝「かめた！100　歩く　120　左回り」!3　繰り返す（赤）図形を作る。
三角形！90 左回り。
ラベル！（三角形！向き？）作る。
```

- **結合する**: 複数の図形を結合して組図形を作ります。
    - （例）三角形と四角形を結合して「家」という名前の組図形を作ります。

```
家＝図形！（三角形）（四角形）結合する。
家！90　左回り。
```

組図形

- 複数の図形をひとつのオブジェクトとして扱えます。

- 図形の結合するメソッドで作られます。
- 要素の図形は組図形を作った後も独立した図形として扱えます。
- 回転の中心は要素の座標の中心です。
- 図形とほぼ同等のメソッドを実行できます。
- 個々の図形に定義された「衝突」イベントは要素の図形同士の衝突が発生する可能性があります。
- 以下の例では、次のように「家」という組図形を作ってから実行してください。

```
かめた＝タートル！作る。
屋根＝「かめた！100　歩く　120　左回り」！3　繰り返す（赤）図形を作る。
壁＝「かめた！100　歩く　90　右回り」！4　繰り返す（緑）図形を作る。
家＝図形！（屋根）（壁）結合する。
家！90　左回り。
```

- **作る**：自分を複製して新しい組図形を作ります。
  - （例）「家」を複製して「家2」を作り、画面上で移動します <code> 家2＝家！作る。
    家2！150　0　移動する </code>
- **右回り**：右に回ります。
  - （例）家は30度、右に回ります <code> 家！30度　右回り。 </code>
- **左回り**：左に回ります。
  - （例）家は30度、左に回ります <code> 家！10度　左回り。 </code>
- **移動する**：右にx歩、上にy歩動きます。
  - （例）家は「右に0歩、上に100歩」動きます <code> 家！0　50　移動する </code>
- **位置**：指定された座標に移動します。画面の中央が中心（0,0）です。
  - （例）家は「（100, 100）」の座標の位置に動きます <code> 家！100　100　位置 </code>
- **消える**：画面から消えます。
  - （例）家を画面から消します <code> 家！消える </code>
- **現れる**：画面に現れます。消えた姿を戻すときに使います。
  - （例）消えている家を表示します。「消える」の例に続けて実行してください <code>家！現れる </code>
- **向き？**：向きを調べます。右向きが0度です。角度は左回りに大きくなります。
  - （例）家の向きを表示します。「90」が表示されます <code> 家！90 左回り。 ラベル！（家！向き？）作る。 </code>
- 以下はJava版のみ
  - **塗る**：要素の図形を色で塗ります。
  - （例）家を「青」で塗ります <code> 家！（青）　塗る。 </code>
  - **拡大する**：自分をn倍します。または縦横にx倍y倍します。
  - （例）家を「縦横2倍」に拡大します <code> 家！2　拡大する </code>
  - （例）家を横に3倍、縦に2倍に拡大します <code> 家！3　2　拡大する </code>
  - **手前に表示**：他のタートルや図形オブジェクトより手前に表示します。
  - （例）家を他のオブジェクトより手前に表示します <code> 家！手前に表示 </code>

画面

- 実行画面を表すオブジェクトです。
- **マウス**という名前でも参照できます。

- **塗る**：画面の背景に色を塗ります。
  - （例）画面の背景を「水色」にします。

```
画面！（水色）塗る。
```

- **幅？**: 画面の幅を返します。
  - （例）画面の横幅を表示します。

```
ラベル！（画面！幅？）作る。
```

- **高さ？**: 画面の高さを返します。
  - （例）画面の高さを表示します。

```
ラベル！（画面！高さ？）作る。
```

- 以下はJava版のみ
  - **横の位置？**: マウスカーソルのX座標を返します。
  - （例）マウスカーソルのX座標を表示します。

```
ラベル！（マウス！横の位置？）作る。
```

- **縦の位置？**: マウスカーソルのY座標を返します。
- （例）マウスカーソルのY座標を表示します。

```
ラベル！（マウス！縦の位置？）作る。
```

- **背景画像**: 画面の背景に画像を表示します。
- （例）画面の背景に「a.png」という画像を表示します。

```
画面！"a.png" 背景画像。
```

- **方眼紙**: 画面に方眼紙の罫線を表示します。色を指定すると、その色の罫線が描かれます。色を指定しないと罫線が消えます。
- （例）画面に「緑」の罫線を表示します。

```
画面！（緑）方眼紙。
```

- （例）画面から罫線を消します。

```
画面！方眼紙。
```

色

- 色を表すオブジェクトです。
- よく使う8色は「黒、赤、緑、青、黄色、紫、水色、白」という変数で用意されています。
- 複数の色を混ぜ合わせるときは、パレットオブジェクトを使います。

- **作る**: 三原色を指定して色を作ります。赤緑青の順に0～255の値を指定します。
  - （例）「赤が255、緑が136、青が255」の明るさの色を作ります。

```
ピンク＝色！255　136　255　作る。
画面！（ピンク）方眼紙。
```

色を16進の数値で指定することもできます。赤、緑、青の明るさを、それぞれ2桁で指定します。

- （例）「赤が0xFF、緑が0x88、青が0xFF」の色を作ります。

```
ピンク＝色！0xFF88FF　作る。
```

```
画面！（ピンク）方眼紙。
```

- **ランダムに作る**: 色をランダムに作ります。
  - （例）色をランダムに作ります。実行するたびに違う色で表示されます。

```
新しい色＝色！ランダムに作る。
画面！（新しい色）方眼紙。
```

- **暗くする**: 色を暗くします。
  - （例）暗い緑色を作ります。

```
濃い緑＝緑！暗くする。
画面！（濃い緑）方眼紙。
```

- **明るくする**: 色を明るくします。暗くした色を再び明るくします。
  - （例）暗くした色を明るくします。「暗くする」の例に続いて実行してください。

```
明るい緑＝濃い緑！明るくする。
画面！（明るい緑）方眼紙。
```

- **半透明にする**: 色を半透明にします。裏側が透けて見える半透明の色を作ります。
  - （例）青い半透明の色を作ります。

```
新しい色＝青！ 半透明にする。
画面！（水色）塗る。
あぶく＝タートル！作る 30 円（新しい色）図形を作る。
タイマー！作る「あぶく！0 2 移動する」実行。
```

パレット

- 色を混ぜ合わせるオブジェクトです。
- あらかじめ「光」「絵具」という2個のオブジェクトが用意されています。
- 「光」は、光を重ねたときの色（**加法混色**）を作ります。
- 「絵具」は、絵の具を重ねたときの色（**減法混色**）を作ります。

- **混ぜる**: 複数の色を混ぜます。
  - （例）「赤」と「緑」の光を混ぜた色を作ります。

```
新しい色＝光！（赤）（緑）混ぜる。
画面！（新しい色）塗る。
```

- （例）「赤」と「緑」の絵具を混ぜた色を作ります。

```
新しい色＝絵具！（赤）（緑）混ぜる。
画面！（新しい色）塗る。
```

1)
V2.1から、「ペンあり」を実行しても、それまでに描いた線が図形として切り離されないようになりました。V2.0までのプログラムで「ペンあり」で線を切り離していた場合は、必要に応じてプログラムを修正してください。

2)
実体は36角形です。

3)
風柳メモの「TwitterアイコンURL取得API」を利用させていただいています。http://d.hatena.ne.jp/furyu-tei/20130730/1375178609

From:
https://dolittle.eplang.jp/ - **プログラミング言語「ドリトル」**

Permanent link:
https://dolittle.eplang.jp/ref_graphics33?rev=1578401671

Last update: **2020/01/07 21:54**