Julia Lightning Round
MIT IAP Tutorial
January 15, 2013

Doug Bates
Jeff Bezanson
Britni Crocker
Iain Dunning
Alan Edelman
Keno Fischer
Stefan Karpinski
Miles Lubin
Jameson Nash
Viral Shah
John Myles White

Quitting

- Two ways to quit the interactive session
  - (known as a REPL: Read-Eval-Print-Loop)
- <cntl> D
- quit() ← Need Parens

- Clear the current Command at the prompt:
- <cntl> C

Julia Documentation

- Well written!
  - http://docs.julialang.org/en/latest/
- google: julia documentation

- Much of Julia is written (elegantly!) in Julia – it won't take you long before you start looking at Julia to learn Julia
  - When you are ready: google "julia source" and click on "base" or "examples" and browse around

Indexing: 1 based
Square brackets

- A=rand(5,5)
- A[1,1]
- rand(5,5)[1,1]
- Remember
  - parens for functions,
  - square brackets for indexing

Comprehensions (elegant “array constructors”)

- [i for i=1:5]
- [trace(rand(n,n)) for n=1:5]
- x=rand(10); [x[i]+x[i+1] for i=1:9]
- {eye(n) for n=1:5}
- [i+j for i=1:5, j=1:5]

  - Vision: automatic parallelism

Parentheses also used for multiple outputs

- A=rand(5,6);

- svd(A)
- (u,s,v)=ans
  - Notice that s is a vector
- type<tab> #tab completion
- ndims(u), typeof(u)
- ndims(s), typeof(s)

Ternary Operator

- si(x) = (x>0) ? 1 : -1
- si(x) = (x>0) ? 1 : ((x<0) ? -1: 0) # Chained
  "sign" (Comment: "#")

Complex Numbers

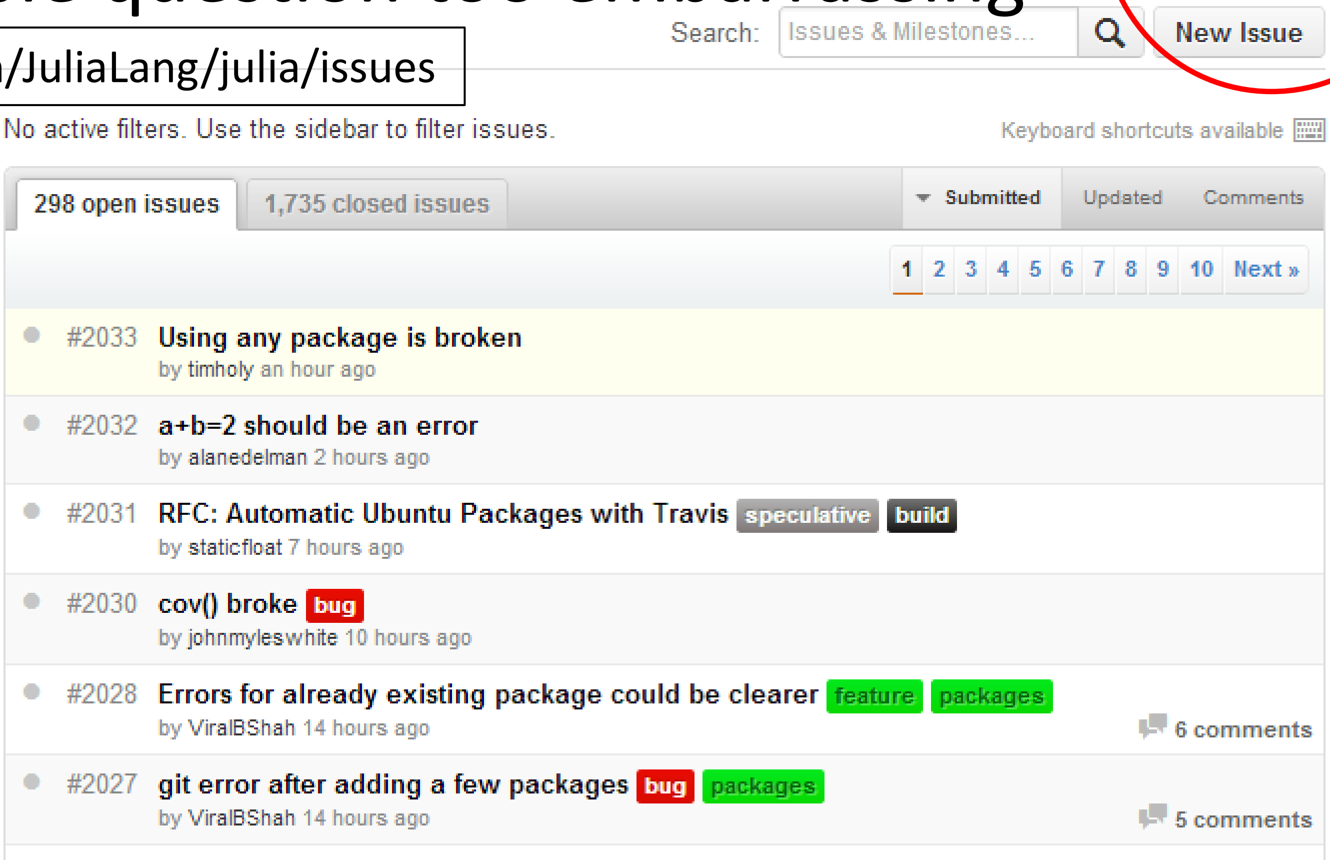
- im
- typeof(2im)
- typeof(2.0im)
- complex(3,4)
- complex(3,4.0) #multiple dispatch (more later)
- sqrt(-1)
- sqrt(complex(-1))

