

- Sisäkkäiset silmukat
 - Helpottuu ajan myötä
- Käytännön koodaus
 - Aluksi hankalaa, koska syntaksin muisteleminen vie paljon aikaa
 - Itse koodaamisessa pääpaino ongelmanratkaisussa -> vaikeaa

PALAUTE

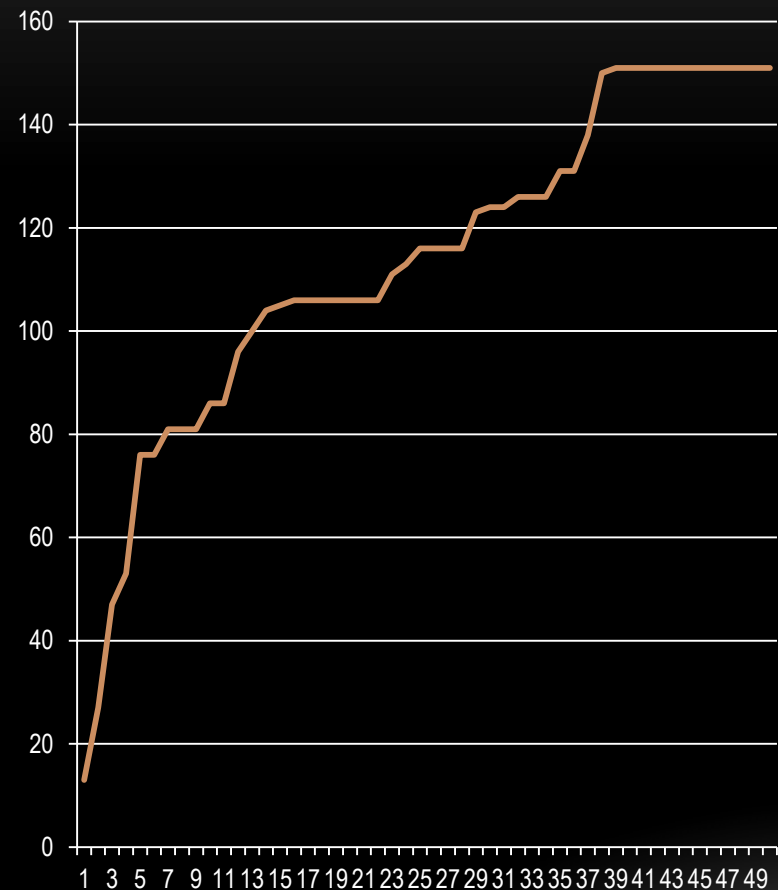
Mikä #-juttu livekoodauksessa?

