初版作成　2019.12.22

最終修正　2024.1.9

5.　角度の扱い方

薮哲郎

# scratchにおける角度

　　　

　　　　　　○○度に向ける　　　　　　　　　　　　　　　　三角関数

　scratchには角度の取り方が2種類あります。

　「○度に向ける」「○歩動かす」というブロックを使うときは左上図のように角度をとります。この角度の取り方は時計の針の角度に合わせていると思われます。

　「10のsin」のように三角関数のブロックを使うときは、右上図のように角度をとります。高校数学で三角関数を扱うときの角度と同じです。ただし、高校数学では三角関数の引数はラジアン角（1周を2とする）で指定しますが、scratchでは1周を360度とする角度を用います。

　角度は通常は 179度～180度の範囲で指定します。この範囲外の値を指定したときは、scratchが360の倍数を引くか足すかして、範囲内の値に直してくれます。

# 三角関数

　三角関数の引数の範囲と返り値の範囲は以下の通りです。角度は「ラジアン」ではなく「度」で指定します。「度」で指定した角度が下記の表の範囲を超えたとき、sin, cosについては範囲内におさまるように、360×n度を「引くor足す」という操作が行われます。tanについては180×n度を「引くor足す」という操作が行われます。

|  |  |  |
| --- | --- | --- |
| 関数名 | 引数 | 返り値 |
| sin, cos | 180～180 | 1～1 |
| tan | 90～90 | ∞～∞ |
| asin, acos | 1～1 | 90～90 |
| atan | ∞～∞ | 90～90 |

# atanの使い方

　「あるスプライトが別のスプライトを追いかける」という動作をさせたいとき、角度は180度～180度の360度の範囲を扱う必要があります。



　図のように追いかけるスプライトが (0, 0) にいて、追いかけられるスプライトが (*x*1, *y*1) にいる場合を考えます。座標値の正負と角度θは以下の関係があります。

|  |  |  |  |
| --- | --- | --- | --- |
| 状態 | *x*1 | *y*1 | 角度 |
| (1) | 正 | 正 | 0～90 |
| (2) | 負 | 正 | 90～180 |
| (3) | 負 | 負 | 180～90 |
| (4) | 正 | 負 | 90～0 |

　C, Fortran, pythonなどのプログラミング言語には、atan2(x, y) という関数があります(注：pythonはMath.atan2(y,x) であり引数の順序が逆)。xとyの符号を利用することで、返り値は360度の範囲をとります（180度～180度になる）。scratchにはatan2はなく、引数が1つのatanしか用意されていません。引数が1個なので「$y\_{1}/x\_{1}$」を関数に与えます。ゆえに「(1)と(3)」の区別ができず、「(2)と(4)」の区別もできません。atanの返り値は 90～90であり、180度の範囲しか持ちません。360度の範囲の値を得るために、以下のようにプログラムを組む必要があります。



　スプライトAがスプライトBを追いかけるスクリプトは以下のように組みます。

　　　　

　　　　　　　スプライトB　　　　　　　　　　　　スプライトA

　スプライトBは人間が操作することを仮定するので、キー入力を入れています。1回の移動距離はdist\_escapeです。キー入力の無限ループを組み、スプライトBのx座標とy座標をそれぞれ変数x, y に入れます。

　スプライトAの移動方向はスプライトBの場所です。前ページの図を言葉で書くと、以下のようになります。

1. x軸の差をdxに入れ、y軸の差をdyに入れます。前ページの図ではdx, dyをそれぞれx1, y1と表記していました。
2. 角度（360度の範囲を持つ）を算出してthetaに入れます。
3. 1回の移動距離をdist\_chaseとすると、x方向の移動距離は $dist\\_chase×\cos( θ)$, y方向の移動距離は $dist\\_chase× sin θ$ です。