```
# NOTE:
 1
        Always import as:
 2
       from rotations import *
 3
 4
    def make_Rbf(angles):
 5
 6
        from math import cos, sin, pi
 7
         [psi, theta, phi] = angles
 8
 9
        psi = psi * (pi/180)
10
        theta = theta * (pi/180)
        phi = phi * (pi/180)
11
12
         R bf = [[cos(psi)*cos(theta), sin(psi)*cos(theta), -sin(theta)],
13
                 [cos(psi)*sin(theta)*sin(phi)-sin(psi)*cos(phi),
14
                 cos(psi)*cos(phi)+sin(theta)*sin(psi)*sin(phi),
 •
                 cos(theta)*sin(phi)],
15
                 [cos(psi)*sin(theta)*sin(phi)+sin(phi)*sin(psi),
                 sin(theta)*sin(psi)*cos(phi)-sin(phi)*cos(psi),
 •
                 cos(theta)*cos(phi)]]
 •
16
17
        return R bf
18
19
    def make_Rbw():
20
        from math import cos, sin, pi
21
         [alpha, beta] = angles
22
23
        # ***WARNING!***
24
        #If alpha or beta> 1 radian (60 deg), this breaks
25
        if alpha > 1 or beta > 1:
26
             alpha = alpha * (pi/180)
27
             beta = beta * (pi/180)
28
29
         R_bw = [[cos(alpha)*cos(beta), 0, 0],
                 [sin(beta), 0, 0],
30
                 [sin(alpha)*cos(beta), 0, 0]]
31
32
33
        return R_bw
34
35
    def b2f(angles, vector):
36
        from numpy import matmul, transpose
37
         R_fb = transpose(make_Rbf(angles))
         return matmul(R_fb, vector)
38
39
40
    def f2b(angles, vector):
/11
        from numny import matmul
```

```
+\bot
        ттош пишру тшрогс шасшит
        R_bf = make_Rbf(angles)
42
43
        return matmul(R_bf, vector)
44
45
    def b2w(angles, vector):
        from numpy import matmul
46
        R_wb = transpose(make_Rbw(angles))
47
48
        return matmul(R_wb, vector)
49
    def w2b(angles, vector):
50
        from numpy import matmul, transpose
51
        R_bw = make_Rbw(angles)
52
53
        return matmul(R_bw, vector)
54
```