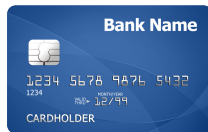


Ĉu vi fidas vian programlingvaĵon?

Martin Constantino–Bodin

19a de Junio

Komputiloj ĉirkaŭ ni



Programaĵoj ĉirkaŭ ni



2042
2020
Déclaration des revenus 2020

20
Ministère de l'Économie et des Finances

DÉCLARATION GÉNÉRALE
DES REVENUS PUBLICS

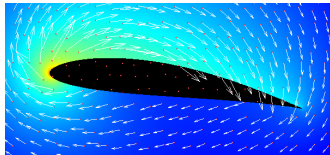
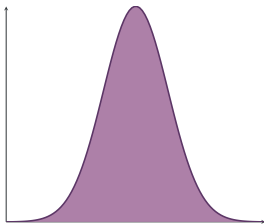
Écrivez votre déclaration au centre des finances publiques
de votre domicile au plus tard le 30 mai
ou déclarez sur impôt.gouv.fr

Vous devez déposer votre déclaration pour la première fois
si vous avez une copie de justificatif de votre domicile
(carte d'identité, passeport, titre de séjour, carte de séjour...)

Vous avez déjà déposé une déclaration, indiquez :
N° SIRET =
N° fiscal =
N° fiscal du conjoint =

ÉTAT CIVIL

DÉCLARANT 1 : Marié ☐ Divorcé ☐
Nom de naissance :
Prénoms :
Date de naissance :
Lieu de naissance :
DÉCLARANT 2 : Marié ☐ Divorcé ☐
Nom de naissance :
Prénoms :
Date de naissance :
Lieu de naissance :



Privateco

Efikeco

Sekureco

Fido

Malĉeesto de eraroj

Ĝustaj rezultoj

Privateco

Efikeco

Sekureco

Fido

Malĉeesto de eraroj

Ĝustaj rezultoj

La problemo

- Oni programas en pli kaj pli **diversaj fakoj**,
- Ofte estas **doktoriĝantoj** kiuj programas,
 - \neq profesiaj programistoj
 - Ĝenerale kun minimuma programa kursaro
- Uzo de **neregulaj** programlingvaĵoj.
 - Programlingvaĵoj kiuj provos **elkrei rezulton** mem se ne havas sencon
 - \neq fiaski





R ŝatas inventi rezultojn

```
1 v <- c(11, 12, 13, 14, 15)
2 v[1]
```

R ŝatas inventi rezultojn

```
1 v <- c(11, 12, 13, 14, 15)
2 v[1]                                # Rezultas en 11
3 indksoj <- c(3, 5, 1)
4 v[indksoj]
```

R ŝatas inventi rezultojn

```
1 v <- c(11, 12, 13, 14, 15)
2 v[1]                                # Rezultas en 11
3 indksoj <- c(3, 5, 1)
4 v[indksoj]                          # Rezultas en c(13, 15, 11)
```

R ŝatas inventi rezultojn

```
1 v <- c(11, 12, 13, 14, 15)
2 v[1]                                # Rezultas en 11
3 indksoj <- c(3, 5, 1)
4 v[indksoj]                          # Rezultas en c(13, 15, 11)
5 v[-2]                               # Rezultas en c(11, 13, 14, 15)
6 v[-indksoj]                         # Rezultas en c(12, 14)
```

R ŝatas inventi rezultojn

```
1 v <- c(11, 12, 13, 14, 15)
2 v[1] # Rezultas en 11
3 indeksoj <- c(3, 5, 1)
4 v[indeksoj] # Rezultas en c(13, 15, 11)
5 v[-2] # Rezultas en c(11, 13, 14, 15)
6 v[-indeksoj] # Rezultas en c(12, 14)
7 v[c(0, 2.9)]
```

R ŝatas inventi rezultojn

```
1 v <- c(11, 12, 13, 14, 15)
2 v[1]                                # Rezultas en 11
3 indksoj <- c(3, 5, 1)
4 v[indksoj]                          # Rezultas en c(13, 15, 11)
5 v[-2]                              # Rezultas en c(11, 13, 14, 15)
6 v[-indksoj]                        # Rezultas en c(12, 14)
7 v[c(0, 2.9)]                      # Rezultas en 12
8 v[c(FALSE, TRUE, FALSE)]
```


R ŝatas inventi rezultojn

```
1 v <- c(11, 12, 13, 14, 15)
2 v[1] # Rezultas en 11
3 indksoj <- c(3, 5, 1)
4 v[indksoj] # Rezultas en c(13, 15, 11)
5 v[-2] # Rezultas en c(11, 13, 14, 15)
6 v[-indksoj] # Rezultas en c(12, 14)
7 v[c(0, 2.9)] # Rezultas en 12
8 v[c(FALSE, TRUE, FALSE)] # Rezultas en c(12, 15)
9 v["a"] # Rezultas en NA
```

