

1. Sarrera

Lehen kapituluaren sarrera baten ohiko osagaiak bildu ditugu: liburuaren helburua, programazio-lengoiaren historia, lengoaia hau aukeratzeko arrazoiak, lengoiaren ezaugarriak eta aukerak, eta bukatzeko programa sortzeko lan-ingurunea, programa-adibideen kodea aurkitzeko bidea eta erabilitako bibliografia.

1.1. HELBURUA

Liburu honen helburua ez da Python lengoiaren aukera guztiak zehaztasun osoz deskribatuko dituen eskuliburu bat egitea, ikasteko material motibatzaile eta gertukoa sortzea baizik. Hori dela-eta, adibideak eta ariketak izango dira liburuaren muina; adibideak aztertzea, probatzea eta programa berriak sortzea baita ikasteko metodorik onena. Beraz, nagusiki tutorial bat da, eta bigarren maila batean erreferentziatzeko liburu bat. Hala eta guztiz ere, adibide kopuru mugatu batekin lengoaia ahaltu honek eskaintzen dituen aukera guztiak erabiltzea ezinezkoa denez, hainbat kasutan adibidean erabili ez diren beste aukera posible batzuk ere azaltzen dira gero deskribapen osagarri baten bidez, edo taula batean laburtuta.

Horrela kapituluak laburrak izango dira, adibideetatik abiatuta aukera ohikoenak azalduko dira hasierako ataletan, gero ariketa osatuagoak aurkeztu, eta bukaeran ariketa berriak proposatuko dira.

Liburuari etekina ateratzeko derrigorrezkoa da alde aurretik algoritmikako oinarrizko kontzeptuak ezagutzea, liburu ez da egokia hutsetik abiatuta programatzen ikasteko. Horretarako, eta liburu honen osagarri gisa, UEUK argitaratutako *Oinarrizko programazioa: Ariketa bilduma* liburu erabil daiteke. Atzigarri dago helbide honetan: <http://www.buruxkak.eus/liburuak_ikusi/1945/oinarrizko_programazioa__ariketa_bilduma.html>.

Liburu bi partetan banatzen da:

- **Oinarriak**, batez ere adibide sinpleen bidez azaltzen direnak.

Lehen partearen helburuak bi dira: lengoiaren nondik norakoak ondo ulertzea eta programa sinpleak edo ertainak programatu ahal izatea. Atal honetan erabiltzen den kodearen ezaugarri nagusia argia eta egituratua izatea da, eta ez hainbeste eraginkorra edo trinkoa izatea.

- **Aplikazio edo proiektu errealak**, ariketa luzeagoak eta erabilgarriagoak direnak, askotan modu librean Internetetik eskura daitezkeenak (halakoetan euskaraz komentatuta azalduko dira gure liburu honetan). Bigarren parte honen helburuak lau dira: (1) ezaugarri konplexu batzuk lantzea, (2) proiektu errealak ulertzea, (3) eginda dauden programak aldatu edo egokitzeko gaitasuna lantzea eta (4) sarean informazioa bilatzen trebatzea (programatzean aurkituko diren arazoentzako konponbideak lortzeko, edo behar ditugun programak bilatzeko). Azken hori oso inportantea da gaur egun, sarean baliabide asko baitago eta askotan lizentzia librearekin, eta beraz, halako kasuetan behintzat, ez dezagun gurpila berriz ere asmatu.

Geroago azalduko denez Python lengoaiaren hainbat bertsio daude, eta, duela gutxi arte Python2 hedatuena izan bada ere, etorkizunari begira ez dago zalantzarik: Python3 ikastea merezi du.

Liburuarekin batera adibide eta ariketen kodea eskaintzen da helbide honetan:

<<http://www.unibertsitatea.net/blogak/python>>

Webguneko programa horiek probatzeko eskaintzea eta jendeak berauen gainean hobekuntzak proposatzea izango da webgunearen helburua. Gainera, webgunearekin liburuaren aberasteko asmoa dugu, ariketa berriak proposatuko dira, soluzioak komentatu, erroreak zuzendu...

