

Johdatus ohjelmointiin

Syksy 2006

Viikko 6

11.10. - 12.10.

1

matti.tedre@cs.joensuu.fi



2

matti.tedre@cs.joensuu.fi



Merkkijonojen käsittely

3

matti.tedre@cs.joensuu.fi



4

matti.tedre@cs.joensuu.fi



Tällä viikolla käsiteltävät asiat

- Merkkijonot
- Kerrataan tähän mennessä käsitellyt asiat

Merkkijonot

- Merkkijonoja ollaan käsitelty joka luennolla
- Merkkijonoja voidaan esimerkiksi
 - Määritellä kiintoarvoina tai muuttujina
 - Lukea käyttäjältä
 - Laittaa suoraan tulostuslauseisiin

```
tekijan_sposti = "matti.tedre@cs.joensuu.fi"  
kayttajan_nimi = raw_input ("Anna nimesi: ")  
print "(C) Seppo 2006"
```

Merkkijonot (2)

- Vielä ei olla tutustuttu merkkijonojen käsittelyyn.
- Merkkijonoja voidaan Pythonissa pätkiä, kopioida, leikata, liimata, jne. monipuolisesti.
 - Merkkijonoja käsitellään hieman samoin kuin *taulukoita*
 - Opitaan siis vähän jo taulukoiden käsittelyä vaikka varsinaisesti taulukoita ei tällä kurssilla käydäkään!

5

matti.tedre@cs.joensuu.fi



Merkkijonojen yhdistäminen

- Merkkijonoja voidaan yhdistää + -operaatiolla

```
kayttajatunnus = raw_input("Anna käyttäjätunnus: ")
spostipalvelin = "@cs.joensuu.fi"
```

```
spostiosoite = kayttajatunnus + spostipalvelin
```

```
print "Sähköpostisi on", spostiosoite
```

6

matti.tedre@cs.joensuu.fi



Merkkijonojen yhdistäminen

- Print-operaatiossa pilkulla toisistaan erotetut tulostettavat asiat saavat väliinsä yhden välilyönnin
- + -operaatiossa merkkijonot tulevat peräkkäin ilman välilyöntiä

```
>>>print "tietojenkäsittely", "tiede"
tietojenkäsittely tiede
>>>print "tietojenkäsittely" + "tiede"
tietojenkäsittelytiede
>>>
```

7

matti.tedre@cs.joensuu.fi



Merkkijono yksittäisinä merkkeinä

- Merkkijono on itse asiassa jono yksittäisiä merkkejä, ja jokaisella merkillä on oma paikkansa jonossa:

"matti"
01234

8

matti.tedre@cs.joensuu.fi



Merkkijono yksittäisinä merkkeinä

- Merkkijonon ensimmäisen merkin paikka ("indeksi") merkkijonossa on 0
 - Ohjelmoinnissa laskeminen alkaa usein 0:sta!

"matti"
01234

9

matti.tedre@cs.joensuu.fi



Merkkijono yksittäisinä merkkeinä

- Merkkijonon yksittäisiä merkkejä voidaan tulostaa ja tarkastella, muttei muuttaa.
- Indeksiä merkitään hakasuluilla []

```
>>> kayttaja = "Matti Tedre"
>>> print kayttaja[0]
M
>>> print kayttaja[6]
T
>>> kayttaja[6]="F"
Traceback (most recent call last):
  File "<pyshell#55>", line 1, in -toplevel-
TypeError: object does not support item assignment
```

10

matti.tedre@cs.joensuu.fi



Merkkijono yksittäisinä merkkeinä

- Tehdään ohjelma joka pyytää käyttäjän nimeä, tarkastelee käyttäjän antaman merkkijonon 1. merkkiä, ja tekee havainnon käyttäjän nimestä.

```
nimi = raw_input("Anna nimesi: ")
if nimi[0]=="M":
    print "Hyvä nimi."
else:
    print "Pöhkö nimi."
```

11

matti.tedre@cs.joensuu.fi



Pätkä merkkijonoa

- Merkkijonosta voidaan napata myös pätkiä
 - Määrätään aloitusmerkki
 - Määrätään lopetuskohta (mistä kirjainvälistä laitetaan poikki)

```
>>> nimi = "Pekka"
>>> print nimi[0:2]
Pe
>>> print nimi[1:4]
ekk
>>>
```