R ŝatas inventi rezultojn

```
1 v <- c(11, 12, 13, 14, 15)
2 v[1] # Rezultas en 11
3 indksoj <- c(3, 5, 1)
4 v[indksoj] # Rezultas en c(13, 15, 11)
5 v[-2] # Rezultas en c(11, 13, 14, 15)
6 v[-indksoj] # Rezultas en c(12, 14)
7 v[c(0, 2.9)] # Rezultas en 12
8 v[c(FALSE, TRUE, FALSE)] # Rezultas en c(12, 15)
9 v["a"] # Rezultas en NA
10 f <- function (a, b) v[a + b]
```

R ŝatas inventi rezultojn

```
1 v <- c(11, 12, 13, 14, 15)
2 v[1] # Rezultas en 11
3 indksoj <- c(3, 5, 1)
4 v[indksoj] # Rezultas en c(13, 15, 11)
5 v[-2] # Rezultas en c(11, 13, 14, 15)
6 v[-indksoj] # Rezultas en c(12, 14)
7 v[c(0, 2.9)] # Rezultas en 12
8 v[c(FALSE, TRUE, FALSE)] # Rezultas en c(12, 15)
9 v["a"] # Rezultas en NA
10 f <- function (a, b) v[a + b] # Oni ne scias kio okazos
```

R ŝatas inventi rezultojn

```
1 v <- c(11, 12, 13, 14, 15)
2 v[1] # Rezultas en 11
3 indksoj <- c(3, 5, 1)
4 v[indksoj] # Rezultas en c(13, 15, 11)
5 v[-2] # Rezultas en c(11, 13, 14, 15)
6 v[-indksoj] # Rezultas en c(12, 14)
7 v[c(0, 2.9)] # Rezultas en 12
8 v[c(FALSE, TRUE, FALSE)] # Rezultas en c(12, 15)
9 v["a"] # Rezultas en NA
10 f <- function (a, b) v[a + b] # Oni ne scias kio okazos
11 '[' <- function(a, b) 42
12 v[indksoj] # Rezultas en 42
```

Kion mi vidas rigardante tiajn semantikojn

coq → coq**s**
chou → chou**x**
chev**a**l → chev**au**x
aval → av**a**l**s**
vie**i**l → vieu**x**
œ**i**l → yeu**x**

rooster → roost**er**s
sky → sk**i**es
child → child**ren**
sheep → shee**p**
life → live**s**
fo**o**t → fee**t**

wom**a**n → wom**e**n
thes**i**s → thes**e**s
mic**e** → mou**s**e
ind**e**x → ind**ice**s
dat**u**m → dat**a**
fung**u**s → fung**i**

Kion mi vidas rigardante tiajn semantikojn

coq → coq**s**
chou → chou**x**
chev**a**l → chev**au**x
aval → av**a**l**s**
vie**i**l → vieu**x**
œ**i**l → yeu**x**

rooster → roost**er**s
sky → sk**i**es
child → child**ren**
sheep → shee**p**
life → live**s**
fo**o**t → fee**t**

wom**a**n → wom**e**n
thes**i**s → thes**e**s
mic**e** → mou**s**e
ind**e**x → ind**ice**s
dat**u**m → dat**a**
fung**u**s → fung**i**

koko → koko**j**

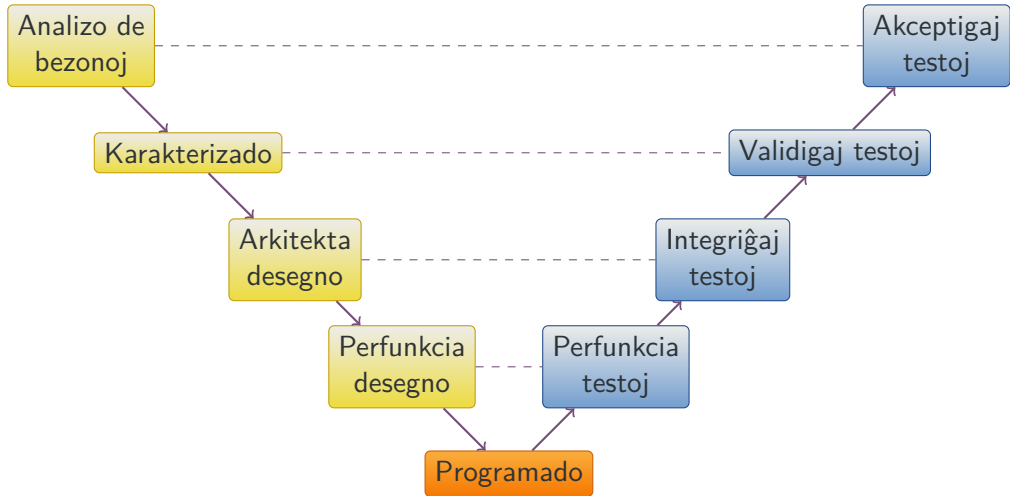
Solvoj

Kiel eviti malĝustajn rezultojn en malregula programlingvaĵo?