1.2. HISTORIA

Guido van Rossum herbeheretarrak sortu zuen Python lengoaiaren lehen inplementazioa 1989an Amsterdango Matematika eta Informatikako Zentroan (CWI, *Centrum Wiskunde & Informatica*). Lengoiari *Python* izena jarri zion hainbat pelikula komiko kritiko egin dituen Monty Python taldearen omenez.

2003. urteaz gero, Python etengabe egon da gehien erabilitako 10 lengoaien artean TIOBE programazio-komunitatearen sailkapenean, eta *urteko programazio-lengoaia* izendatu zuten 2007 eta 2010 urteetan. 2015eko irailean bosgarrena zen sailkapen horretan.

Bereziki egokia da string-en tratamendu sakona eta hiztegi-bilaketa asko egin behar dituzten programazio-problema bideratzeko. Enpresa handi askotan erabiltzen da Python, besteak beste honako hauetan: Google, Yahoo, CERN eta NASA.

2000. urtean sortu zen Python 2.0 bertsioa ezaugarri berriekin, Unicode erabiltzeko aukerarekin, esate baterako. Geroago 2008.ean sortu zen Python 3.0 (Python 3000 edo py3k ere deitua). Bertsio horrek funtsezko aldaketa batzuk ekarri zituen eta ez da bateragarria aurreko bertsioaren programekin. Python 3.0 bertsioaren ezaugarri berri asko Python 2 lengoaiara ere eraman izan ziren geroago eta Python 2.7 bertsioa sortu zuten horrela.

1.3. ZERGATIK PYTHON?

Python lengoaia simple, ulergarri eta ahaltsua da. Lengoiak bere barruan dakarren funtzionalitatea ez da oso zabala, baina oso erraza da oinarritzko muin hori zabaltzea. Modulu eta metodo ahaltzu asko erraz aurkitzen dira sarean, eta gehienetan lizentzia libre batekin. Horrela, esate baterako, erraz txerta daiteke interfaze programagarria behar duten aplikazioetan.

Lengoaia interpretatua da, erabiltzeko erraza hasieran. Aginduak banan-banan egikaritu daitezke aldeztatik konpiladore batek programa osoa konpilatu gabe.

Python software libre eta irekia da. Mozkin-asmorik gabeko fundazio batek (*Python Software Foundation*) kudeatzen du Python lengoaiaren ohiko inplementazioa (*CPython*).

Testu-tratamenduetan oso erabilia den Perl lengoaiaren filosofia eta Pythonena nahiko diferenteak dira programen idazteko filosofiaren ikuspuntutik. Perl lengoaiaren hainbat modu diferente egoten dira tratamendu bera lortzeko (*There is more than one way to do it*), baina Python lengoaiaren filosofia horren kontra doa eta saiatzen da adierazpide bakar eta intuitiboa definitzen (*There should be one—and preferably only one—obvious way to do it*). Helburua da programa simple, ulergarri eta ahaltzuak lortzea.

Bestalde, webguneak garatzeko ere egokia da Python. Aplikazio horiek liburu honen esparrutik kanpo geratzen badira ere, Django framework-a erabiltzeko, adibidez, derrigorrezkoa da Python jakitea.