12

matti.tedre@cs.joensuu.fi



Pätkiä merkkijonosta

"matti"

01234

```
>>> nimi="matti"  
>>> print nimi[1:3]  
at  
>>> print nimi[3:5]  
ti  
>>> print nimi[0:3] + "-" + nimi[3:5]  
mat-ti
```

13

matti.tedre@cs.joensuu.fi



Merkkijonon loppupää

- Jos lopetuskohta jätetään pois ja annetaan vain aloitusmerkki, niin tulos on merkkijono aloitusmerkistä merkkijonon loppuun asti

```
>>> nimi = "Pekka Pelkonen"  
>>> print nimi[1:]  
ekka Pelkonen  
>>>
```

14

matti.tedre@cs.joensuu.fi



Merkkijonon alkupää

- Jos aloitusmerkki jätetään pois ja annetaan vain lopetuskohta (kirjainväli), niin tulos on merkkijono alkuperäisen merkkijonon alusta lopetuskohtaan asti

```
>>> nimi = "Pekka Pelkonen"  
>>> print nimi[:4]  
Pekk  
>>>
```

15

matti.tedre@cs.joensuu.fi



Merkkijonon käsittelyä

- Nyt osataan siis:
 - Yhdistellä merkkijonoja
 - Napata merkkijonosta pätkiä
- Tehdään ohjelma joka pyytää käyttäjältä nimeä ja suunnittelee käyttäjälle paremman nimen.

```
nimi = raw_input("Anna nimesi: ")  
parempi_nimi = "P" + nimi[1:]  
print "Parempi nimi olisi", parempi_nimi
```

16

matti.tedre@cs.joensuu.fi



Merkkijonon käsittelyä

- Esimerkki: irc-galleriaan tarvitaan ohjelmanpätkä joka luo tosi katu-uskottavia käyttäjätunnuksia.
- Ohjelma pyytää käyttäjältä etunimen, sukupuolen, ja syntymävuoden.
- Ohjelma luo irc-galleriaan sopivan käyttäjätunnuksen (ohjelman toiminta perustuu tarkkaan analyysiin siellä jo olemassa olevista käyttäjätunnuksista).

17

matti.tedre@cs.joensuu.fi



Irc-nick-ohjelma

- Ohjelma pyytää käyttäjältä kolme merkkijonoa: etunimen, sukupuolen, ja syntymävuoden.

```
print "Anna tietosi niin teen sinulle nickin"

nimi = raw_input("Nimesi: ")
vuosi = raw_input("Syntymävuotesi: ")
sukupuoli = raw_input("Tyttö vai poika (t/p): ")
```

18

matti.tedre@cs.joensuu.fi



Irc-nick-ohjelma (2)

- Luodaan etunimen, sukupuolen ja syntymävuoden perusteella nick.

```
if sukupuoli=="t":
    nick = nimi[:3] + "leena" + vuosi[2:4]
elif sukupuoli=="p":
    nick = nimi[:3] + "Zi" + vuosi[2:4]
else:
    nick = "Anonyymi"
```

19

matti.tedre@cs.joensuu.fi



Valmis irc-nick-ohjelma

```
print "Anna tietosi niin teen sinulle nickin"

nimi = raw_input("Nimesi: ")
vuosi = raw_input("Syntymävuotesi: ")
sukupuoli = raw_input("Tyttö vai poika (t/p): ")

if sukupuoli=="t":
    nick = nimi[:3] + "leena" + vuosi[2:4]
elif sukupuoli=="p":
    nick = nimi[:3] + "Zi" + vuosi[2:4]
else:
    nick = "Anonyymi"

print "Nickisi on", nick
```

20

matti.tedre@cs.joensuu.fi



Pasi Palindromimestarin ohjelma

- Pasista palindromit ovat hauskoja.
- Pasi harrastaa palindromien tekemistä.
- Pasi päättää tehdä Pythonilla ohjelman joka kääntää sille annetun sanan takaperin.

