

1. Izveden je niz naredbi:

```
d= lambda p: p*2
t= lambda p: 3*d(p)
x=5
x=d(x)
x=t(x)
print x
```

Što će se ispisati? Odgovor: **60**,  $x=t(d(5))=t(10)=3*d(10)=6*10=60$

2. def change(x):

```
    x[0]='b'; ...(još neke naredbe,nebitno)

    x="proba"
    change(x)
    print x
```

Kaj će se dogoditi? Odgovor: **Error**. String je nepromjenjiv tip podataka.

3. Ispravno definiranje rječnika... **D={'a':3, 'b':2, 'c':3}**

4. Imamo datoteku koja sadrži retke oblika: "login";"Ime";"Prezime";JMBAG;"email".

```
Nadopuniti skriptu da joj izlaz bude: JMBAG      Prezime Ime
While(<>){
    _____ }
```

- a) `s/(.*/\4\™\3 \2\n/g){print $_;}` – ovo je bugovito do bola...
- b) `if (@a= /;\"(.*)\";/){print "@a[3]\t ..."} – nema opcije „g“ kod matchanja`
- c) `if(@a= /\\"*(.*)\"/*;/g){print "$a[3]\t $a[2] $a[1]\n";}` - ovo je točno, iako je užasno čudan regex... puno toga nepotrebnog i zbunjujućeg
- d) `@a = $_.split(';');` ... – nije ovo python xD Al skoro me zeznulo xD
- e) `If (/;(.*)/g){print "\3\t\2 \1\n";}` – tako se ne koristi backreference, trebaju biti \$1 itd.

5. Neispravan ključ za rječnik u Pythonu: `D[[3,2,1]]=7` (prolazi sve osim promjenjivih tipova)

6. `S="spam"; S = S[1:3]+S[:]` Koliko je objekata u memoriji (nakon dovoljno vremena da se obavio garbage collection)? Odgovor: **1!** Zato što se izgubi referenca na string „spam“ pa ga pokupi garbage collector, a stvori se novi string od dijelova starog stringa i na njega se pokaže starom varijablom S.

7. `@a = (1,2,3,4)`

```
@b = (5,@a,6,7)
```

```
$c=@b; print $c . " : " . $b[1]
```

Isprinta se -> 7 : 1 , zato jer u perlu član liste ne može biti lista nego se u jednostavno u listu b ubace svi članovi lista a. Pa zato lista b ima 7 članova, a drugi (s indexom 1) je prvi član lista a, a to je 1.

8. def f(): pass

def xyzy(a,b): return f()

Što vraća funkcija xyzy? Odgovor: **objekt None**. To zato što funkcija koja nema return po defaultu vraća vrijednost None. Dakle xyzy() vraća f(), a f() vraća None.

9. b=""; l="jedan 19 08 dva tri 456 66.9"

for i in l.split(): b=b+i[1:-1]

Što se nalazi u b na kraju izvođenja? Odgovor: **edavr56**. Split bez argumenta razdvoji string po razmacima i onda se stringu b dodaju sve 'riječi' iz string l, ali bez prvog i zadnjeg znaka.

10. U datoteci modul.py pišu sljedeće naredbe:

X=1; Y=1; if x==2: y=2

Što će se ispisati nakon izvođenja sljedećih naredbi:

import modul

print modul.X, modul.Y,

modul.X = 2

import modul

print modul.X, modul.Y

Odgovor: **1 1 2 1**. Razlog tomu je što se naredba import izvodi samo jednom za jedan modul. Ako bi željeli ponovno učitati modul trebali bi koristiti metodu reload() (al onda bi se i vrijednosti ponovno postavile na početne...)

11. [[1,2,3]+[4,[5]],6],[7,8,[9]]][1][2][0] + 3

Rezultat ovoga je: **12**. Drugi član liste je [7,8,[9]]. Treći član te liste je [9]. Prvi član te liste je 9. 9+3=12 ☺

12. def f(1,b,c):

return a+b \* c

print f('12','3',3)

Ispiše se: **12333**. Zato što množenje ima prednost pred zbrajanjem pa se prvo 'izmnoži' '3' i dobije se '333', a onda se tek to zbroji na '12'.

13. Ispisati sve datoteke iz tekućeg kazala i svih podkazala koje imaju bar 2 samoglasnika u imenu. Rješenje: ima ih svakakvih, nisam sve ponuđene zapisao. Ugl, ovo je točno:

**find . -regex "[aeiouAEIOU].[aeiouAEIOU].\*" -print**

Objašnjenje: sve piše u šalabahteru. Find traži u zadanom direktoriju i njegovim poddirektorijima, a to je ovdje točka što predstavlja tekući direktorij. Opcija regex služi baš onome što zadatak traži... da se nađu samo datoteke kojima ime odgovara danom regularnom izrazu. I to je to ☺

14. `class Account():`  
    `def __init__(self,id):`  
        `self.id=id; id=666`

`acc=Account(123)`  
`print acc.id`

Ispiše se: **123**. Jer `acc.id` poziva vrijednost `id` parametra točno tog primjerka klase, a to je `self.id` za taj primjerak te klase. Ono `id=666` apsolutno ništa ne radi jer kad završi funkcija `__init__` „`id`“ uopće više ne postoji kao takav.

