

Kode-poesia

2019-2020

Antologia

Aurkibidea

Aurkibidea	1
Hitzaurrea	2
Parte hartu!	3
Poesiak	4
Spring love / Udaberriko maitasuna	5
Ilargiraino ta buelta	6
Tartaloren eta ardi lehenen ipuina	7
Ordainpeko lanaz	8
Trago bat.....	9
Bizitza.....	11
Noizarte.....	12
Zain.....	13
Beti Gose.....	14
Zure izena	15
Hautzaroa	17
Hauteskunde eguna	18
Boterearen konkista	19
Zu gabe ez naiz ezer, zu gabe zertarako bizi (Mikel Markez)	20
Kode Bertsoa	21
Jo ta ke!.....	22
Bibliografiak	23

Hitzaurrea

“Développez votre étrangeté légitime”
René Char

Poesiak, hitzen edo zentzua duten esaldien bitartez, sortzailearen eta irakurlearen arteko norabide bakarreko elkarrizketa bideratu eta ulermen maila ukiezinean lotzen ditu. Ezinbesteko baldintza da biek, elkar ulertu ahal izateko, hitzen, hautemanen eta sentitzeko moduen hizkuntza edo lengoia partekatu dezatela.

Ziberliteraturak edo literatura digitalak formatu digitalerako berariaz sortutako literatura-lanak izendatzen ditu. Aurrerapen teknologikoen eskutik 80.hamarkadan jazo ziren generoaren estreinako zantzuak sormena eta teknologiaren arteko gurutzebideak bultzatuta. Elena Asins Rodriguez (Madril, 1940 - Azpirotz, 2015) aitzindaria izan zen programazio-lengoia helburu artistikoarekin erabiltzen. Geroztik, begiradek AEBtara egin duten arren, Belén García Nieto (Sevilla, 1982), poesia kritikoa aldakarrikatzen duen sortzailea, poemak programazio-lengoian idazten aitzindaria da.

Euskarazko lehen kode-poesia bilduma da *Kode-poesia 2019-2020 antologia*. Eredu digitalaren eboluzioak eta teknologiaren etengabeko garapenak elikatzen du literatura-generoaren moldea. Zaila da jakiten etorkizuneko kode-poesiaren joera zein izango den; literatura etengabe sortzen ari da.

Leire Palacios

Parte hartu!

```
if ( zuk.gustuko( "poesia" ) && zuk.gustuko( "programazioa" ) ) {  
    while ( zuk.probatzekoGogoa ) {  
        kodePoesia = zuk.kodePoesiaIdatzi();  
        zuk.bidali( kodePoesia );  
    }  
}
```

Poesiak

Spring love / Udaberriko maitasuna

```
from random import randint as MargaritaZelaian

X = MargaritaZelaian(0, 100)
Y = 0
while Y < X:
    Y += 1
    if Y % 2 == 0:
        print('Maite nau')
    else:
        print('Ez dozte maite')
```

Ilargiraino ta buelta

```
for (i=0;i<ilargia;i++){nik.maite(zu);}
for (i=ilargia;i>0;i--){nik.maite(zu);}
```

Tartaloren eta ardi lehenen ipuina

```
#!/bin/bash
echo "Artzainak 100 ardi zituen eta Tartalo jatunak guztiak nahi zituen jan."
echo
echo "Artzainak zenbaki lehenak ziren ardi guztiak eskaini zizkion opari
bezala."
echo
echo "Tartalok tratua onartu zuen eta artzainak agindutakoa bete zuen:"
for i in {1..100}
do
    printf "%0${i}d" | grep -P "^0?|^?(00+?)\1+$" >/dev/null \
    || echo "- $i. ardia zuretzat."
done
echo
echo "Tartalok, 25 ardiak bere kobazulora eraman eta gordinik jan zituen."
```