21

matti.tedre@cs.joensuu.fi



"matti"
01234

- Jos sanassa on viisi kirjainta, täytyy tulostaa merkit `sana[4]`, `sana[3]`, `sana[2]`, `sana[1]`, ja `sana[0]`.
- Tehdään alkuehtoinen toisto joka lähtee liikkeelle sanan viimeisestä kirjaimesta ja käy läpi ja tulostaa kaikki sanan kirjaimet ensimmäiseen asti (4,3,2,1,0 – i,t,t,a,m).

23

matti.tedre@cs.joensuu.fi



Pasi Palindromimestarin ohjelma

- Pasi ohjelman pitää
 - Lukea käyttäjältä sana
 - Tulostaa sanan kirjaimet lopusta alkuun

22

matti.tedre@cs.joensuu.fi



"matti"
01234

- 5-merkkisen sanan "matti" viimeinen kirjain on 4. lokerossa
- 12-merkkisen sanan viimeinen kirjain on 11. lokerossa
- Täytyisi tietää siis sanan pituus jotta tiedettäisiin mikä kirjain tulostetaan ensimmäiseksi

24

matti.tedre@cs.joensuu.fi



Merkkijonon pituus

- Pythonissa merkkijonon pituuden saa selville komennolla `len`

```
>>> kotikaupunki = "Tyrvää"
>>> print len(kotikaupunki)
6
>>> kotikaupunki = "Ii"
>>> print len(kotikaupunki)
2
>>> kotikaupunki = "Uusikaarlepyy"
>>> print len(kotikaupunki)
13
```

25

matti.tedre@cs.joensuu.fi



Suunnitellaan ohjelmaa...

```
>>> sana = "assi-kissa"
>>> print len(sana)
10
>>>
"assi-kissa"
0123456789

>>> print sana[0]
a
>>> print sana[9]
a
>>> print sana[len(sana)]
IndexError: string index out of range
```

26

matti.tedre@cs.joensuu.fi



Suunnitellaan ohjelmaa...

```
"assi-kissa"
0123456789
```

```
>>> print sana[9]+sana[8]+sana[7]+sana[6]+sana[5]
assik
>>>
```

27

matti.tedre@cs.joensuu.fi



Ratkaisu 1:

Tulostetaan merkkijono väärin päin

- Tulostetaan siis kaikki merkit viimeisestä ensimmäiseen:
- Viimeinen merkki on kohdassa `len(sana) - 1`
- Ensimmäinen merkki on kohdassa 0

```
>>> sana = "Assi-Kissa"
>>> kirjain_nro = len(sana)-1
>>> while kirjain_nro>=0:
    print sana[kirjain_nro],
    kirjain_nro = kirjain_nro - 1
```

```
a s s i K - i s s A
>>>
```

28

matti.tedre@cs.joensuu.fi



Parempi ratkaisu: Kootaan uusi merkkijono

- Jos tulostetaan niin että print-komennot erotetaan pilkulla, niin joka väliin tulee ylimääräinen välilyönti!
- Tehdään niin että lisätään ohjelmaan muuttuja johon *kootaan* uusi, väärin päin oleva merkkijono

- Alustetaan tyhjäksi
`toisin_pain = ""`



Parempi ratkaisu: Kootaan uusi merkkijono

- Ensimmäiseksi (tyhjään) merkkijonomuuttujaan lisätään viimeinen merkki
- Sitten toiseksi viimeinen merkki
- jne...
- Viimeiseksi merkkijonomuuttujaan lisätään merkkijonon ensimmäinen merkki

Pasin Palindromi-ohjelma

```
# toisin_pain: kokooja
# sana: tuoreimman säilyttäjä
# kirjain_nro: askeltaja

toisin_pain = ""

sana = raw_input("Anna sana: ")

kirjain_nro = len(sana)-1
while kirjain_nro>=0:
    toisin_pain = toisin_pain + sana[kirjain_nro]
    kirjain_nro = kirjain_nro - 1

print "Sana toisin päin on", toisin_pain
```

Muuttujien roolit Pasin Palindromi-ohjelmassa



```
toisin_pain = ""

sana = raw_input("Anna sana: ")

kirjain_nro = len(sana)-1
while kirjain_nro>=0:
    toisin_pain = toisin_pain + sana[kirjain_nro]
    kirjain_nro = kirjain_nro - 1

print "Sana toisin päin on", toisin_pain
```


