Documentation Language: Swift

Vision / NormalizedRect

Structure

NormalizedRect

The location and dimensions of a rectangle.

iOS 18.0+ | iPadOS 18.0+ | macOS 15.0+ | tvOS 18.0+ | visionOS 2.0+

```
struct NormalizedRect
```

Topics

Creating a normalized rectangle

`init(x: CGFloat, y: CGFloat, width: CGFloat, height: CGFloat)`

Creates a rectangle with the specified coordinates.

`init(imageRect: CGRect, in: CGSize)`

Creates a normalized rectangle from a rectangle in an image coordinate space.

`init(imageRect: CGRect, in: CGSize, normalizedTo: NormalizedRect)`

Creates a rectangle normalized to a region of interest in an image from a rectangle in an image coordinate space.

`init(normalizedRect: CGRect)`

Creates a rectangle from the specified Core Graphics rectangle.

`static var fullImage: NormalizedRect`

A normalized rectangle with an origin at zero and a width and height of one.

Inspecting a normalized rectangle

```
let cgRect: CGRect
```

The normalized rectangle as a Core Graphics rectangle.

```
var origin: CGPoint
```

The lower left-hand corner of the rectangle.

```
var width: CGFloat
```

The width of the rectangle.

```
var height: CGFloat
```

The height of the rectangle.

Converting rectangles

```
func toImageCoordinates(CGSize, origin: CoordinateOrigin) -> CGRect
```

Converts a rectangle in normalized coordinates into image coordinates.

```
func toImageCoordinates(from: NormalizedRect, imageSize: CGSize, origin: CoordinateOrigin) -> CGRect
```

Converts a rectangle normalized to a region within an image into full image coordinates.

Flipping a normalized rectangle

```
func verticallyFlipped() -> NormalizedRect
```

Returns a normalized rectangle with the origin flipped between the top and bottom of the image.

Relationships

Conforms To

`CustomStringConvertible`
`Decodable`
`Encodable`
`Equatable`
`Hashable`
`Sendable`

`SendableMetatype`

See Also

Image locations and regions

`struct NormalizedPoint`

A point in a 2D coordinate system.

`typealias NormalizedRegion`

A polygon composed of normalized points.

`struct NormalizedCircle`

The center point and radius of a 2D circle.

`protocol BoundingBoxProviding`

A protocol for objects that have a bounding box.

`protocol BoundingRegionProviding`

A protocol for objects that have a defined boundary in an image.

`protocol QuadrilateralProviding`

A protocol for objects that have a bounding quadrilateral.

`enum CoordinateOrigin`

The origin of a coordinate system relative to an image.