Where to store package:

```
FileNameJoin[{$UserBaseDirectory, "Applications"}]
```

```
C:\Users\OWNER\AppData\Roaming\Mathematica\Applications
```

```
Needs@"QuaternionsHM`";
```

```
? quat*
```

QuaternionsHM`

quat　quatRotateVector　quatToFromList　quatToFromθV
quatFromAlignedMatrix　quatToFromEulerZYX　quatToFromMatrix

quatToFromθV

```
qA = quatToFromθV[30. °, {0, 0, 1}]
```

```
quat[0.965926, 0, 0, 0.258819]
```

```
qB = quatToFromθV[{60. °, {0, 7, 0}}]
```

```
quat[0.866025, 0, 0.5, 0]
```

```
quatToFromθV[qB]
```

```
{1.0472, {0, 1, 0}}
```

```
quatToFromθV[θ, {x, y, z}]
```

$$\mathrm{quat}\left[\mathrm{Cos}\left[\frac{\theta}{2}\right], x\,\mathrm{Sin}\left[\frac{\theta}{2}\right], y\,\mathrm{Sin}\left[\frac{\theta}{2}\right], z\,\mathrm{Sin}\left[\frac{\theta}{2}\right]\right]$$

```
quatToFromθV[θ, {x, y, z}] // quatToFromθV
```

```
{θ, {x, y, z}}
```

quatToFromList

```
quatToFromList@qA
```

```
{0.965926, 0, 0, 0.258819}
```

```
quatToFromList@quatToFromList@qA
```

```
quat[0.965926, 0, 0, 0.258819]
```

Multiplication

```
qC = qA ** qB
```

```
quat[0.836516, -0.12941, 0.482963, 0.224144]
```

$$qC ** \frac{1}{qB}$$

```
quat[0.965926, 0, 0, 0.258819]
```

```
quat[p0, p1, p2, p3] ** quat[q0, q1, q2, q3]
```

```
quat[p0 q0 - p1 q1 - p2 q2 - p3 q3, p1 q0 + p0 q1 - p3 q2 + p2 q3,
  p2 q0 + p3 q1 + p0 q2 - p1 q3, p3 q0 - p2 q1 + p1 q2 + p0 q3]
```

Power

```
qPowBase = quatToFromθV[57. °, {1, 2, 3}]
```
quat[0.878817, 0.127526, 0.255052, 0.382578]

$qPowBase^2$

quat[0.544639, 0.224144, 0.448288, 0.672432]

```
qPowBase ** qPowBase
```
quat[0.544639, 0.224144, 0.448288, 0.672432]

$qCubeRoot = qPowBase^{1/3}$

quat[0.986286, 0.0441108, 0.0882217, 0.132332]

$qCubeRoot^3$

quat[0.878817, 0.127526, 0.255052, 0.382578]

Reciprocal and Conjugate

```
qC
```
quat[0.836516, -0.12941, 0.482963, 0.224144]

Normalized

qC^{-1}

quat[0.836516, 0.12941, -0.482963, -0.224144]

```
Conjugate[qC]
```
quat[0.836516, 0.12941, -0.482963, -0.224144]

$qC ** qC^{-1}$

quat[1, 0, 0, 0]

```
qC ** Conjugate[qC]
```
quat[1, 0, 0, 0]

Not normalized

```
qCLarge = 3 qC
```
quat[2.50955, -0.388229, 1.44889, 0.672432]

$qCLarge^{-1}$

quat[0.278839, 0.0431365, -0.160988, -0.0747146]

```
Conjugate[qCLarge]
```
quat[2.50955, 0.388229, -1.44889, -0.672432]

$qCLarge ** qCLarge^{-1}$

quat[1, 0, 0, 0]

```
qCLarge ** Conjugate[qCLarge]
```
quat[9, 0, 0, 0]

Norm and Normalize

```
Norm@qCLarge
```

```
3.
```

```
Normalize@qCLarge
```

```
quat[0.836516, -0.12941, 0.482963, 0.224144]
```