#<kommentti>

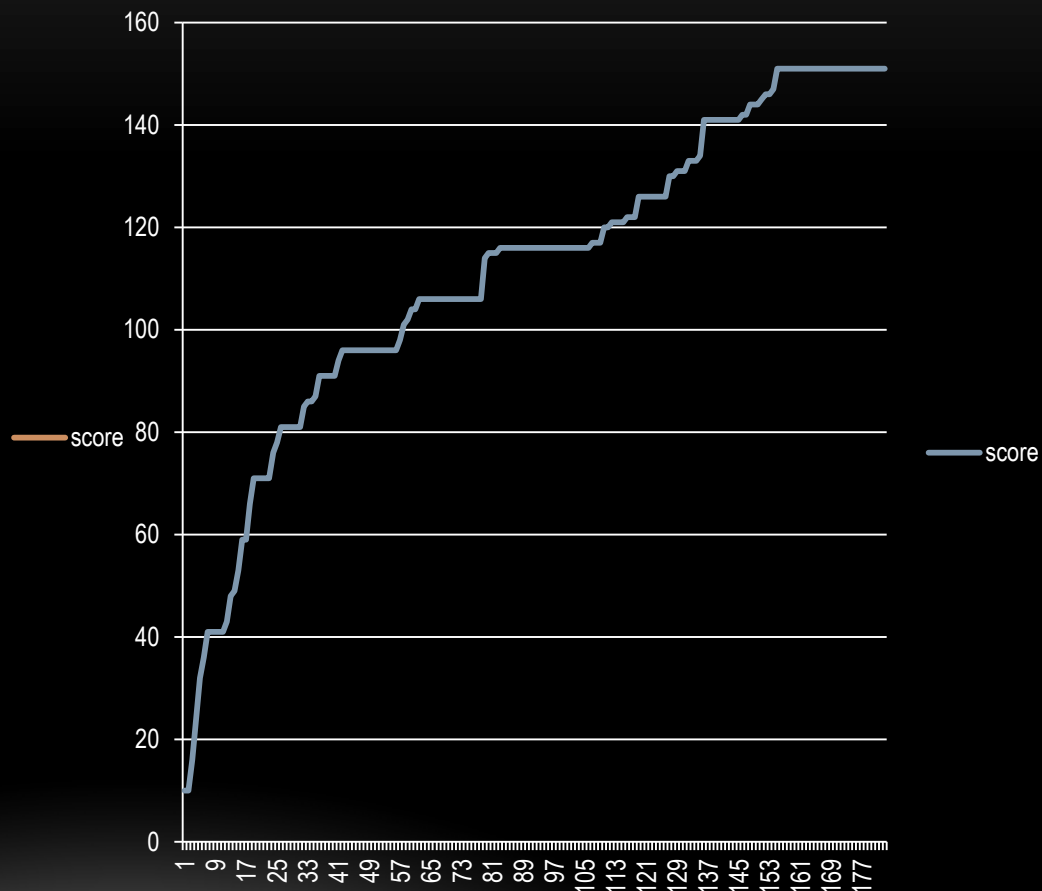
Ohjelmoijaa (sinua + muita)
varten; Python hyppää yli

PALAUTE

tutoriaaliscore_pääaine



tutoriaaliscore_sivuaine



PALAUTE

```
list = [1]
```

```
FOR x in list:
```

```
    IF job done:
```

```
        break
```

```
    ELSE:
```

```
        list.append(1)
```

```
        sleep(500)
```

```
WHILE NOT job_done:
```

```
    sleep(500)
```

FUNKTIOT JA PARAMETRIT

AGENDA

- Kertaus
 - Metodit
 - Parametrit
 - **Taukojumppa**
 - Funktiot
-

Muuttuja

Lause vs. Lauseke

KERTAUS

Muuttuja

Tietokoneen keskusmuistissa sijaitseva muistipaikka, johon tallennetaan arvo.

KERTAUS

Lause vs. Lauseke
Lausekkeella on arvo
Lause suoritetaan

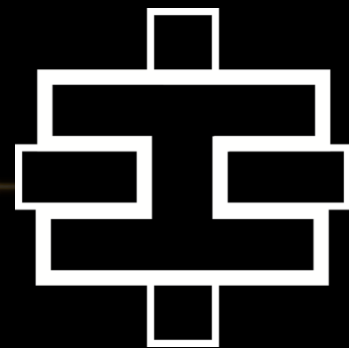
KERTAUS

Salaseuraan on soluttautunut
vakooja.

Salaisesta viestistä
selvisi, että vakoojan
toimipaikan ja iän tulo
on 3452291



JUONENKÄÄNNE

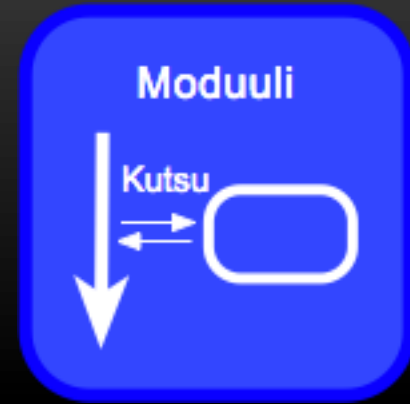


Moduuli on erillinen algoritmi, jota kutsutaan alkuperäisen algoritmin osaongelmien ratkaisemiseksi

