



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικό και Καποδιστριακό
Πανεπιστήμιο Αθηνών

— ΙΔΡΥΘΕΝ ΤΟ 1837 —

Μια σύντομη εισαγωγή στη γλώσσα Julia

Γιάννης Τσαγκαρόπουλος

Αθήνα 2022

Julia

Χαρακτηριστικά της Julia

Η Julia είναι μια υψηλού επιπέδου, δυναμική γλώσσα προγραμματισμού. Τα χαρακτηριστικά του είναι κατάλληλα για αριθμητική ανάλυση και υπολογιστική επιστήμη. Είναι μία γλώσσα με πολύ υψηλές αποδόσεις, ενώ παραμένει φιλική προς τον χρήστη. Μερικά από τα χαρακτηριστικά που την κάνουν να ξεχωρίζει βρίσκονται παρακάτω:

- Είναι open source.
- Διαθέτει απίστευτα απλή σύνταξη, που θυμίζει πολύ αυτή της Matlab
- Είναι μια compiled γλώσσα, οπότε ο κώδικάς της μπορεί να εκτελεστεί απευθείας από την CPU του υπολογιστή και αυτό παρέχει υψηλή ταχύτητα. Πράγματι, χρησιμοποιήθηκε σε υπερυπολογιστές και έφτασε τα 1,54 petaflops.
- Είναι τόσο dynamic όσο η Ruby.
- Υποστηρίζει παραμετρικό πολυμορφισμό (generic programming). Χρησιμοποιώντας παραμετρικό πολυμορφισμό, μία συνάρτηση ή ένας τύπος δεδομένων μπορεί να γραφεί γενικά ώστε να χειρίζεται πανομοιότυπα τις τιμές ανεξάρτητα από τον τύπο τους.
- Οι μαθηματικοί συμβολισμοί της Julia μοιάζουν με αυτούς της Matlab. Επίσης, χρησιμοποιώντας το πακέτο LinearAlgebra, δίνει τη δυνατότητα για εύκολους υπολογισμούς Γραμμικής Άλγεβρας.
- Είναι τόσο φιλική προς τη στατιστική όσο και η R. Διαθέτει πακέτα που παρέχουν κατανομές, συναρτήσεις από τη στατιστική και εργαλεία για τη χάραξη παραστάσεων
- Υποστηρίζει multiple dispatch, δηλαδή έναν κανόνα εκτέλεσης με τον οποίο η συμπεριφορά μιας συνάρτησης καθορίζεται από τον συνδυασμό και τον αριθμό των τύπων ορίσματος. Δηλαδή, αποφασίζει ποια μέθοδος θα εκτελεστεί ανάλογα με τον συνδυασμό των τύπων εισόδου μιας συνάρτησης.
- Έχει αυτόματο garbage collection.
- Υποστηρίζει distributed computing, δηλαδή τη διαδικασία σύνδεσης πολλών υπολογιστών μέσω δικτύου, ώστε να μπορούν να λειτουργούν μαζί ως ένας ενιαίος εξαιρετικά ισχυρός υπολογιστής. Το δίκτυο αυτό προσφέρει τα εξής οφέλη:
 - Εύκολη επεκτασιμότητα: Απλώς προσθέστε περισσότερους υπολογιστές για να επεκτείνετε το σύστημα.
 - Δεδομένου ότι πολλά διαφορετικά μηχανήματα παρέχουν την ίδια υπηρεσία, αυτή η υπηρεσία μπορεί να συνεχίσει να λειτουργεί ακόμα και αν ένας (ή περισσότεροι) από τους υπολογιστές βγει εκτός λειτουργίας.
- Υποστηρίζει parallel computing, δηλαδή τη διαδικασία εκτέλεσης υπολογιστικών εργασιών σε πολλούς επεξεργαστές ταυτόχρονα για τη βελτίωση της υπολογιστικής ταχύτητας και απόδοσης. Χωρίζει τις εργασίες σε υπο-εργασίες και τις εκτελεί ταυτόχρονα μέσω διαφορετικών επεξεργαστών.
- Η Julia επιτρέπει τη χρήση συμβόλων unicode για την αντικατάσταση ονομάτων μεταβλητών και για κλήσεις συναρτήσεων.

- Με τη δυνατότητα ενδοσκόπησης κώδικα (code introspection) είναι δυνατή η εξέταση κλάσεων και συναρτήσεων αναφορικά με τη φύση και τη συμπεριφορά τους
- Οι περισσότερες βιβλιοθήκες είναι γραμμένες 100% στην Julia κάνοντας τη συμμετοχή στην ανάπτυξή τους πολύ πιο απλή.