Ordainpeko lanaz

```
#!/bin/bash
lan() {
    lan | lan &
}
lan
```

Trago bat

```
function tragoBat(asteKoEguna) {
  switch (asteKoEguna) {
    case "Astelehena":
      ura();
      etxera();
    case "Asteartea":
      muztioa();
      etxera();
    case "Asteazkena":
      zurittoa();
      etxera();
    case "Osteguna":
      while (!pintxopoteBukaera) {
        zurittoa()
        if (new Date().getHours() == 21 || gehitxoEdanda()) {
          pintxopoteBukaera = true;
        }
      }
      etxera();
    case "Ostirala":
      if (osteguneanPasata()) {
        muztioa();
        etxera();
      } else {
        while (!ostiralBukaera) {
          if (!lagunakEtxera() || !gehitxoEdanda()) {
            if (new Date().getHours() < 23) {
              zurittoa();
            } else {
              kubata();
            }
          } else {
            ostiralBukaera = true;
          }
        }
      }
    }
  }
}
```

```
case "Larunbata":
    if (ostiraleanPasata()) {
        muztioa();
        etxera();
    } else {
        while (!larunbatBukaera) {
            if (!lagunakEtxera() || !gehitxoEdanda()) {
                if (new Date().getHours() < 23) {
                    zurittoa();
                } else {
                    kubata();
                }
            } else {
                larunbatBukaera = true;
            }
        }
    }
case "Igandea":
    if (larunbateanPasata()) {
        resakola();
        etxera();
    } else {
        marianito();
        etxera();
    }
}
```

Bizitza

```
public class Bizitza {
    int PATUA=42; //Bizitzaren, unibertsoaren eta beste guztiaren erantzuna
    public Bizitza() {
        int taupadak=2000000000; //Bi mila milio taupada
        int aukerak =1111111111; //Hamaika aukera

        jaio();
        while(taupadak > 0) {
            int zoria = (int)(Math.random()*aukerak+1);
            if (zoria==PATUA) { //Zorionak!
                switch ((int)(Math.random()*3+1)) {
                    case 1:
                        zuhaitzaLandatu();
                        break;
                    case 2:
                        semealabaIzan();
                        break;
                    case 3:
                        liburuaIdatzi();
                        break;
                }
            }
            bizi();
        }
        hil();
    }
}
```


Zain

```
import random

X = random.randint(0, 100)
Y = X+8
while Y > X:
    X+=1
    Y+=1
print("maite nau")
```

Beti Gose

```
/**
 *
 * BetiGose programak "fork bomb" bat inplementatzen du. Programa hau "Jesus
 * Mari eta Jose! Beti jan da beti gose" Gketx Loiola s.v. beti-gose Poemaren
 * egokitzapen bat da. Kasu honetan, programak memoria jaten doa sistemaren
 * memoria kontsumitu arte.
 *
 * @author Igor Ruiz Agúndez
 * @version 1.0
 * @since 2019-10-01
 */

//
public class BetiGose {

    /**
     * Constructor honek memoria gose dela adierazten du eta errekurtsiboki
     * betiGose
     * objektuak eraikitzen ditu.
     */
    public BetiGose() {
        while (true) {
            System.out.println("MEMORIA gose");
            new BetiGose();
        }
    }

    public static void main(String[] args) throws java.io.IOException {
        System.out.println("Jesus Mari eta Jose! Beti jan da beti");
        new BetiGose();
    }
}
```