Exp and Log

```
Exp@qC
```

```
quat[1.97037, -0.283992, 1.05987, 0.491889]
```

```
Log@qC
```

```
quat[0, -0.136958, 0.511133, 0.237218]
```

```
qC // Log // Exp
```

```
quat[0.836516, -0.12941, 0.482963, 0.224144]
```

```
qC // Exp // Log
```

```
quat[0.836516, -0.12941, 0.482963, 0.224144]
```

quatToFromMatrix

```
mC = quatToFromMatrix@qC
```

```
{{0.433013, 0.25, -0.866025}, {-0.5, 0.866025, 0}, {0.75, 0.433013, 0.5}}
```

```
mC // MatrixForm
```

$$\begin{pmatrix} 0.433013 & 0.25 & -0.866025 \\ -0.5 & 0.866025 & 0 \\ 0.75 & 0.433013 & 0.5 \end{pmatrix}$$

```
quatToFromMatrix@mC
```

```
quat[0.836516, -0.12941, 0.482963, 0.224144]
```

```
quatToFromMatrix[quatToFromθV[θ, {0, 0, 1}]] // MatrixForm
```

$$\begin{pmatrix} \text{Cos}[\theta] & \text{Sin}[\theta] & 0 \\ -\text{Sin}[\theta] & \text{Cos}[\theta] & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
quatToFromMatrix[
  quatToFromθV[α, {0, 0, 1}] ** quatToFromθV[β, {0, 1, 0}] ** quatToFromθV[γ, {1, 0, 0}]
] // MatrixForm
```

$$\begin{pmatrix} \text{Cos}[\alpha]\,\text{Cos}[\beta] & \text{Cos}[\beta]\,\text{Sin}[\alpha] & -\text{Sin}[\beta] \\ -\text{Cos}[\gamma]\,\text{Sin}[\alpha]+\text{Cos}[\alpha]\,\text{Sin}[\beta]\,\text{Sin}[\gamma] & \text{Cos}[\alpha]\,\text{Cos}[\gamma]+\text{Sin}[\alpha]\,\text{Sin}[\beta]\,\text{Sin}[\gamma] & \text{Cos}[\beta]\,\text{Sin}[\gamma] \\ \text{Cos}[\alpha]\,\text{Cos}[\gamma]\,\text{Sin}[\beta]+\text{Sin}[\alpha]\,\text{Sin}[\gamma] & \text{Cos}[\gamma]\,\text{Sin}[\alpha]\,\text{Sin}[\beta]-\text{Cos}[\alpha]\,\text{Sin}[\gamma] & \text{Cos}[\beta]\,\text{Cos}[\gamma] \end{pmatrix}$$

quatRotateVector

```
quatRotateVector[qC, {5, 7, 8}]
```

```
{4.66506, 10.7763, -0.330127}
```

```
quatRotateVector[qC, #] & /@ IdentityMatrix[3]
```

```
{{0.433013, 0.25, -0.866025}, {-0.5, 0.866025, 0}, {0.75, 0.433013, 0.5}}
```

quatToFromEulerZYX

```
? quatToFromEulerZYX
```

Symbol

Converts a quaternion to Euler ZYX angles, or vice verse.

Numeric input only. Input and output angles in decimal degrees.

quatToFromEulerZYX[*quat*]. Returns a table with the possible angle sequences.

quatToFromEulerZYX[*angle z*, *angle y*, *angle x*]. Returns quaternion.

quatToFromEulerZYX[{*angle z*, *angle y*, *angle x*}]. Returns quaternion.

```
qD = quatToFromEulerZYX[30, 60, 70]
```

```
quat[0.75946, 0.3738, 0.524184, -0.0934082]
```

```
quatToFromEulerZYX@qD
```

Z	Y	X
30	60	70
-150	120	-110

```
quatToFromEulerZYX[qD] // First
```

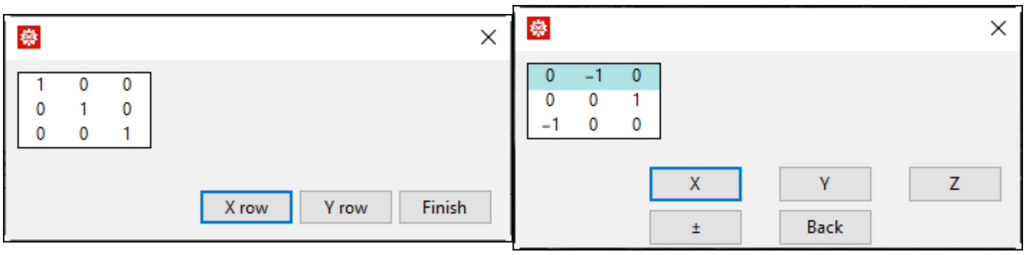
```
{{30, 60, 70}, {-150, 120, -110}}
```

```
quatToFromEulerZYX@quatToFromEulerZYX[30, -90, 70]
```

Z	Y	X
100	-90	0
0	-90	100
t	-90	100 - t

quatFromAlignedMatrix

```
qAM = quatFromAlignedMatrix[]
```




```
quat[0.5, 0.5, -0.5, -0.5]
```

```
quatToFromMatrix[qAM] // MatrixForm
```

$$\begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{pmatrix}$$