Happy holiday! Remember to take care of yourself and your loved ones!

```julia
mutable struct MyDiscreteWalker
    x::Int64
end
```

```julia
mutable struct SomethingElse
    x::Int64
    y::Float64
    name::String
end
```

MyDiscreteWalker

```julia
MyDiscreteWalker
```

2 methods for type constructor:

- Main.workspace#3.MyDiscreteWalker(x::**Int64**) in Main.workspace#3 at **/Users/virajvaidya/Downloads/MITCOVID7.jl#==#abbfa05c-7c28-11ec-1462-a3cbed7dd664:2**
- Main.workspace#3.MyDiscreteWalker(x) in Main.workspace#3 at **/Users/virajvaidya/Downloads/MITCOVID7.jl#==#abbfa05c-7c28-11ec-1462-a3cbed7dd664:2**

```julia
methods(MyDiscreteWalker)
```

w = MyDiscreteWalker(3)

```julia
w = MyDiscreteWalker(3)
```

MyDiscreteWalker

```julia
typeof(w)
```

3 + 4im

```julia
Complex(3, 4)
```

3 + 4im

```julia
3 + 4im
```

Complex(re::**T**, im::**T**) *where T<:Real* in Base at **complex.jl:14**

```julia
@which Complex(3,4)
```

InexactError: Int64(3.1)

1. **Int64** @ *float.jl:812* [inlined]
2. **convert** @ *number.jl:7* [inlined]
3. **Main.workspace#3.MyDiscreteWalker**(::Float64) @ [*Other: 2*]
4. **top-level scope** @ [*Local: 1*] [inlined]

```
MyDiscreteWalker(3.1)
```

```
struct ExampleImmutable
    x:: Int
    y:: Int
end
```

z = ExampleImmutable

```
z= ExampleImmutable
```

3

```
Int64(3.0)
```

jump! (generic function with 1 method)

```
function jump!(w::MyDiscreteWalker)
    w.x += rand( (-1, +1) )
end
```

pos (generic function with 1 method)

```
pos(w::MyDiscreteWalker) = w.x
```

3

```
w.x
```

(:x)

```
propertynames(w)
```

(:name, :super, :parameters, :types, :instance, :layout, :size, :hash, :flags)

```
propertynames(z)
```

set_pos! (generic function with 1 method)

```
function set_pos!(w, x)    #setter function
    w.x = x
end
```

jump (generic function with 1 method)

```
jump(MyDiscreteWalker) = rand( (-1, +1) )
```

jumpplus! (generic function with 1 method)

```julia
function jumpplus!(w::MyDiscreteWalker)
    old_pos = pos(w)
    set_pos!(old_pos + jump(w))
end
```

walk! (generic function with 1 method)

```julia
function walk!(w::MyDiscreteWalker, N)
    for i in 1:N
        jump!(w)
    end

    return w
end
```

2 methods for generic function **walk!**:

- walk!(w::**Main.workspace#3.MyDiscreteWalker**, N) in Main.workspace#69 at **/Users/virajvaidya/Downloads/MITCOVID7.jl#==#fda70178-b570-45f0-8643-a967164123f0:1**
- walk!(w, N) in Main.workspace#71 at **/Users/virajvaidya/Downloads/MITCOVID7.jl#==#49b79999-8f46-48d3-b858-f05b6fdeb73f:1**

```julia
methods(walk!)
```

```julia
mutable struct MyContinuousWalker
    y::Float64
end
```

```julia
Enter cell code...
```

false

```julia
w isa MyContinuousWalker
```

2 methods for generic function **jump!**:

- jump!(w::**Main.workspace#3.MyDiscreteWalker**) in Main.workspace#48 at **/Users/virajvaidya/Downloads/MITCOVID7.jl#==#fda9db2d-a65a-4a4b-91aa-d7122ebee621:1**
- jump!(w) in Main.workspace#69 at **/Users/virajvaidya/Downloads/MITCOVID7.jl#==#e9deba45-147c-411d-8a56-3608b1f3c509:1**

```julia
methods(jump!)
```

jump (generic function with 2 methods)

```julia
jump(w::MyContinuousWalker) = randn()
```

jump! (generic function with 2 methods)

```julia
function jump!(w)  # takes argument w of *any* type
    old_pos = pos(w)
    set_pos!(w, old_pos + jump(w))
end
```

1 method for generic function **pos**:

- pos(w::**Main.workspace#3.MyDiscreteWalker**) in Main.workspace#35 at **/Users/virajvaidya/Downloads/MITCOVID7.jl#==#7bb41264-635b-4e89-94c6-07f8c44c6086:1**

```julia
methods(pos)
```

set_pos! (generic function with 2 methods)

```julia
function set_pos!(w::MyContinuousWalker, pos)
    w.y = pos
end
```

walk! (generic function with 2 methods)

```julia
function walk!(w, N)
    for i in 1:N
        jump!(w)
    end

    return w
end
```