Χρήσεις της Julia

Η Julia χρησιμοποιείται για το χειρισμό πολύπλοκων μαθηματικών, για υπολογισμούς μεγάλων δεδομένων, μηχανική μάθηση, τεχνητή νοημοσύνη, επιστημονικούς υπολογισμούς, εξόρυξη δεδομένων και πολλά άλλα.

Με εύκολη ενσωμάτωση σε ένα ευρύ φάσμα τομέων, η Julia βρίσκεται στο ραντάρ όλων, από εμπόρους και στελέχη επιχειρήσεων έως διαχειριστές πληροφορικής, προγραμματιστές και επιστήμονες δεδομένων.

Πράγματι,

- Η NASA χρησιμοποιεί τη Julia για να μοντελοποιήσει τη δυναμική αεροσκαφών
- Το Cern τη χρησιμοποιεί στον επιταχυντή LHC στο πείραμα LHCb.
- Η Federal Reserve Bank της Νέας Υόρκης τη χρησιμοποίησε για να μοντελοποιήσει την οικονομία των ΗΠΑ ελαχιστοποιώντας τον χρόνο εκτέλεσης από εβδομάδες σε ημέρες κάνοντας τα προγράμματα να εκτελούνται 10 φορές πιο γρήγορα από πριν τα αντίστοιχα προηγούμενα προγράμματα γραμμένα σε MatLab.
- Μια ερευνητική ομάδα ρομποτικής στο MIT τη χρησιμοποίησε για να προγραμματίσει ρομπότ που μπορούν να ανεβαίνουν σκάλες και να περπατούν σε ανώμαλο έδαφος.
- Η Astrazeneca και η Priois.ai τη χρησιμοποίησαν για να προβλέψουν την τοξικότητα των φαρμάκων με ένα Μπεϋζιανό neural network.
- Μια άλλη ερευνητική ομάδα του MIT βελτιστοποίησε με τη χρήση της Julia τα δρομολόγια λεωφορείων στην πόλη της Βοστώνης, καταργώντας 200 σχολικά λεωφορεία, εξοικονομώντας 18 εκατομμύρια \$ και αφήνοντας τους μαθητές να κοιμούνται περισσότερο.
- Χρησιμοποιήθηκε από τη Voxel8s για την κατασκευή τρισδιάστατων εκτυπωτών για κρίσιμο μαθηματικοποιημένο κώδικα και αλγόριθμους για την κατασκευή κεραιών επόμενης γενιάς, μπαταριών, drones, ρομποτικών χεριών, ηλεκτρονικών εξαρτημάτων και πολλά άλλα.

Άλλες καινοτόμες εφαρμογές της Julia μπορεί να βρει κάποιος [εδώ](#).

Κώδικας σε Julia

Παράδειγμα 1. Στα αριστερά βρίσκεται μία απλή υποθετική έκφραση. Ο κώδικας στα δεξιά είναι ισοδύναμος· εάν η έκφραση πριν από το ερωτηματικό είναι αληθής, επιστρέψε το αντικείμενο πριν από την άνω και κάτω τελεία. Εάν είναι ψευδής, επιστρέψε το αντικείμενο μετά από την άνω και κάτω τελεία.

```
if x<y
    return x
else
    return y
end
```

```
x<y?x:y
```

Παράδειγμα 2. Παράδειγμα ορισμού μιας πολυμορφικής συνάρτησης. Αρχικά ορίζουμε δύο μεθόδους:

- `add_`: `str` παίρνει σαν είσοδο 2 ορίσματα συμβολοσειράς και τα συνενώνει.
- `add_int`: παίρνει σαν είσοδο 2 ακέραια ορίσματα και τα προσθέτει.

Ορίζουμε την πολυμορφική συνάρτηση ως εξής: Όταν η συνάρτηση λάβει 2 ορίσματα συμβολοσειράς χρησιμοποιεί την `add_str` και όταν λάβει 2 ακέραια ορίσματα χρησιμοποιεί την `add_int`.

```
function add_str(str1, str2)
    str1*str2
end

function add_int(int1, int2)
    int1+int2
end

add(it1::Int64, it2::Int64) = add_int(it1, it2)
add(it1::String, it2::String) = add_str(it1, it2)
```

```
julia> add(4, 5)
9
julia> add("Julia ", "is awesome!")
"Julia is awesome!"
```

Κώδικας σε Julia που χρησιμοποιεί πακέτα

Παράδειγμα 3 (LinearAlgebra package). Η εντολή `using LinearAlgebra` θα χρησιμοποιήσει το πακέτο LinearAlgebra. Αν το πακέτο δεν είναι εγκατεστημένο στη συσκευή μας, θα εγκατασταθεί αυτόματα στο μηχάνημά μας και στη συνέχεια θα χρησιμοποιηθεί και θα έχουμε πρόσβαση στις εντολές του. Με την εντολή `lu` που παρέχεται από το πακέτο, θα λάβουμε την παραγοντοποίηση LU του A όπου οι κάτω και οι άνω τριγωνικοί πίνακες είναι εύκολα προσβάσιμοι όπως φαίνεται παρακάτω.