Ohjelman rakenne

```
toisin_pain = ""
-----
sana = raw_input("Anna sana: ")
-----
kirjain_nro = len(sana)-1
while kirjain_nro>=0:
    toisin_pain = toisin_pain + sana[kirjain_nro]
    kirjain_nro = kirjain_nro - 1
-----
print "Sana toisin päin on", toisin_pain
```

33

matti.tedre@cs.joensuu.fi



Hermanni Historianopiskelijan ohjelma (2)

- Koska Hermanni ei osaa (vielä!) tehdä ohjelmaa joka laskisi ihan oikeasti *sanojen* määrän, Hermanni tekee ohjelman, joka laskee *välilyöntien* määrän. Tästä Hermanni voi päätellä esseiden pituuden.
- Ohjelmaansa Hermanni tarvitsee muuttujan johon koko essee (!) talletetaan

35

matti.tedre@cs.joensuu.fi



Hermanni Historianopiskelijan ohjelma

- Hermanni opiskelee historiaa.
- Hermannin pitää kirjoittaa paljon esseitä joiden minimisanamäärä on määritelty (esim. 500 sanaa)
- Hermanni haluaisi pitää esseet mahdollisimman lyhyinä, joten Hermanni päättää tehdä Python-kielisen ohjelman joka laskee esseiden pituuden.
 - Hermannin aineissa välilyönti erottaa sanat toisistaan

34

matti.tedre@cs.joensuu.fi



Hermanni Historianopiskelijan ohjelma (3)

- Hermanni tarvitsee myös muuttujan joka pitää yllä tietoa siitä montako välilyöntiä esseestä on löytynyt
 - Lähtee liikkeelle 0:sta ja jokainen löydetty välilyönti kasvattaa muuttujan arvoa yhdellä.
- Ja lopuksi Hermanni tarvitsee muuttujan joka käy läpi jokaisen merkin käyttäjän antamasta esseestä.
 - Lähtee liikkeelle esseiden alusta eli merkistä 0, ja käy läpi kaikki merkit viimeiseen asti; `len(esset) - 1`

36

matti.tedre@cs.joensuu.fi



Välilyöntilaskuri

- Pyydetään käyttäjää syöttämään merkkijono, talletetaan se muuttujaan nimeltä *essee*
- Alustetaan askeltaja *valeja* nolaksi
 - Ohjelman alussa ei ole tietenkään laskettu yhtään välilyöntiä
- Alustetaan askeltaja *kirjain* nolaksi
 - Lähtee liikkeelle ensimmäisestä kirjaimesta eli merkkijonon kohdasta 0

37

matti.tedre@cs.joensuu.fi



Välilyöntilaskuri (2)

```
essee = raw_input("Kirjoita essee: ")  
  
valeja = 0  
kirjain = 0
```

38

matti.tedre@cs.joensuu.fi



Välilyöntilaskuri (3)

- Tehdään toistolause joka käy läpi kaikki merkkijonon merkit, eli
 - lähtee liikkeelle 0:sta
 - lopettaa kun askeltaja menee merkkijonon viimeisen merkin ohi

"matti"

01234

39

matti.tedre@cs.joensuu.fi



Välilyöntilaskuri (4)

- Tarkastellaan jokaista merkkiä, ja jos merkki on välilyönti " ", niin kasvatetaan askeltajaa *valeja*
- Joka toistokierroksen loppuun siirrytään aina tarkastelemaan seuraavaa kirjainta

```
essee = raw_input("Kirjoita essee: ")  
  
valeja = 0  
kirjain = 0  
  
while kirjain < len(essee):  
    if essee[kirjain] == " "  
        valeja = valeja + 1  
    kirjain = kirjain + 1
```

40

matti.tedre@cs.joensuu.fi



Välilyöntilaskuri (5)

- Lopetetaan toisto kun askeltaja *kirjain* menee ohi merkkijonon lopun
- Lopuksi tulostetaan vastaus

```
essee = raw_input("Kirjoita essee: ")
valeja = 0
kirjain = 0

while kirjain<len(essee):
    if essee[kirjain]==" ":
        valeja = valeja + 1
    kirjain = kirjain + 1
print "Esseessä oli", valeja, "välilyöntiä."
```