MODULAARISUUS

Nimiä

- Metodi
- Rutiini
- Alirutiini
- Aliohjelma
- Moduuli



MODULAARISUUS

Moduuleilla haetaan

- Uudelleenkäytettävyyttä
- Yksinkertaisuutta
- Selkeyttä
- ABSTRAHOINTIA

MODULAARISUUS

Syntaksi

```
def <moduulinNimi>():  
    lohko
```

Tätä lohkoa kutsutaan myös **metodin rungoksi**

MODULAARISUUS

Moduulia kutsutaan

Kutsun syntaksi

<moduulinNimi>()

MODULAARISUUS

```
def tulostaYksi():  
    print "*"
```

```
def tulostaKaksi():  
    print "**"
```

```
def tulostaKolme():  
    print "***"
```

```
def tulostakolmio():  
    tulostaYksi()  
    tulostaKaksi()  
    tulostaKolme()
```

```
if halutaanKolmio:  
    tulostakolmio()  
else:  
    print "ei kolmiota"
```

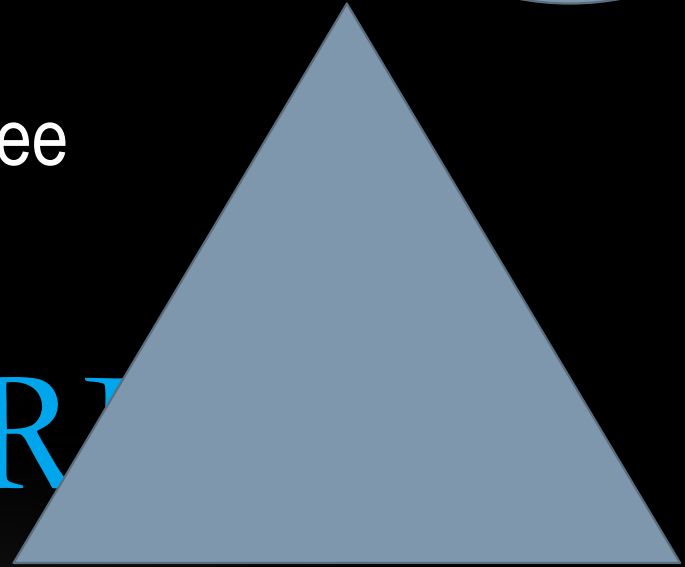
MODULAARISUUS:

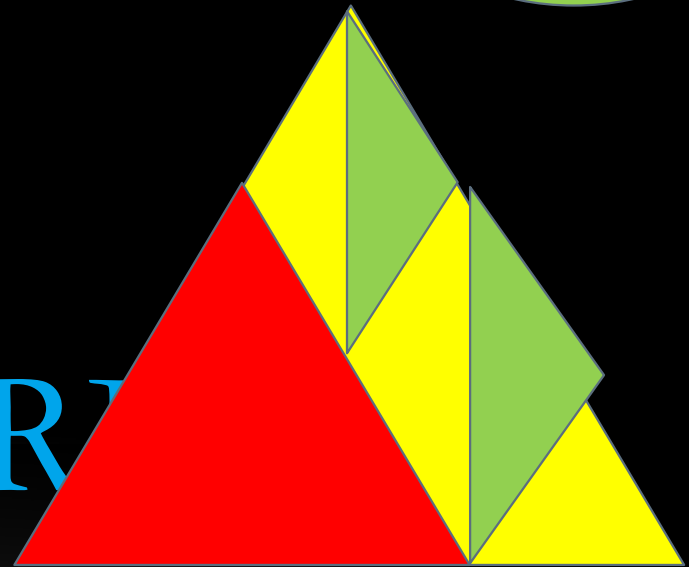
KÄYTTÖ



Yksi algoritmi löytää heidät,
se yksi heitä hallitsee,
se yksi heidät hajoittaa ja hallitsee

