

# julia memo

Masaya Kameyama

2021-07-29

```
versioninfo()
```

```
Julia Version 1.10.0
Commit 3120989f39b (2023-12-25 18:01 UTC)
Build Info:
  Official https://julialang.org/ release
Platform Info:
  OS: macOS (arm64-apple-darwin22.4.0)
  CPU: 8 × Apple M1
  WORD_SIZE: 64
  LIBM: libopenlibm
  LLVM: libLLVM-15.0.7 (ORCJIT, apple-m1)
  Threads: 1 on 4 virtual cores
Environment:
  JULIA_NUM_THREADS =
```

**julia**

**for**

for  $x$ .

```
x = rand(1:5, 10, 3)
```

```
10×3 Matrix{Int64}:
 4  2  4
 3  3  3
 1  5  1
 1  3  5
 2  4  3
```

2 4 3  
4 3 4  
5 1 3  
2 4 1  
2 3 4

for :

```
for e in x
  println(e)
end
```

4  
3  
1  
1  
2  
2  
4  
5  
2  
2  
2  
3  
5  
3  
4  
4  
3  
1  
4  
3  
4  
3  
1  
5  
3  
3  
4  
3  
1  
4

eachrow :

```

for row in eachrow(x)
    println(row)
end

```

```

[4, 2, 4]
[3, 3, 3]
[1, 5, 1]
[1, 3, 5]
[2, 4, 3]
[2, 4, 3]
[4, 3, 4]
[5, 1, 3]
[2, 4, 1]
[2, 3, 4]

```

```

for col in eachcol(x)
    println(col)
end

```

```

[4, 3, 1, 1, 2, 2, 4, 5, 2, 2]
[2, 3, 5, 3, 4, 4, 3, 1, 4, 3]
[4, 3, 1, 5, 3, 3, 4, 3, 1, 4]

```

*y*

```

y=zeros{Int,3} for i=1:10
for i=1:10
    for j=1:3
        y[i][j]=rand{1:5}
    end
end
y

```

```

10-element Vector{Vector{Int64}}:
 [3, 4, 4]
 [1, 4, 4]
 [3, 3, 2]
 [5, 5, 2]
 [5, 2, 2]

```

```
[2, 1, 5]
[3, 2, 1]
[5, 5, 3]
[5, 2, 2]
[3, 2, 4]
```

:

```
for row in y
    println(row)
end
```

```
[3, 4, 4]
[1, 4, 4]
[3, 3, 2]
[5, 5, 2]
[5, 2, 2]
[2, 1, 5]
[3, 2, 1]
[5, 5, 3]
[5, 2, 2]
[3, 2, 4]
```

filter :

```
?filter
```

Base.Meta.ParseError: ParseError:

# Error @ /Users/masaya/projects/notebooks2/posts/2021-07-12-julia\_memo.ipynb:1:1

```
?filter
```

```
not a unary operator
```

filter :

```
x = reshape([rand{Int} for i=1:10*3], (:, 3))
```

10×3 Matrix{Int64}:

```
1434155849139153490  8792201589749886045  -349363450898840745
5845810908379303736 -8265185926902104011  -5065380767449374288
```

-3447320615839289584	2389268345481233176	-6485329445295354633
-5365742947389604229	-2326303111811756091	8698018437924141657
3595047806792093185	1172421900134288996	-4750125639239034923
1363143167553767130	7000736503973028484	5847702750677365622
-9135160136868605419	-2968857547344022737	-3598011947351858499
-7733185694899413747	-9217370134120495903	-7561702952325334262
-7716510106645479528	5657573041425477116	-1159079173192823842
-3856910791373303962	2502860500295826394	7127390806145534861

```
filter(isodd, skipmissing(x))
```

15-element Vector{Int64}:

```
-5365742947389604229
 3595047806792093185
-9135160136868605419
-7733185694899413747
 8792201589749886045
-8265185926902104011
-2326303111811756091
-2968857547344022737
-9217370134120495903
 -349363450898840745
-6485329445295354633
 8698018437924141657
-4750125639239034923
-3598011947351858499
 7127390806145534861
```

```
2 :
```

```
filter(x->iseven(x[2]), x)
```