Zure izena

```
from Japonia import animea
from Radwimps import soinu_banda
import time
import random

#####
# Zure Izena #
#####
# Kimi no Na wa #
#####

def meteoritoaErrori(): pass

def bularrakUkitu(): pass
def gosaldu(): pass
def eskolaraJoan(): pass
def lagunekinEgon(): pass
def laneraJoan(): pass

def TokioraJoan(laguna): pass
def ItomoriraJoan(laguna): pass

def izenaAhaztu(): pass
def meteoritoaErroriBezpera(): pass
def meteoritoaSahiestu(laguna): pass

def egunaPasa(laguna, ametsa):
    if laguna == "Mitsuha" and ametsa:
        bularrakUkitu()
        gosaldu()
        eskolaraJoan()
        lagunekinEgon()
        if laguna == "Taki":
            laneraJoan()
    return "egunerokoan laburpena idatzi"
```



```
Taki = "Taki"
Mitsuha = "Mitsuha"

while !meteorittoaErori():
    if meteorittoaEroriBezpera():
        TokioraJoan(Mitsuha)
    else:
        ametsa = 0
        if (random.randint(0, 7) % 3) == 1:
            ametsa = 1
            Taki, Mitsuha = Mitsuha, Taki
            egunaPasa(Taki, ametsa)
            egunaPasa(Mitsuha, ametsa)
            Taki, Mitsuha = "Taki", "Mitsuha"
# meteorittoa erori da Itomorin
soinu_banda()
ItomoriraJoan(Taki)
izenaAhaztu()
Taki, Mitsuha = Mitsuha, Taki
print("Zure izena Taki da")
print("Zure izena Mitsuha da")
soinu_banda()
Taki, Mitsuha = "Taki", "Mitsuha"
meteorittoaSahiestu(Mitsuha)
izenaAhaztu()
TokioraJoan(Mitsuha)
soinu_banda()
print("Zure izena da ...?")

while 1:
    soinu_banda()
```

Haurtzaroa

```
public static void zikloa() {
    switch (zerDut) {
        case GOSE:
        case LOGALE:
        case KAKA:
            itxaron(PITTIN*1); // PITTIN <= 10 mikrosegundo
            hasiNegarrez();
            if (Gurasoa.ulertuDu()) {
                geldituNegarra();
            } else {
                zikloa();
            }
            break;
        default:
            eginIrribarre();
            break;
    }
}
```

Hauteskunde eguna

```
<hauteskundeak>  
  <hauteslekuak>  
    <hauteskunde-mahaiak>  
      <hautesleak>  
        <bokalak>  
          M-26an mendirik ez  
        </bokalak>  
      </hautesleak>  
    </hauteskunde-mahaiak>  
  </hauteslekuak>  
</hauteskundeak>
```

Boterearen konkista

```
from os import system, name
import time
boterea = 0
herria = 10

def boterea_herriarentzat(rows):
    global boterea, herria
    print('\nBoterea lortzen...\n')

    for i in range(rows):
        print(' '*(rows-i-1) + '*'*(2*i+1))
        if boterea == i:
            print((" "*(4))+"- BOTEREA -"+" "*(4))
        if herria == i:
            if herria:
                print((" "*(4))+"- HERRIA -"+" "*(4))
            else:
                print((" "*(6))+"BOTEREA"+" "*(6))
                print((" "*(4))+"HERRIARENA DA"+" "*(4))

def aldatetakDatoz():
    # Diskriminaziorik gabe, denok dugu askatasuna lortzeko eskubidea
    if name == 'nt': # windows
        _ = system('cls')
    else: # linux eta mac
        _ = system('clear')

def boterearen_konkista():
    global boterea, herria
    aldatetakDatoz()
    print("\nLoading... Boterearen konkista\n")
    time.sleep(3)
    for i in range(10):
        aldatetakDatoz()
        boterea+=1
        herria-=1
        boterea_herriarentzat(10)
        time.sleep(1)

boterearen_konkista()
```

Zu gabe ez naiz ezer, zu gabe zertarako bizi (Mikel Markez)

```
if (!zu) {  
    ni=0;  
}  
if (!zu) {  
    Thread.stop(); //deprecated method  
}
```

Kode Bertsoa

```
# Doinua ispiluarena

import a_aldu_gauza
import anteak_aurpegira_esateak
import a_ote_ote
import a_h as ieratik_agertzeak
for ma_askotan_eg in daiteke(
    aukeratu_bikoteak):
    lagun = dezake([dudarik, gabe])
    askatzen = gure(kateak)

#

barne = [sagr]
if izio_h and ie_(ta) == not ablek:
    eskatzen = (dittu_h or relako)
    egoera = bat[bizitze(k)]
    baina = ["harremana", "salba", "daikie", "barruko", "egia"]
    print("zek")
    _ = !(baidu == "mezi")
    ixiltasuna.append(izis([bihurtzen + uztek]))
```

Jo ta ke!

```
for(i=0; i<irabazi; i++){  
    jo && ke;  
}
```

Bibliografiak

Ion Lizarazu

Programatzaile ofizioz, idatzi zale tarteka, kode-poeta behin.