Programazio-lengoaia	Zertarako sortua	Agintzailea	Objektuei zuzendua	Funtzionala	Prozedurazkoa	Generikoa	Erreflektiboa	Gertaeretara zuzendua	Beste paradigmatik
Ada	Aplikazioak, txertatua, denbora erreala, sistemak	X	X		X	X			konkurrentea, banatua
C	Aplikazioak, sistemak, helburu orokorra, behe-mailako eragiketak	X			X				
C#	Aplikazioak, RAD, gestioa, bezeroaren aldea, orokorra, zerbitzariaren aldea, Web	X	X	X	X	X	X	X	egituratua, konkurrentea
C++	Aplikazioak, sistemak	X	X	X	X	X			
COBOL	Aplikazioak, gestioa	X	X		X				
Cobra	Aplikazioak, gestioa, orokorra, Web	X	X	X		X	X		
Common Lisp	orokorra	X	X	X	X	X	X	X	
Fortran	Aplikazioak, zenbaki-konputazioa	X	X		X	X			
Haskell	Aplikazioak			X		X			ebaluazio alperra
Java	Aplikazioak, gestioa, bezeroaren aldea, orokorra, mobile development, zerbitzariaren aldea, Web	X	X	X	X	X	X		konkurrentea
JavaScript	bezeroaren aldea, zerbitzariaren aldea, Web	X	X	X			X		prototipotan oinaritua
Mathematica	kalkulu sinbolikoa	X	X	X	X				logikoa, banatua
MATLAB	Highly domain-specific, zenbaki-konputazioa	X	X		X				
Oxygene	Aplikazioak	X	X			X			
Pascal	Aplikazioak, hezkuntza	X			X				
Perl	Aplikazioak, scripting, text processing, Web	X	X	X	X	X	X		
PHP	Zerbitzariaren aldea, Web Aplikazioak, Web	X	X		X		X		
Prolog	Aplikazioak, adimen artifiziala								logikoa
Python	Aplikazioak, orokorra, Web, scripting, Adimen artifiziala, kalkulu zientifikoa	X	X	X			X		aspektuetara zuzendua
Ruby	Aplikazioak, scripting, Web	X	X	X			X		aspektuetara zuzendua

1.1. taula. Programazio-lengoaia ezagunen arteko konparazioa (Ingeles Wikipediatik erazua).

1.4. EZAUGARRI OROKORRAK

Programatzeko paradigma asko erabil daitezke Pythonekin: *Objektuei orientatutako programazioko* eta *programazio egituratuko* kontzeptu guzti-guztiak hartzen ditu bere gain, baita *programazio funtzionaleko* eta *aspektuetara zuzendutako programazioko* hainbat kontzeptu ere. *Programazio logikoa* eta *kontratu bidezko diseinua* ere erabil daitezke lengoaiaren luzapenen bitartez. Lengoiaren ezaugarri horiek kontuan hartuta, 1.1. taulan konparazio bat ikus daiteke beste lengoaia ezagun batzuekin, ingelesezko Wikipediatik erazi dugu hori.

Python bidez idatzitako kodea oso trinko eta irakurgarria izan ohi da, horri begira diseinatuta dago-eta. Trinkoa da, beste arrazoi batzuek gain duen ezaugarri bati esker: aldagaiak ez dira deklaratu/erazagutu behar; erabili ahala esleitzen baitzaie mota. Hori errore-iturri arriskutsua izan badaiteke ere, interpretatzaileak errore asko detektatzen ditu exekuzio-garaian. Bestalde, interpretazioa eraginkorra izan dadin programen formatua eta programazio-egitura batzuk zurrun samarrak dira. Hauek dira kontuan hartzeko xehetasun garrantzitsu batzuk:

- Tabulazioa. Ikusiko denez, agindu-blokeen mugak tabulazio bidez antzematen ditu interpretatzaileak (C lengoaiako giltzen orde, adibidez). *Tab* teklarekin edo zuriuneekin sortuko da, baina modulu berean era kontsistentean egin behar da, beti berdin, edo beti *Tab* bitartez edo beti zuriune bitartez.
- ':' karakterea: programazio-egitura batzuetan (*if*, *else*, *while* eta *for* esaterako) karaktere hori jarri behar da lehenengo lerroaren bukaeran egiturako espresioaren ondoren (benetako helburua bloke-hasiera markatzea da). Beste programazio-lengoaietan aritutakoei ':' karakterea idaztea ahaztu egiten zaie lengoaia honetan hastean.
- Parentesien erabilera. Baldintzak ez dira derrigorrez parentesi artean idatzi behar, baina funtzio-deietan beharrezkoak dira. Python2 lengoian ez ziren beharrezkoak funtzio deietan, baina Python3-n bai. Adibidez, inprimatzeko erabiltzen den *print* funtzioan argumentuak parentesi artean zehaztu behar dira beti, bai ala bai, baina ohitura zaharrak direla-eta horiek ahaztea ohiko errorea izaten da.