41

matti.tedre@cs.joensuu.fi

JOENSUUN
YLIOPISTO

Kertaus / läpileikkaus käsitellyistä asioista

43

matti.tedre@cs.joensuu.fi

JOENSUUN
YLIOPISTO

Välilyöntilaskuri (6)

```
# essee: tuoreimman säilyttäjä
# valeja: askeltaja
# kirjain: askeltaja

valeja = 0
kirjain = 0

essee = raw_input("Kirjoita essee: ")

while kirjain<len(essee):
    if essee[kirjain]==" ":
        valeja = valeja + 1
    else:
        # ei tehdä mitään
        kirjain = kirjain + 1
```

42

```
print "Esseessä oli", valeja, "välilyöntiä"
```

matti.tedre@cs.joensuu.fi

JOENSUUN
YLIOPISTO

Mitä kurssilla ollaan opittu? ...Ohjelmoinnista

- Python-tulkin käyttö
- Ohjelman suunnittelu
 - Mietitään mitä ohjelma saa syötteenä ja mitä sen pitää tulostaa
 - Tarkennetaan suunnitelmaa asteittain
- Millaista tietoa tietokone käsittelee?
 - Dataa (merkkejä, merkkijonoja, numeroita)
 - Tietokoneelle data ei "merkitse" mitään—kuten vaa'alle sen näyttämä paino ei "merkitse" mitään.

44

matti.tedre@cs.joensuu.fi

JOENSUUN
YLIOPISTO

...Ohjelmoinnista

- Kaikki mitä tietokone osaa tehdä, myös ihminen osaa tehdä
 - Joissain tapauksissa tietokone tekee työn paljon nopeammin ja vähemmän virhealttiisti
- Tietokone osaa tehdä vain hyvin pienen osan niistä asioista joita ihminen osaa tehdä
 - Rutiininomaisia datankäsittelytehtäviä
 - Joitain toistoa ja tarkkuutta vaativia tehtäviä
 - Peruslaskutoimituksia ja vertailuja
 - Tiedon lukeminen, käsittely, ja tulostus

45

matti.tedre@cs.joensuu.fi



...Peruskäsitteistä

- *Esim:*
 - Tietokone
 - Tietokoneohjelma
 - Ohjelmointikieli
 - Muuttuja
- ...jne...
- ...jne...

46

matti.tedre@cs.joensuu.fi



...Muuttujista

- Ohjelmoijan nimeämiä tunnuksia joilla on joku arvo
 - Erilaisia tyyppejä (mm. kokonaisluku, merkkijono, desimaaliluku)
 - Erilaisia rooleja ohjelmassa (mm. kokooja, kiintoarvo, askeltaja...)
 - Rajoituksia nimeämiselle! (mm. ei ääkkösiä)
 - Voidaan *alustaa*, eli antaa alkuarvo, ja arvoa voidaan vaihtaa esim. käyttäjän syötteen, laskutoimituksen, tai toisen muuttujan arvon perusteella—riippuen muuttujan roolista ohjelmassa

47

matti.tedre@cs.joensuu.fi



...Muuttujista

- Tällä kurssilla tehdyissä ohjelmissa muuttujilla on ollut seuraavanlaisia rooleja:

- Kiintoarvo
- Askeltaja
- Tuoreimman säilyttäjä
- Sopivimman säilyttäjä
- Kokooja
- Tilapäissäilö



48

matti.tedre@cs.joensuu.fi



...Peruskäsitteistä (2)

- Kolme tärkeää käsitettä ohjelman suorituksessa
- Peräkkäisyys
 - Ohjelman suoritus etenee aina järjestelmällisesti yksi toiminto kerrallaan (yksiselitteisesti!)
- Valinta
 - Joitain toimintoja voidaan tehdä tai jättää tekemättä, riippuen siitä täytyvätkö halutut ehdot
- Toisto
 - Joitain toimintoja voidaan toistaa tietty määrä

```
if ika>18:
```

