### Types

# Old-style definitions

immutable Point{T<:AbstractFloat}

index::Int

x::T

y::T

end

abstract A

type B <: A end

typealias P Point{Float16}

# New-style definitions

struct Plus

f::typeof(+)

end

mutable struct Mut

mutable::A # mutable should not be highlighted (not followed by struct)

primitive::B # primitive should not be highlighted (not followed by type)

end

primitive type Prim 8 end

abstract type Abstr end

### Modules

module M

using X

import Y

importall Z

export a, b, c

end # module

baremodule Bare

end

### New in 0.6

# where, infix isa, UnionAll

function F{T}(x::T) where T

for i in x

i isa UnionAll && return

end

end

### Miscellaneous

#=

Multi

Line

Comment

=#

function method0(x, y::Int; version::VersionNumber=v"0.1.2")

"""

Triple

Quoted

String

"""

@assert π > e

s = 1.2

変数 = "variable"

if s \* 100\_000 ≥ 5.2e+10 && true || x === nothing

s = 1. + .5im

elseif 1 ∈ [1, 2, 3]

println("s is $s and 変数 is $変数")

else

x = [1 2 3; 4 5 6]

@show x'

end

local var = rand(10)

global g = 44

var[1:5]

var[5:end-1]

var[end]

opt = "-la"

run(`ls $opt`)

try

ccall(:lib, (Ptr{Void},), Ref{C\_NULL})

catch

throw(ArgumentError("wat"))

finally

warn("god save the queen")

end

'\u2200' != 'T'

return 5s / 2

end