Programaren konplexutasuna minimo batetik pasatzen denean, eta programazio-egituratuaren eta objektuei orientatutako programazioaren metodologiari jarraitu ahal izateko funtzio asko definitu behar direnez, funtzio eta definizio lagungarri horien guztien definizioa ondo antolatu behar da. Horrelakoetan aparteko fitxategi batean bildu ohi dira objektu bakoitza definitzeko behar diren funtzio guztiak (klase bat, bere metodoekin), edo elkarren artean oso lotuta dauden hainbat funtzio (nahiz eta klase bat ez osatu). Fitxategi horiei *modulu* esaten zaie. Modulu asko erabil daitezke aplikazio batean, eta beren artean erabilera-menpekotasun ugari ere egon daitezke. Interneten hainbat klase eta modulu publiko aurki daitezke.

1.5. PYTHON2 VS PYTHON3

Python3 bertsioa etorkizuneko Python izango da, zenbait hobekuntza garrantzitsu baitauzka. Baina arazoa da gaur egun oso lagungarri diren hainbat eta hainbat programa-liburutegi oraindik Python2 bertsioan eskaintzen direla. Gauzak ondo badoaz, liburutegi interesgarrienak urte gutxi barru Python3 bertsioan ere eskuragarri egongo dira. Horregatik uste dugu Python ikasi nahi duen hasiberriarentzat egokiena Python3 ikastea dela.

Desberdintasunak ez dira asko, baina noski, arazoak eman ditzakete exekuzioan. Diferentzia garrantzitsuenetako batzuk hauek dira:

- Python3-k hobeto erabiltzen ditu Unicode karaktereak. Hainbat alfabeto desberdin (arabierarena, errusieraren edo japonierarena, esaterako) erabili ahal izateko aukera ematen duen kodetze-sistema da Unicode.
- *print* funtzioan parentesiak beti erabili behar dira Python3 erabiltzen denean, gainerako beste funtzioetan bezala. Hau da gehien-gehienetan nabaritutako duzun diferentzia. Adibidez:

```
# Python3-z ez da onartzen da Python2-ko agindu hau:
print 'Hello'
# Python2-z eta Python3-z onartzen da agindu baliokide hau:
print('Hello')
```

- Beste aldaketa txiki batzuk idazkeran: erroreak jasotzeko salbuespenetan, zenbaki osoen arteko zatiketean, zenbaki oso luzeetan, hiztegietako iterazioetan...

Diferentziei buruz informazio gehiago bilatzeko, eta Python2 eta Python3 bertsioen arteko bateragarritasuna lantzeko, hainbat webgune dago, adibidez helbide hauetan:

- *Python2 or Python3* - Wiki Python
<https://wiki.python.org/moin/Python2orPython3>
- *Easy, clean, reliable Python 2/3 compatibility*
http://python-future.org/compatible_idioms.html

Etorkizunari begira jarrita gure liburuan Python3 aukeratu dugu, noski. Zenbait kasutan aurreko bertsioarekiko desberdintasunak azaltzen dira liburuan, baina ez modu sistematikoan.

1.6. LAN-INGURUNEA

Python programazio-lengoaia interpretatua da, banan-banan egikaritu daitezke aginduak aldeztatik programa osoa konpilatu beharrik gabe, eta, beraz, programa zuzenean proba daiteke. Probatu ahal izateko, noski, Python interpretatzaile bat instalatu egin beharko da.