49

matti.tedre@cs.joensuu.fi



...Syötöstä ja tulostuksesta

- Olisi hyvä erottaa toisistaan:
- Tiedon syöttäminen
 - Tällä kurssilla tieto on luettu käyttäjältä
 - (Voidaan lukea muillakin tavoilla, esim. tiedostoista tai mittalaitteista)
- Tiedon tulostaminen
 - Tällä kurssilla tiedon tulostaminen ruudulle
 - (Voidaan tehdä toisinkin, esim tiedostoon)
- Tiedon käsittely

50

matti.tedre@cs.joensuu.fi



...Tulostuksesta

- Tällä kurssilla `print`-komennolla
- Python-kielessä `print`:illä voidaan tulostaa monenlaisia tyyppisiä (merkkijonoja, desimaalilukuja, kokonaislukuja, joukkoja...)
- Samalle riville tulostettavat asiat erotetaan toisistaan pilkulla

51

matti.tedre@cs.joensuu.fi



...Tiedon syöttämisestä

- Python-kielessä esim. komennoilla `raw_input` ja `input`.
- Näihin komentoihin liittyy *kehote* joka tulostetaan käyttäjälle.
- Tehdään usein sijoituslauseilla:

```
nimi = raw_input("Anna nimesi:")
```

52

matti.tedre@cs.joensuu.fi



...Peruslaskutoimituksista

- Laskutoimitukset
 - Yhteen, vähennys, kerto- ja jakolasku
 - * + - /
 - Miten laskutoimituksen tyyppi määräytyy
 - Jos laskutoimituksessa on vain kokonaislukuja, tulos on kokonaisluku.
 - Jos mukana on yhtään desimaalilukua, tulos on desimaaliluku.
 - Desimaalilukuja voidaan muuttaa kokonaisluvuiksi ja toisin päin-- `int()` ja `float()`
- Jakolaskussa kokonais- ja desimaalilukujen ero on usein erityisen tärkeä!

53

matti.tedre@cs.joensuu.fi

JOENSUUN
YLIOPISTO

...Muuttujien käytöstä

- *Sijoituslauseella* voi muuttaa muuttujan arvoa
 - Muuttuja voi olla sijoituslauseen yhtäsuurikuinmerkin (“operaattorin”) = molemmilla puolilla
 - Oikean puolen lausekkeen arvo lasketaan ensiksi
 - Laskettu arvo sijoitetaan vasemmalla puolella olevaan muuttujaan
 - Toisin sanoen: = -merkin vasemmalla puolella olevan muuttujan arvoksi tulee = -merkin oikealta puolelta saatu arvo
 - Muuttujalla on aina vain yksi arvo kerrallaan!

54

matti.tedre@cs.joensuu.fi

JOENSUUN
YLIOPISTO

...Vertailuista

- *Vertailulausekkeet*
 - “Vertailuoperaattorit”: < <= == >= > !=
 - Millaisia muuttujia voidaan vertailla keskenään?
 - Vertailun tulos on aina *tos*i tai *epätosi*
- Vertailujen yhdistäminen loogisilla operaattoreilla
 - “Boolean operaattorit”: `and`, `or`, `not`
 - Tällä kurssilla täytyy osata `and` ja `or`.
- Vertailulausekkeitä käytetään esimerkiksi ehtolauseissa ja alkuehtoisissa toistoissa

55

matti.tedre@cs.joensuu.fi

JOENSUUN
YLIOPISTO

...Ehtolauseista (*if..elif..else*)

- Ehtolauseilla suoritus “haarautuu”
- Jos vertailu on tosi (*if*):
 - suoritetaan sisennetty osuus
- Muussa tapauksessa, jos toinen vertailu on tosi (*elif*):
 - suoritetaan muu sisennetty osuus
- ...
- Kaikissa muissa tapauksissa (*else*):
 - suoritetaan vielä joku muu sisennetty osuus

56

matti.tedre@cs.joensuu.fi

JOENSUUN
YLIOPISTO

...Alkuehtoisestä toistosta

- Niin kauan kuin vertailu on tosi, suorita sisennetty osuus uudestaan ja uudestaan:
 - Tee jotain sisennetyt osuuden aikana
- Nimi *alkuehtoinen* tulee siitä että toistettavan osan alussa oleva ehto määrää toiston alkamisen ja jatkumisen
- Toistojen määrää ei tiedetä toistoa aloitettaessa!
 - Toistojen määrää ei ole edes rajoitettu (“*ikuinen silmukka*”)