Τέλος, θέλουμε να βρούμε τον αντίστροφο ενός γινομένου πινάκων. Η Julia εκτελεί πρώτα τον γινόμενο πινάκων και υπολογίζει το αντίστροφο του αποτελέσματος με πολύ μεγάλη ταχύτητα.

```
julia> using LinearAlgebra

julia> A = [1 3 3 2; 2 7 3 4]
2×4 Matrix{Int64}:
 1  3  3  2
 2  7  3  4

julia> x = lu(A)
LU{Float64, Matrix{Float64}}
L factor:
2×2 Matrix{Float64}:
 1.0  0.0
 0.5  1.0
U factor:
2×4 Matrix{Float64}:
 2.0  7.0  3.0  4.0
 0.0 -0.5  1.5  0.0

julia> x.L
2×2 Matrix{Float64}:
 1.0  0.0
 0.5  1.0

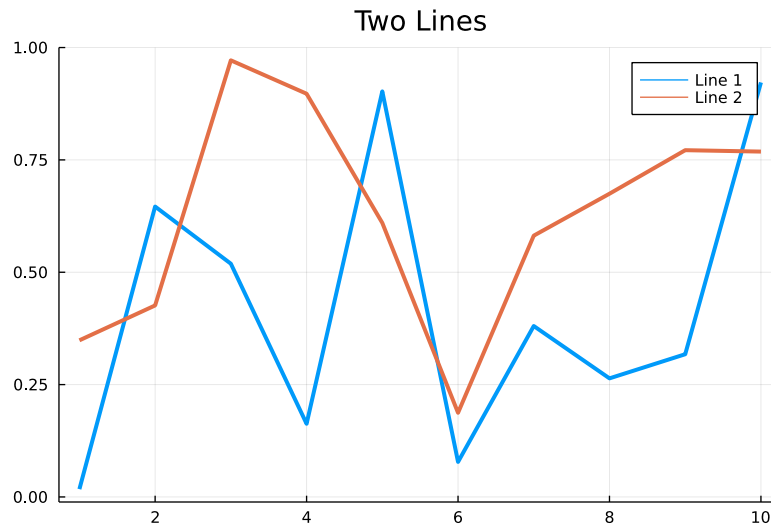
julia> x.U
2×4 Matrix{Float64}:
 2.0  7.0  3.0  4.0
 0.0 -0.5  1.5  0.0

julia> inv(A * [1 2 ; 3 2 ; 1 4; 4 2])
2×2 Matrix{Float64}:
-0.111765  0.0823529
 0.123529  -0.0647059
```

Παράδειγμα 4 (Plots package). Ορίζουμε ένα διάνυσμα x από το 1 έως το 10. Στη συνέχεια ορίζουμε έναν πίνακα 10×2 , έστω y με στοιχεία στο $(0, 1)$.

```
using Plots
x=1:10; y= rand(10,2)
plot(x,y, title="Two lines", label=["Line 1", "Line 2"], lw=3)
```

Το αποτέλεσμα είναι το εξής:



Επίλογος

Συνολικά, η Julia έχει βρεθεί στο προσκήνιο του προγραμματισμού για μία μόνο δεκαετία. Έχει πάνω από 40 εκατομμύρια λήψεις με ετήσια αύξηση 44%. Έχει πάνω από 8000 εγγεγραμμένα πακέτα μέχρι σήμερα και περισσότερα αναπτύσσονται καθημερινά.

Ένας προγραμματιστής περνάει πολύ χρόνο με μια γλώσσα προγραμματισμού και για να είναι αποδοτικός πρέπει να περνάει ευχάριστα τον χρόνο αυτό. Η Julia βγήκε 5η στην επισκόπηση του stack overflow για τις αγαπημένες γλώσσες προγραμματισμού στην κοινότητα προγραμματιστών και αυτό είναι ένα μεγάλο βήμα για την καθολική εξάπλωσή της.

The future seems **B r i g h t**