Issues Culture

- Old days: wait for a new release
- Julia: easy bugs fixed at the speed of light, rationale explained
- No newbie question too embarrassing

https://github.com/JuliaLang/julia/issues

Press me: Don't be shy



No active filters. Use the sidebar to filter issues.

#2033 Using any package is broken
by timholy an hour ago

#2032 a+b=2 should be an error
by alanedelman 2 hours ago

#2031 RFC: Automatic Ubuntu Packages with Travis speculative build
by staticfloat 7 hours ago

#2030 cov() broke bug
by johnmyleswhite 10 hours ago

#2028 Errors for already existing package could be clearer feature packages
by ViralBShah 14 hours ago — 6 comments

#2027 git error after adding a few packages bug packages
by ViralBShah 14 hours ago — 5 comments

Vectors

- A=rand(5,5)
- v=rand(5,1) ; w=rand(5)
- typeof(v)   # Array{Float64,1}
- typeof(1.0 : 5) #Range1{Float64}

  w=1.0: 5; A*w; #error (maybe shouldn’t be?)

  w=[1.0:5]; A*w; #ok
- ones(5) #vector!
- eye(5) #matrix (makes sense!)

running a file

- include("file.jl")
  - note that commands without semicolons won't print without "println" (print line)

Deployment

deploy.jl

```
n=int(ARGS[1])
println(randn(n,n))
```

- in shell:
- ./julia deploy.jl 5

- (lots more in “Getting Started” doc)

.. or even better dice

```
#! fullpath/julia
n=int(ARGS[1])
println(randn(n,n))
count=0
for i=1:n
  a=randi(6,3)
  count += (3==length(unique(a)))
end
println(count/n)
```

- in shell:
- chmod +x dice
- ./dice 1000

Outside Calls

- Shell
  - run(`cal`)
  - run(`cal` | `grep Sa`)
- C-function call
  - ccall(:clock, Int32, ())
  - bytestring(ccall(:ctime, Ptr{Uint8}, ()))

Punctuation Review

() Parentheses:
- Function Calls
    - Required! quit(),tic(),toc(),help()
- Output Arguments

[] Brackets:
- Indexing
- Array Constructors
    - Comprehensions

{} Braces:
- Any Arrays

Packages

- check out all the available packages off of docs.julialang.org
- Click, for example, on Calendar and get to the github project page

```
Pkg.add("Calendar")   #Only first time
using Calendar  #Calendar exists now
Calendar.now()
now()
```

Packages (cont)

- now()
- Calendar.<tab>
- quit() # get out of julia and back in
- now   # find line number in source
- #click on “src”
- typeof(now())
- n=now()
- n.tz
- n.millis
- z=convert(Array, @parallel [Calendar.now().millis for x=1:10]);z-mean(z)

Interesting Type

```
strang(n)=SymTridiagonal(2*ones(n),-ones(n-1))
lit=strang(500)
big=full(strang(500))
@time eigvals(lit)
@time eigvals(big)
big+big;
lit+lit;  #watch it break
import Base.+
xdump(lit)
+(a::SymTridiagonal,b::SymTridiagonal)=SymTridiagonal(a.dv+
b.dv,a.ev+b.ev)
lit+lit
```

Tasks (“produce”)
“pause” and “play” later

```
function stepbystep()
for n=1:3
      produce(n^2)
  end
end

p=Task(stepbystep);
consume(p)
consume(p)
consume(p)
consume(p)  #What should happen now?
```

Parallelism (more later)

- julia –p 5 #5 local processes
- julia –machinefile file #hosts in file
- addprocs_local(5) #inside a julia session
- @parallel #execute using every processor

Design Decisions

- Return type should not depend on value
- sqrt(-1) #error
- anothersqrt(x) = x<0 ? sqrt(complex(x)) : sqrt(x)
- [anothersqrt(x) for x=-2:3]