MODULAARI
ESIMERKKI



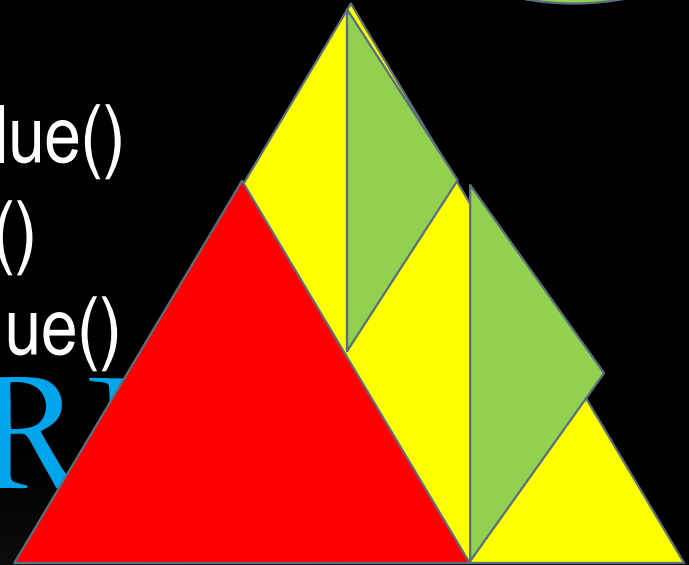


MODULAARI
ESIMERKKI



```
def ratkaiseSininenAlue():  
    while not ratkaistu:  
        ratkaisePunainenAlue()  
        ratkaiseVihreäAlue()  
        ratkaiseKeltainenAlue()
```

MODULAARI
ESIMERKKI



IF kirjoitat jo kirjoittamaasi koodia:
tee lohkosta moduli

IF modulista tulee pitkä:
hajota moduuli alimoduleihin

MODULAARISUUS:
SUUNNITTELU

Kirjoita kerran; käytä useasti

Koodi vähemmän 'koodin' näköistä

MODULAARISUUS:
EDUT



PARAMETRIT

```
def nostaPurjeet():
```

```
    nostaPurje1()
```

```
    nostaPurje2()
```

```
    nostaPurje3()
```

```
#Laivassa 15 raakapurjetta
```

PARAMETRIT

Parametri

Välitetään moduulille

Yleistää moduulia

Aina lauseke

PARAMETRIT

Moduulin syntaksi parametreilla

MP = Muodollinen parametri

```
def <metodinNimi>(<MP1>,  
<MP2> , ... , <MPn>):
```

lohko

PARAMETRIT

Muodollinen parametri?

Metodille välitetty arvo; eri
muuttuja kuin **todellinen**
parametri

PARAMETRIT

```
def nostaPurjeet():
```

```
    nostaPurje1()
```

```
    nostaPurje2()
```

```
    nostaPurje3()
```

```
#Laivassa 15 raakapurjetta
```

PARAMETRIT

```
def nostaPurjeet(purjeet):  
    for x in range(1, purjeet+1):  
        nostaPurje(x)
```

PARAMETRIT

```
def ???(a, b):  
    a = a + 1  
    b = b + 1
```

```
a = 3  
b = 5  
???(a, b)  
print a, b
```

PARAMETRIT

```
def kasvataArvoja(a, b):
```

```
    a = a + 1
```

```
    b = b + 1
```

```
a = 3
```

```
b = 5
```