Aitzol Naberan

Ikasketaz Ingeniari Informatikoa EHUUn. Eguneroko lan jardunean gehienbat web garapena eta sistema kudeaketaz arduratzen naiz CodeSyntax-en. Python programazio lengoia erabiltzen dut nagusiki joan diren 15 urteetan.

Igor Ruiz Agúndez

Informazio Sistemen Doktorea. Ingeniari informatikoa. Ikertzaile, irakasle, garatzaile, proiektu kudeatzailea, familia burua. Kodean oreka, edertasuna, eta iratzarra aurki daitekeelako poesia eta informatika lotzeko bidean lanean.

Manex Garaio Mendizabal

Ez naiz idazlea, poeta ere ez. Programatzailea, batzuetan soilik. Kode-poesia egin zitekeela ikustean, baina, nire atentzioa lortu zuen. Arrakasta gehiago edo gutxiagorekin, egin dut kode-poesia saiakeraren bat. Ea zer deritsozuen.

Marko Txopitea

Ura deitu beharko genukeen plantetan, duela ziklo eskas batzuk, materia birziklatuarekin martxan jarritako beste biomakina horietako bat da Txopi. Bere arbasorik zaharrena, Unibertsoko espazio-denbora berdintsuan, bizi gabeko materiak autoeratutako makina autopoietikoa bat izan zen. Arbaso urrun eta ezdeus horretatik, biomakina xume eta iragankor honetara heltzeko egindako ibilbide apetatsua ahalbidetu duen kodea, hori ere bada kode-poesia.

Dabid Martinez

Dabid Martinez informatika teknikaria eta software librea eta jakintza askean aritua da (sekula ez aditua). Hacktismoan ibili da azken urteetan [hack.in#badakigu](https://hack.in/#badakigu) informatika elkarte askearekin, gizarte mugimenduak ingurugiro teknologikotik lagundu nahian. Blogaria da eta zenbait hedabideetan kolaboratzaile. Elkarbanatu zigun kode-poesia erabat zirikatzailea zen eta bash terminal batean exekutatzeko gero, ibili badabil. Norberaren kontu kode zati hori exekutatzeko Dabidetaz fidatzea. Edo ez ;)

Bittor Poza

Bittor Poza, Bitti lagunentzat. 24 urteko Bergaratarra naiz. Aplikazio mugikorren eta web garatzailea CodeSyntax-en. Teknologiarekin jolastearen maitale, eta bidean, gauza berriak ikasteko irrikan.

Aingeru Mayor

Alejandro Barredo Arrieta

Satan Txiki



Kode-poesia 2019-2020 Antologia

Poesia klasikoaren ezaugarriak programazio-lengoiarekin edo kode-informatikoekin nahasten dituen literatura digitala da kode-poesia. Gizakiek edo konputagailuek idatz dezaketen poesia generoa da. Kode-poema ordenagailuan exekutagarria izan daiteke edo ez, interaktiboa edo estatikoa izan daiteke eta digitala zein analogikoa izan daiteke. Programazio-lengoiarekin sortutako poemak poesia gisa irakurriak izan daitezke edota soinua, irudia eta mugimendua edertasunez exekutatu dezake irakurtzeko gaitasuna duen ordenagailuak.

Kode-poesia 2019-2020 antologiak PuntuEUS Fundazioak abiatutako euskarazko lehen kode-poesia laborategian bildutako lanek osatzen dute.