57

matti.tedre@cs.joensuu.fi



...Askeltavasta toistosta

- Toistokertojen määrä tiedetään toistoa aloitettaessa
- *Askeltaja* käy läpi haluttuja arvoja
- Sisennetty osuus toistetaan jokaisella askeltajan saamalla arvolla.
- Tyypillinen esimerkki:
 - Luodaan range-komennolla joukko
 - Askeltaja käy for-lauseella läpi kaikki joukon alkiot
 - Jokaista askeltajan saamaa arvoa kohti suoritetaan sisennetty osuus

58

matti.tedre@cs.joensuu.fi



For-silmukasta

- For-lauseella voidaan Pythonissa käydä läpi *alkiojoukkoja*
- Esim. kokonaislukujen joukko [1, 2, 3, 4, 5, 6]
- Range-komennolla voidaan Pythonissa luoda alkiojoukkoja:
 - Range (aloitusarvo, lopetusarvo)
 - Range (aloitusarvo, lopetusarvo, hyppy)
 - Lopetusarvo on yläraja* joka ei kuulu enää joukkoon

59

* Kurssilla käsiteltiin lopetusarvoa *alarajana* vain ohimennen

matti.tedre@cs.joensuu.fi



...Sisäkkäisistä rakenteista

- Yhdellä sisennyksellä voidaan erottaa esim. toistettava tai ehdollisesti suoritettava osa ohjelmaa
- Sisennetyissä osissa voi niissäkin olla ehtoja tai toistettavia osia
- Käytetään astetta syvempää sisennystä erottamaan nämä osat

60

matti.tedre@cs.joensuu.fi



...Tyypillisistä virheistä

- “Päätymätön toisto” a.k.a. “ikuinen silmukka”
 - Toisto jossa alkuehto toteutuu aina, eli toisto jatkuu “ikuisesti”
- “Vailinainen ehtorakenne”
 - Ehtorakenne, jossa jollekin vaihtoehdolle ei ole omaa haaraa
- “Yhdellä huti”
 - Ohjelmat jotka toimivat melkein kaikissa tapauksissa oikein—mutta esim. viimeinen toisto jää puuttumaan tai jätetään merkkijonon 1. merkki (indeksissä 0) tarkastelematta

61

matti.tedre@cs.joensuu.fi

JOENSUUN
YLIOPISTO



...Merkkijonoista

- Merkkijonoja voidaan esimerkiksi
 - Määritellä kiintoarvoina tai muuttujina
 - Lukea käyttäjältä (`raw_input`)
 - Laittaa suoraan tulostuslauseisiin
 - Yhdistää toisten merkkijonojen kanssa + -operaattorilla
 - Tarkastella osittain tai merkki merkiltä (merkkien indeksien perusteella)
- Muista:
 - `4+2` on 6, mutta `"4"+"2"` on `"42"`

62

matti.tedre@cs.joensuu.fi

JOENSUUN
YLIOPISTO



...Merkkijonoista

- Merkkijono on itse asiassa lokerikko jossa joka lokerossa on yksi merkki
- Jokaisella lokerolla on numero (*indeksi*)
 - Lokeroiden numerointi alkaa nollassa
 - Ohjelma voi tutkia merkkijonon lokeroitten sisältöä (yksittäisiä merkkejä) indeksin perusteella
- Merkkijonon merkkien indeksit annetaan hakasuluissa
 - `sana[0], sana[1], ... sana[len(sana)-1]`

63

matti.tedre@cs.joensuu.fi

JOENSUUN
YLIOPISTO



...Merkkijonoista

- Hakasuluissa voi olla merkin indeksi tai `[aloitusmerkki:lopetusväli]`
- Aloitusmerkki ja/tai lopetusväli voivat olla tyhjiä, jolloin tarkoitetaan merkkijonoa “alusta lopetusväliin asti” tai “aloitusmerkistä loppuun asti”.
- `len` kertoo merkkijonon pituuden

64

matti.tedre@cs.joensuu.fi

JOENSUUN
YLIOPISTO