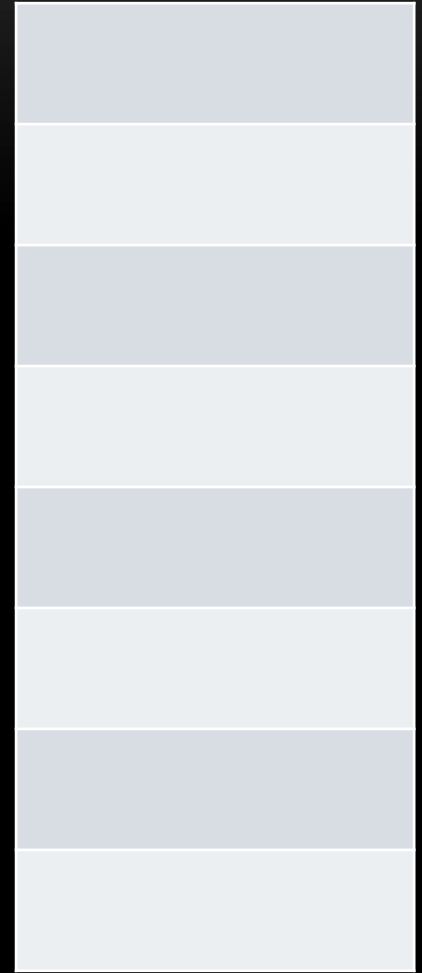
```
kasvataArvoja(a, b)
```

```
print a, b
```

Muodolliset parametrit
Arvoparametrit

PARAMETRIT

Keskusmuisti



PARAMETRIT

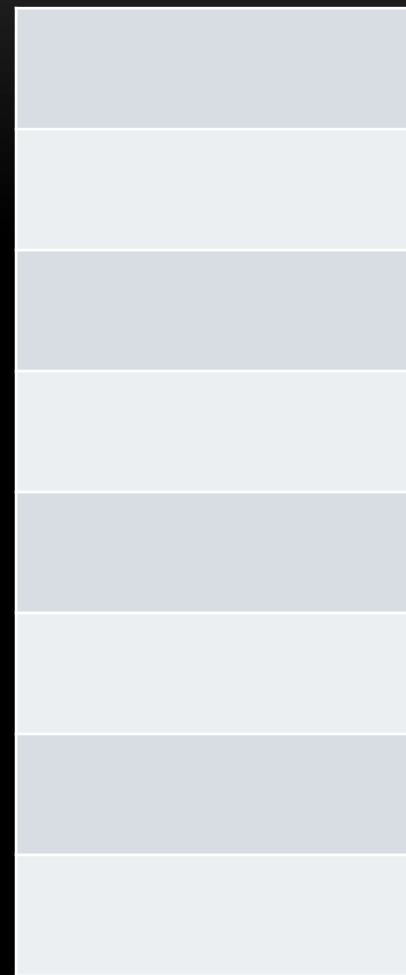


```
def kasvataArvoja(a, b):
```

```
    a = a+1
```

```
    b = b+1
```

Keskusmuisti



PARAMETRIT

```
def kasvataArvoja(a, b):  
    a = a+1  
    b = b+1
```

 a = 3

Keskusmuisti

a

3

PARAMETRIT

```
def kasvataArvoja(a, b):  
    a = a+1  
    b = b+1
```

Keskusmuisti

a = 3
b = 5



b

5

a

3

PARAMETRIT

```
def kasvataArvoja(a, b):
```

```
    a = a+1
```

```
    b = b+1
```

```
a = 3
```

```
b = 5
```



```
    kasvataArvoja(a+1, b)
```

```
    print a, b
```

Keskusmuisti



PARAMETRIT



```
def kasvataArvoja(a,b):
```

```
    a = a + 1
```

```
    b = b + 1
```

```
a = 3
```

```
b = 5
```

```
kasvataArvoja(a+1,b)
```

```
print a,b
```

Keskusmuisti

b	5
a	4
	kasvataArvoja
b	5
a	3

PARAMETRIT

def kasvataArvoja(a, b):

$a = a + 1$ $5-1$

$b = b + 1$

$a = 3$

$b = 5$

kasvataArvoja(a+1,b)

print a,b

Keskusmuisti

b	5
a	4
	kasvataArvoja
b	5
a	3

PARAMETRIT

```
def kasvataArvoja(a, b):
```

```
    a = a + 1
```

```
    b = b + 1
```

```
a = 3
```

```
b = 5
```

```
kasvataArvoja(a+1,b)
```

```
print a,b
```