Python3 lengoaia instalatuta etortzen da Linux sistema eragilearen banaketa guztietan. Windows erabiltzen baduzu, berriz, bi aukera dituzu: instalatu edo *hodeitik* zuzenean erabili. Lehen aukerakoa lortzeko zenbait urrats bete beharko dira (<https://docs.python.org/3/using/windows.html>), eta kontuan hartu beharko dira Windowsen bertsioak. Hodeiarena, berriz, aukera neutroena da, nabigatzailea bat

baino ez baita behar. Aukera horretarako *codeskulptor.org* gunea gomendatzen da ikastaro askotan (baina Python2 erabiltzen du).

Linuxera itzuliz, hasierako programa txikiak martxan jartzeko edo aldezturik ditugun programen probak egiteko nahikoa da terminal batean *python3* interpretatzailea martxan jartzea eta aginduak edo funtzio-deiak egitea. Adibidez:

```
$ python3
Python 3.4.3 (default, Oct 14 2015, 20:33:09)
[GCC 4.8.4] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print ("Kaixo guztioi")
Kaixo guztioi
>>>
```

Baina programa ez oso trivialeak idatzi nahi badira, hobe da ez idaztea kode osoa zuzenean interpretatzailearekin. Horrelakoetan lan egiteko modu ohikoenak bi hauek izaten dira:

- Programa-fitxategiak osatzea testu-editore batean (*emacs* edo *gedit* esaterako) eta aldi berean paraleloan terminal batean programa horren exekuzioak egitea.
- *idle3* Python ingurunea erabiltzea. Kasu honetan ere beti izango dugu gutxienez bi leiho zabalik, bat programa idazteko eta bestea aginduak interpretatzeko.

Noski, kasu bietan funtsezkoa izango da jakitea editorean nola egin funtzio hauek:

- Lerro baten edo lerro-multzo baten tabulazio-maila aldatzea (Python lengoian tabulazio-maila da programaren barne-egitura definitzen duena).
- Lerro bat edo lerro-multzo bat iruzkin bihurtzea.
- Fitxategia gordetzea.
- Fitxategia egikaritzea.

Ariketa: Programa baten fitxategian exekuzio-adibide bat gehitzea editorea erabiliz.

Programa editorearekin apur bat praktikatzeko, osa ezazu 2. kapituluko *p2-2.py* fitxategia (*kap2* karpetan dago) bukaeran haren exekuzio bat iruzkin gisa gehituz. Hau da fitxategiak duen testua:

```
izena = input("Sartu zure izena: ")      # sarrera
print("Kaixo", izena)                   # irteera
```

Eta hau da bukaeran fitxategi horretan eduki nahi dugun testua:

```
izena = input("Sartu zure izena: ")      # sarrera
print("Kaixo", izena)                   # irteera

##      >>>
##      Sartu zure izena: Pythondegi
##      Kaixo Pythondegi
##      >>>
```

idle3 editorea erabiltzen baduzu, urrats hauek egin beharko dituzu:

- Terminal batean kokatu *kap2* direktorioan eta exekutatu *idle3* komandoa.
- *p2-2.py* fitxategia zabaltzeko:
(*File > Open...*)

```
izena = input("Sartu zure izena: ")      # sarrera
print("Kaixo", izena)                   # irteera
```

- Zabaldutako fitxategiko programa exekutatzeko (fitxategiaren leihoan):
(*Run > Run Module*) edo *F5* tekla bakarrik

```
Python 3.4.3 (default, Oct 14 2015, 20:33:09)
[GCC 4.8.4] on linux
Type "copyright", "credits" or "license()" for more information.
>>>
Sartu zure izena: Pythondegi
Kaixo Pythondegi
>>>
```

- Exekuzioko testua kopia eta eraman programaren fitxategira, bukaeran jarri.

```
izena = input("Sartu zure izena: ")      # sarrera
print("Kaixo", izena)                   # irteera

>>>
Sartu zure izena: Pythondegi
Kaixo Pythontdegi
>>>
```