BoundsError: BoundsError: attempt to access Int64 at index [2]

```
filter :
```

```
x[x[:,2] .%2 .==0,:]
```

5×3 Matrix{Int64}:

-3447320615839289584	2389268345481233176	-6485329445295354633
3595047806792093185	1172421900134288996	-4750125639239034923
1363143167553767130	7000736503973028484	5847702750677365622
-7716510106645479528	5657573041425477116	-1159079173192823842
-3856910791373303962	2502860500295826394	7127390806145534861

```
x[iseven.(x[:,2]),:]
```

5×3 Matrix{Int64}:

-3447320615839289584	2389268345481233176	-6485329445295354633
3595047806792093185	1172421900134288996	-4750125639239034923
1363143167553767130	7000736503973028484	5847702750677365622
-7716510106645479528	5657573041425477116	-1159079173192823842
-3856910791373303962	2502860500295826394	7127390806145534861

```
x[findall(a -> iseven(x[a,2]), 1:size(x)[1]),:]
```

5×3 Matrix{Int64}:

-3447320615839289584	2389268345481233176	-6485329445295354633
3595047806792093185	1172421900134288996	-4750125639239034923
1363143167553767130	7000736503973028484	5847702750677365622
-7716510106645479528	5657573041425477116	-1159079173192823842
-3856910791373303962	2502860500295826394	7127390806145534861

```
x[findall(iseven,x[:,2]),:]
```

5×3 Matrix{Int64}:

-3447320615839289584	2389268345481233176	-6485329445295354633
3595047806792093185	1172421900134288996	-4750125639239034923
1363143167553767130	7000736503973028484	5847702750677365622
-7716510106645479528	5657573041425477116	-1159079173192823842
-3856910791373303962	2502860500295826394	7127390806145534861

**list**                      **join**

atcoder                      for                      .

```
l=rand(0:9,100);
```

```
@time println(join(l))
```

08256702720409845107586473771897909877939939500976038582751332769489236262025737464924764229  
0.021755 seconds (8.97 k allocations: 606.688 KiB, 97.98% compilation time: 13% of which w

```
s=""
for n in l
    s*=string(n)
end
@time println(s)
```

08256702720409845107586473771897909877939939500976038582751332769489236262025737464924764229  
0.000111 seconds (18 allocations: 536 bytes)

```
function j(l)
    println(join(l))
end
@time j(l)
```

08256702720409845107586473771897909877939939500976038582751332769489236262025737464924764229  
0.083234 seconds (742 allocations: 46.375 KiB, 95.49% gc time, 4.17% compilation time)

```
function jj(l)
    s=""
    for n in l
        s*=string(n)
    end
    println(s)
end
@time jj(l)
```

08256702720409845107586473771897909877939939500976038582751332769489236262025737464924764229  
0.016438 seconds (2.65 k allocations: 170.320 KiB, 97.48% compilation time)

1.

```
@time begin
    s=""
    for n in l
        s*=string(n)
    end
    println(s)
end
```

08256702720409845107586473771897909877939939500976038582751332769489236262025737464924764229  
0.000128 seconds (418 allocations: 19.180 KiB)

**n**

**for**

```
for i=1:3, j=1:5
    println(i, " ",j)
end
```

```
1 1
1 2
1 3
1 4
1 5
2 1
2 2
2 3
2 4
2 5
3 1
3 2
3 3
3 4
3 5
```

**n CartesianIndices**

```
n=3
for c in CartesianIndices(ntuple(d->0:2, n))
    # vector
    x=collect(c.I)
    println(x)
end
```

```
[0, 0, 0]
[1, 0, 0]
[2, 0, 0]
[0, 1, 0]
[1, 1, 0]
[2, 1, 0]
[0, 2, 0]
[1, 2, 0]
```



[2, 2, 0]  
[0, 0, 1]  
[1, 0, 1]  
[2, 0, 1]  
[0, 1, 1]  
[1, 1, 1]  
[2, 1, 1]  
[0, 2, 1]  
[1, 2, 1]  
[2, 2, 1]  
[0, 0, 2]  
[1, 0, 2]  
[2, 0, 2]  
[0, 1, 2]  
[1, 1, 2]  
[2, 1, 2]  
[0, 2, 2]  
[1, 2, 2]  
[2, 2, 2]