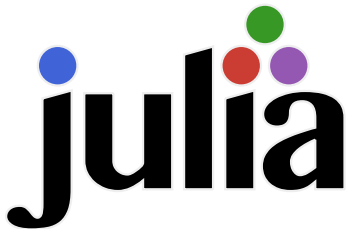
~_(_)_/_~

Programlingvaĵa malreguleco ne estas la nura fonto de eraroj.

Solvo 2: Testi, testadi, testadadi



Solvo 3: Uzi pli regulajn programlingvaĵojn



julialang.org



typescriptlang.org

Solvo 3: Krei pli regulajn programlingvaĵojn

- Kompreni kion bezonas / kiel pensas la programistoj
- \approx metodoj el lingvistiko



julialang.org

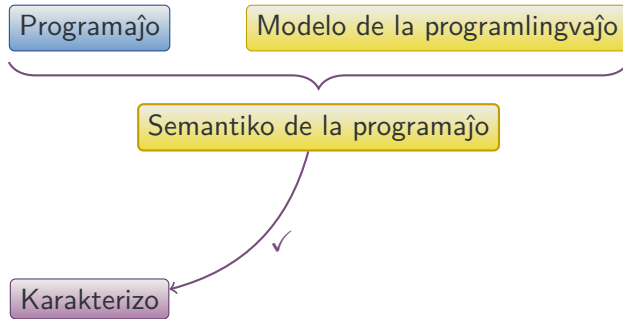


typescriptlang.org

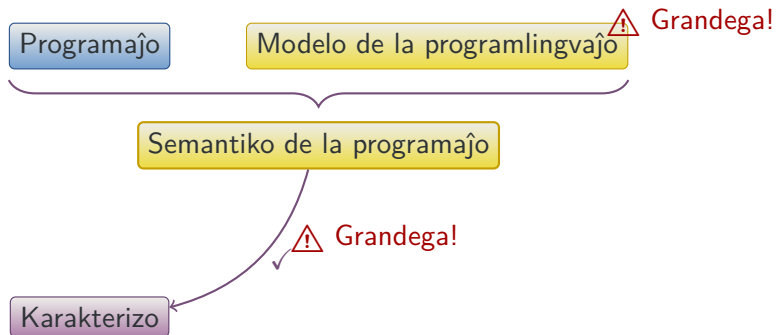
Solvo 4: Matematike pruvi la programajon ĝustan



Solvo 4: Matematike prui la programajon ĝustan



Solvo 4: Matematike prui la programajon ĝustan



Coq





```
1 Definition divides (a b : nat) := exists c, b = a * c.
2 Notation "a | b" := (dividas a b).
3
4 Definition primo p := forall n, n | p -> n = 1 \/ n = p.
5
6 Theorem primas_2 : primo 2.
7 Proof.
8   (* Pruvo *)
9   Qed.
```

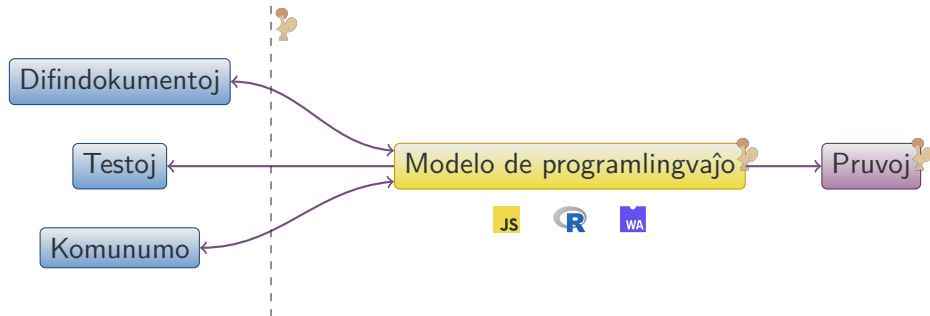

Modelo de programlingvaĵo

JS

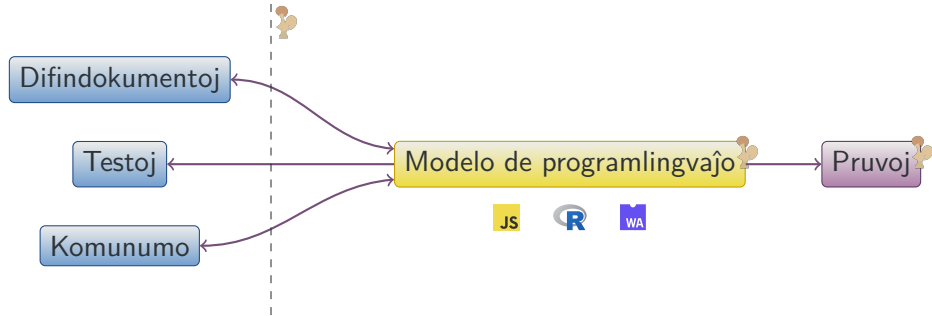
R

WA





Dankon pro via aŭskultado!



Ĉu vi nun fidas vian programlingvaĵon?

1 La problemo

2 Solvoj

- Ne zorgi
- Testi
- Uzi regulajn programlingvaĵojn
- Matematikaj pruvoj

3 Coq