- Exekuzioko lerroak aukeratu eta tabulazio-maila bat emanda eskuinera eramateko:
(*Format > Indent region*) edo *ALT+]* tekla-konbinazioa


```

izena = input("Sartu zure izena: ")      # sarrera
print("Kaixo", izena)                   # irteera

>>>
Sartu zure izena: Pythondegi
Kaixo Pythondegi
>>>

```

- Lerro-multzo hori aukeratu eta iruzkin bihurtzeko:
(*Format > Comment out region*) edo *ALT+3* tekla-konbinazioa

```

izena = input("Sartu zure izena: ")      # sarrera
print("Kaixo", izena)                   # irteera

## >>>
## Sartu zure izena: Pythondegi
## Kaixo Pythondegi
## >>>

```

- Fitxategia gordetzeko:
(*File > Save*)

Programaren funtzionamendua egokia ez denean, askotan zaila da jakitea zergatik den eta garrantzi handiko tresna bat erabiltzea gomendatzen da halakoetan: *debugger* edo arazle izeneko tresna. Pythonekin arazle desberdinak erabil daitezke, baina PDB izenekoa da erabiliena. Agian hemengo ariketak osatzeko ez duzu beharko, baina benetako proiektu baterako derrigorrezko tresna da. Hona hemen dagokion eskuliburua: <<https://docs.python.org/3.2/library/pdb.html>>.

1.7. KODEA

Esan bezala, liburu honetan erakusten diren programa guztiak blog honetan eskuragarri daude:

<http://www.unibertsitatea.net/blogak/python>

Programa guztiak daude *kodea* izeneko karpeta batean eta kapituluaren arabera azpikarpetetan antolatu dira (*kap2, kap3...*).

Blog horretan azalduko dira liburuari buruz etorkizunean sortuko diren zehaztasun, zuzenketa, azalpen edo material osagarri berriak ere.

1.8. BIBLIOGRAFIA

Testuan zehar webgune askoko erreferentziak emango dira, baina hemen erreferentzia nagusiak bildu nahi izan ditugu.

Liburuak euskaraz

Arantza Díaz de Ilarraza, Kepa Sarasola. *Oinarrizko programazioa: ariketa bilduma*. Udako Euskal Unibertsitatea. 1999. <<http://ueu.es/download/liburua/Oinarrizkoprogramazioa.pdf>>.

Aitzol Astigarraga, Koldo Gojenola, Kepa Sarasola, Aitor Soroa. *TAPE Testu-analisirako PERL erremintak*. Udako Euskal Unibertsitatea. 2009. <<http://ueu.es/download/liburua/PERL.BERRIA.osoa.pdf>>.

Iñaki Alegría, Nestor Garay. *C programazio-lengoaia*. Elhuyar. 1995.

Liburuak (on-line) ingelesez

The Python Tutorial: <<https://docs.python.org/3/tutorial/>>.

Introduction to Python: <http://nbviewer.jupyter.org/github/ehmatthes/intro_programming/blob/master/notebooks/index.ipynb>.

Erreferentziak

Ohiko erreferentzia: <<https://docs.python.org/3/>>.

Kontsultak: <<http://stackoverflow.com>>, <<http://stackoverflow.com/questions/tagged/python>>.

Esan bezala, programazio-lana, gero eta gehiago, bilaketan oinarritzen da: batzuetan proiekturako baliagarri izango diren modulu/klaseak bilatzea, beste batzuetan aurretik norbaitek egindako (antzeko) algoritmoa bilatzea, eta sarritan guri gertatzen zaigun antzeko akatsa/arazoa notifikatu duten mezuak bilatzea, erantzunaz baliatzeko. Azken bilaketarako berebiziko garrantzia du *Stackoverflow* webguneak, programatzaileen sare soziala, laguntza biltzeko eta eskaintzeko diseinatua. Software libererako gero eta erabiliagoa da *github* plataforma (<https://github.com/>), softwarea lankidetzan garatzeko pentsatua.