utils

- dist()
- dot()
- min-by()
- normalise()
- to-angle()
- vec-diff()
- vec-sum()

dist

A helper function to calculate the distance between two points.

Parameters

```
dist(
  pt1: array,
  pt2: array,
  kind
) -> int float decimal
```

pt1 `array`

The first point.

pt2 `array`

The second point.

dot

A helper function to calculate the dot product of two vectors.

Parameters

```
dot(
  vec1: array,
  vec2: array
) -> int float decimal
```

vec1 `array`

The first vector.

vec2 `array`

The second vector.

min-by
A filter function that returns the minimum value based on a given function.

Parameters

```
min-by(
  values: array,
  func: function,
  default
) -> any
```

values `array`

The values to filter.

func `function`

The function to filter by.

normalise
A helper function to normalise a vector.

Parameters

```
normalise(
  vec: array,
  to: int float decimal
) -> array
```

vec `array`

The vector to normalise.

to `int` or `float` or `decimal`

The length of the vector.

Default: 1

to-angle
A helper function to convert types into an angle.

Parameters

```
to-angle(value: angle ratio int float decimal) -> angle
```

value `angle` or `ratio` or `int` or `float` or `decimal`

The rotation of the hexagon.

vec-diff
A helper function to calculate the difference of vectors.

Parameters

```
vec-diff(
  vec1: array,
  ..vecs: array
) -> array
```

vec1 `array`

The vector to subtract from.

..vecs `array`

The vector to subtract.

vec-sum
A helper function to calculate the sum of vectors.

Parameters

```
vec-sum(..vecs: array) -> array
```

..vecs `array`

The vector to sum.

coordinates

- to-euclidean()

to-euclidean
Transforms an axial coordinate to an euclidean coordinate.

Parameters

```
to-euclidean(
  r,
  q
) -> tuple
```