Function

simd_slerp(_:_:_:)

Returns a spherical linearly interpolated value along the shortest arc between two quaternions.

```
func simd_slerp(
    _ q0: simd_quatf,
    _ q1: simd_quatf,
    _ t: Float
) -> simd_quatf
```

Mentioned in

Working with Quaternions

See Also

Applying geometric operations to quaternions

`func simd_act(simd_quatf, simd_float3) -> simd_float3`

Returns a vector rotated by a quaternion.

`func act(SIMD3<Float>) -> SIMD3<Float>`

Returns the specified vector rotated by the quaternion.

`func simd_angle(simd_quatf) -> Float`

Returns the angle by which a quaternion rotates.

```
func simd_axis(simd_quatf) -> simd_float3
```

Returns the normalized axis about which the action of the specified quaternion rotates.

```
func simd_bezier(simd_quatf, simd_quatf, simd_quatf, simd_quatf, Float)
-> simd_quatf
```

Returns the spherical cubic Bezier interpolation between quaternions.

```
func simd_conjugate(simd_quatf) -> simd_quatf
```

Returns the conjugate of a quaternion.

```
func simd_imag(simd_quatf) -> simd_float3
```

Returns the imaginary (vector) part of a quaternion.

```
func simd_negate(simd_quatf) -> simd_quatf
```

Returns the negation of a quaternion.

```
func simd_real(simd_quatf) -> Float
```

Returns the real (scalar) part of a quaternion.

```
func simd_slerp_longest(simd_quatf, simd_quatf, Float) -> simd_quatf
```

Returns a spherical linearly interpolated value along the longest arc between two quaternions.

```
func simd_spline(simd_quatf, simd_quatf, simd_quatf, simd_quatf, Float)
-> simd_quatf
```

Returns an interpolated value between two quaternions along a spherical cubic spline.

```
func simd_dot(simd_quatf, simd_quatf) -> Float
```

Returns the dot product of two quaternions.

```
func dot(simd_quatf, simd_quatf) -> Float
```

```
func simd_length(simd_quatf) -> Float
```

Returns the length of a quaternion.

```
func simd_normalize(simd_quatf) -> simd_quatf
```

Returns a quaternion pointing in the same direction as the supplied quaternion with a length of 1.