15. `print "%s" % 'jabuka'[3:5]`  
Ispiše se: **uk**

16. `(round(8/3.) // 2)`  
Vrijednost izraza je: **1.0** Važno je naglasiti tip koji se vraća, a to je `float` (važni je jer bilo ponuđeno i samo 1, bez .0 iza... dakle `int` 1). Stvar je u tome da funkcija `round` vraća vrijednost tipa `float`. A operator `//` skida decimale, ali ostavlja tip varijable s kojim računa. Dakle `float // int` daje `float`...bez obzira što vrati cijelobrojnu vrijednost.

17. `my $str = '112133';`  
`$str =~ s/(.)\1/$1/g;`  
U varijabli `$str` nakon izvršavanja nalaziti će se string '**1213**'. Izvršit će se zamjena bilo koja 2 uzastopna ista znaka sa samo jednim tim istim znakom... i to nad varijablom `$str` (bindanje).

18. Ima hrpa nekih imena datoteka, pitanje je koja od njih će ispisati sljedeća linija koda u `bash-u`: `ls ^[CA]*[NB]?.*t*`  
Ovo ispisuje sve datoteke u tekućem direktoriju čiji naziv: NE počinje sa 'C' ili 'A' iiii koje u sebi sadrže podniz: **(N ili B)(jedan bilo koji znak).t** . Ono čega se morate sjetiti je da ovo nije regularni izraz nego sintaksa `bash-a`, a u njoj `*` predstavlja isto što i `.*` u `regexu`, a `?` predstavlja isto što i `.` u `regexu`.

19. `a="OkoSokolovo"; b=a.split('o'); a=b[0][0].join(b)`  
Prvo `b` postane ['Ok', 'S', 'k', 'l', 'v', ''] – primjetite prazni string na kraju, to je zato što se znak po kojem se splitalo nalazi na kraju niza.  
Zatim `a` postane 'O'.join(b)... što znači da se svi članovi lista `b` spoje, sa jednim znakom 'O' između. Na kraju je dakle rezultat: **OkOSOkOIloVo**

20. `i=[[1],[2,3],[[],4,5]]`; `a= "123" + i[2][1]`  
`i[2][1]` je BROJ 4. Naglašavam broj zato što je rješenje ovog zadatka: **ERROR**, budući da `python` ne zna zbrajati string sa brojem.

21. `a=[1,2,3,None(),[],{}]`  
`print len(a)`  
rješenje: **7** (samo izbrojite članove... sve su to članovi, bez obzira što su i kako se zovu)

22. Koja od ponuđenih petlji je petlja kroz sve retke standardnog ulaza (stdin). Nisam ih sve zapisao, ali nije ni bitno jer su neke očito krive. Ali 2 su jako slične i čovjek se lako zezne. Dakle jedna je: **while y=sys.stdin.readline(): ...** , a druga je **for x in sys.stdin.readlines(): ...** Točan odgovor je ova druga! Ovo prvo, iako naočigled točno, se jednostavno ne može napraviti u pythonu. Python ne podržava takvo pridruživanje varijabli unutar uvjeta...

23. `S= 'C:\test\test.c'`  
`print "%r" % S`

Što se ispiše? ... ispiše se **C:\test\test.c** Nemojte da vas zezne ovaj %r... bilo je ponuđeno "ne postoji takav format"...očito postoji :) A ovo se ispiše jer je to to, string se zadaje kao raw po defaultu. Da napišete `print S` on bi ispisao: `C: est est.c` jer bi tumačio \t kao tab.

24. Kako u bash-u evaluirati izraz  $(e-4)/d$  i pridružiti rezultat varijabli a.  
Odgovor: **`$( ( $e-4 ) / $d )`** Izrazi se evaluiraju u dvostrukim zagradama prethođenim dolar-znakom. Usput samo da kažem... unutar tih `$( )` izraza se mogu izostavljati \$ pred imenom varijabli. Dakle `$( ( e-4 ) / d )` je jednako dobar izraz.

25. `$string = "To dream od the person you would like to be is to waste the person you are.\n";`  
`@a = $string =~ /\b(\w{5,})\s/g;`

Ovo će vratiti u listu @a sve podnizove u varijabli \$string koji su dugački barem 5 znakova i započinju znakom \b (označava početak riječi) i završavaju nekim praznim znakom (znači razmak ili \n ili štogod...). To znači da će lista @a imati 5 članova: **dream, person, would, waste, person.**

26. Postoji datoteka pythonmodul.py u kojoj piše:

```
X=3
def func():
    print X,
    X=2; print X
```

Što će se ispisati kad mi izvršimo naredbe:

```
import pythonmodul
pythonmodul.func()
```

Ispisat će se **GREŠKA**. Razlog tomu je što pozivamo funkciju nekog modula izvan tog modula, a ta funkcija pokušava pristupiti globalnoj varijabli tog modula... a to jednostavno ne ide. Funkcija se može tako pisati i radila bi ako bi ju se pozivalo iz modula u kojem je napisana jer bi onda mogla koristiti podrazumijevanu globalnu varijablu X u prvom redu svog tijela. Budući da se poziva izvana... neće ić.

To bi bilo to 😊 Nadam se da će ovo biti nekom od pomoći... pozdrav svima i sretno na skriptnima.

Eugen Rožić (Simpsonim)