Keskusmuisti

b	5
a	5
	kasvataArvoja
b	5
a	3

PARAMETRIT

```
def kasvataArvoja(a, b):
```

```
    a = a+1
```

```
    b = b+1
```

```
a = 3
```

```
b = 5
```



```
    kasvataArvoja(a+1, b)
```

```
    print a, b
```

Keskusmuisti

b	6
a	5
	kasvataArvoja
b	5
a	3

PARAMETRIT

```
def kasvataArvoja(a, b):
```

```
    a = a+1
```

```
    b = b+1
```

```
a = 3
```

```
b = 5
```

```
    kasvataArvoja(a+1, b)
```

```
print a, b
```



Keskusmuisti

b	5
a	3

PARAMETRIT

Proseduuri

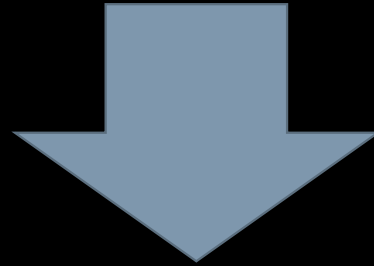
PROSEDUURI

Syntaksi

```
def <metodinNimi>(<parametrit>):  
    lohko
```

PROSEDUURI

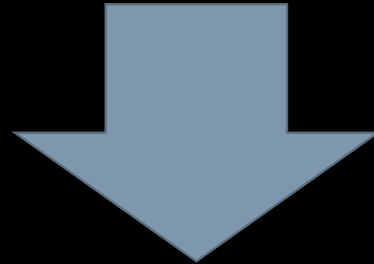
Proseduuri suorittaa asian



Proseduurit ovat itsetekoisia
lauseita

PROSEDUURI

Proseduuri suorittaa asian

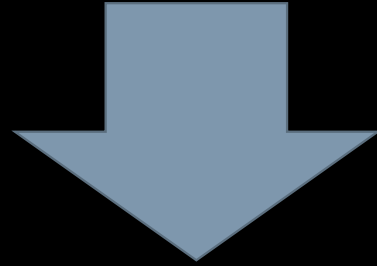


Proseduurit eivät palauta arvoa

PROSEDUURI

PROSEDUURI

Funktio kuvaa arvoa



Funktiot ovat itsetekoisia
lausekkeita

FUNKTIO

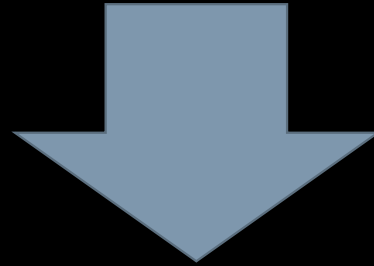
Syntaksi, funktio

```
def <metodinNimi>(<parametrit>):  
    lohko*
```

*Sisältää vähintään yhden return-lauseen

FUNKTIO

Funktio kuvaa arvoa



Funktiot palauttavat aina arvon

FUNKTIO

```
def laskeHypotenuusa(a,b):  
    c = sqrt(a**2 + b**2)  
    print c  
    return c
```

FUNKTIO?

```
def laskeHypotenuusa(a,b):  
    c = sqrt(a**2 + b**2)  
    print c
```

FUNKTIO?

```
def laskeHypotenuusa(a,b):  
    c = sqrt(a**2 + b**2)
```

FUNKTIO?

```
def summaaLuvut(luvut):  
    summa = 0  
    for x in luvut:  
        summa = summa+x  
    return summa
```

RETURN-LAUSE

```
def laskeKeskiarvo():  
    lista = lueLuvutKäyttäjältä()  
    summa = summaaLuvut(lista)  
    return summa / len(lista)
```

RETURN-LAUSE

```
def laskeKeskiarvo():
```

```
 lista = lueLuvut(Käyttäjältä())
```

```
 summa = summaaLuvut(lista)
```

```
 return summa / len(lista)
```



RETURN-LAUSE

"lause pienellä kirjaimella"

def lause_pienella_kirjaimella(lause):

return lause.capitalize()

