Some useless Julia code

The following Julia code contains all operators, some strings and some comments. Moreover a few names of variables consist of greek letters, superscripts or subscripts. However the code itself is rather useless unless you want to test how Julia code is displayed by the jlcode package.

```
#= A comment that consits of several lines.
Hence it has to be a little longer, of course.
Otherwise it would be silly as it could fit in one line. =#

# This line will be my reference line, which will exactly contain 80 characters.

# This line is a comment containing operators like &, - and %
# A comment with the German word "Übergrößengeschäft" (store for oversizes)
# This line contains some special unicode characters: α, γ, w², Δₓ
# A comment with some numbers: 424, 1.23, 0.2E-5, -9.9e+9



# defining a useless testfunction
function Style_3rd_Test(x, y)

    myver = v"2.00"
    mystr = "A string with \"Übergrößengeschäft\", α, π and the + operator."
    mychar = 'W'
    mychar⁽²⁾ = '€'
    mychar⁽³⁾ = '↯'
    z1vec = rand(Int8, 3)
    z₂vec = Array{Int8}(3)
    z₂vec[1:2] = [x % y, y \ x]
    t = x % 2 == 0 ? x : x + 1
    t = ~(t & x | y)
    α = 0.3
    β^α = 3.2e+5^α
    myβvar = 0.12E-2 * β^α
    z₂vec[3] = y^2 + 3.4x*y - (α + myβvar) * t/2
    z₂vec = (z₂vec + z1vec).^2
    if !(norm(z1vec) < norm(z₂vec) + e + pi + γ + ϕ)
        mystr = String( mystr, " signed ", mychar)
        println( mystr)
        return true;
    elseif norm(z₂vec - z1vec) > 2.69
        println( String( "Error in ", myver, "!"))
        return false;
    end

    # KNOWN ISSUES:
    # identifier name with number that follows
    # directly behind a special unicode character:
    myβ2ndvar = 2 * 0.12E-2 * β^α
    # identifier name, which contains a π, γ or ϕ:
    myϕvar = sqrt(2)
    # numbers in E-notation without using a + sign:
    evar = 3.99e400
    evar₂ = 3.99E400

end
```