FUNKTIO

2, 4, 2

```
def laske2AsteenYhtälö(a,b,c):  
    D = math.sqrt(b**2-4*a*c)  
    xMinus = (-b - D) / (2*a)  
    xPlus = (-b + D) / (2*a)  
    return (xMinus, xPlus)
```

FUNKTIO

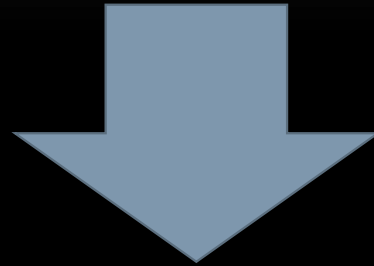
2010

```
def arvoNumero(15max):
```



FUNKTIO

Moduuleiden ei tarvitse kuitenkaan toimia
kaikilla arvoilla



Moduulin alkuehdot

FUNKTIO

#Heronin kaava. a,b,c kolmion sivut.

Palauttaa alan (float). a,b,c >=0

```
def laskeKolmionAla(a,b,c):
```

```
    s = (a+b+c)/2
```

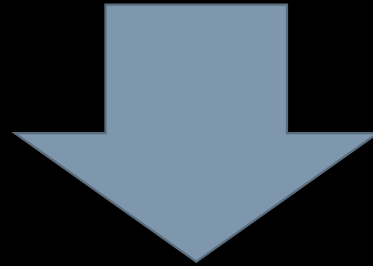
```
    return sqrt( s*(s-a)*(s-b)*(s-c) )/4
```

ALKUEHTO

Funktiot ovat lausekkeita

&

Parametrit ovat lausekkeita



Funktiokutsu parametriksi!

FUNKTIO

```
def lisaaYksi(x):
```

```
    return x+1
```

```
def summaaJaKerro(a,b,kerroin):
```

```
    return a+b*kerroin
```

```
summaaJaKerro(lisaaYksi(4), 3-2, 2*2)
```

FUNKTIO

 `def lisaaYksi(x):`

`return x+1`

 `def summaaJaKerro(a,b,kerroin):`

`return a+b*kerroin`

 `summaaJaKerro(lisaaYksi(4)  3,  2,  2)`

FUNKTIO

- `len(a)`
 - Palauttaa kokoelman `a` alkioden määrän (pituuden)
- `s.split(a)`
 - Katkaisee merkkijonon `s` jokaisen `a`:n kohdalta. Palauttaa listan
- `str(i)`
 - Muuttaa parametrin `i` merkkijonoksi
- `int(s)`
 - Muuttaa parametrin `s` kokonaisluvuksi
- `range([a,]b[,c])`
 - Generoi listan `a`:sta `b`:hen (ekslusiivinen), `c`:n askeleella
 - Hakasulkeilla merkityt vapaaehtoisia; `a`:n oletusarvo 0, `c`:n oletusarvo 1

VALMIITA METODEITA

Oletetaan merkkijono, jossa pilkulla erotettu alkiot:

"455,7,62,-7,2,8". Mikä on 4. pilkulla erotettu arvo?

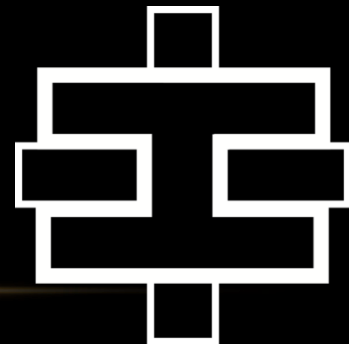
KÄYTÄNNÖN HARJOITE

Oletetaan merkkijono, jossa pilkulla erotettu alkiot:

"455,c,62,-7,www,8". Mikä on 4. pilkulla erotettu arvo?

KÄYTÄNNÖN HARJOITE

Merkkijono kurssimateriaalisivuilta. Arvot
on eroteltu jollakin toisella merkkijonolla.
Mikä on 100. arvo?



KOTITEHTÄVÄ

Nyt kaikki perusrakenteet opittu

Sou waht?

KAIKEN TÄMÄN MERKITYS