Merkkijonot 2

Ohjelmoinnin perusteet

3

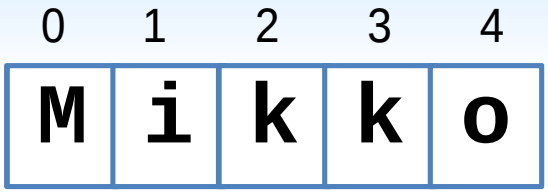
Turun yliopisto
University of Turku

Merkkijono (kertaus)

- Merkkijono on tyhjä tai se koostuu yhdestä tai useammasta merkistä.
- Merkkijonoihin tallennetaan siis yksittäisiä merkkejä, sanoja, lauseita tai vastaavia.

Merkkien indeksointi (kertaus)

- Merkkien indeksointi alkaa nollasta: ensimmäinen merkki merkkijonossa on siis indeksin nolla kohdalla, toinen indeksin 1 ja niin edelleen:



Merkkien poimiminen (2)

- Esimerkkejä:

```
mjono = "abcdef"
print mjono[0] # Tulostaa a
mj2 = mjono[1]
print mj2 # Tulostaa b
mjono3 = mjono[2] + mjono[4]
print mjono3 # Mitä tulostaa?
```

Merkkijonon pituus

- Merkkijonon pituus voidaan palauttaa funktion `len` avulla. Syntaksi:

```
len(<merkkijono>)
```

- Funktio palauttaa kokonaisluvun, joka kertoo merkkijonon merkkien määrän.

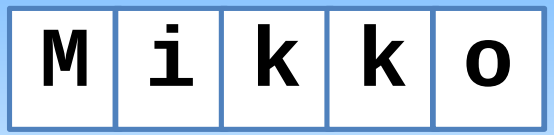
Esimerkkejä

```
mjono = "abcdef"
print len(mjono) # Tulostaa 6

mj2 = mjono[1] + mjono[3]
print len(mj2) # Tulostaa 2

pituus = len("ab" + "de")
print pituus * 2 # Mitä tulostaa?
```

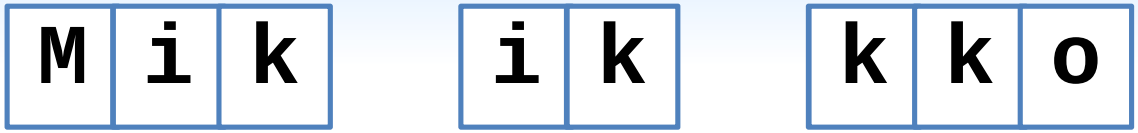
Merkkijonot ja alijonot

- Alijonolla tarkoitetaan jotain jonoa, joka löytyy merkkijonon sisältä. Esimerkiksi:

Merkkijono:



Alijonoja:



Miksi alijonoja?

- Alijonoilla on useita käyttötarkoituksia ohjelmissa.

- Esimerkiksi etunimen poimiminen koko nimestä, lauseen ensimmäisen sanan poimiminen tai maalimäärän poimiminen tuloksen ilmoittavasta merkkijonosta "87 – 23".

Alijonot

- Merkkijonosta voidaan poimia alijono []-operaattorin avulla.

Syntaksi:

```
<merkkijono>[ <alku> : <loppu> ]
```

Alijonot (2)

- Operaattori on siis sama kuin yksittäistä merkkiä poimittaessa.

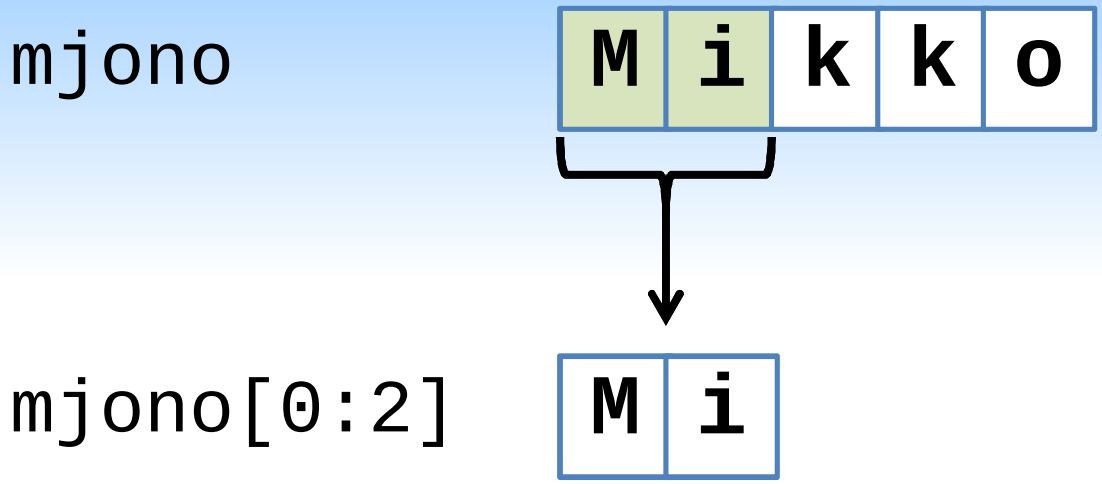
- Alijonoa poimiessa annetaan kuitenkin alijonon alkuindeksi ja loppuindeksi.

Alijonojen indeksoinnista

- Operaattorille annetaan siis alku- ja loppuindeksi, ja se palauttaa näiden välisen alijonon.

- Alijonossa on mukana alkuindeksin mukainen merkki, mutta **ei loppuindeksin kohdalla olevaa**.

Alijonojen indeksoinnista



Alijonojen indeksoinnista (2)

- Esimerkiksi:

```
mjono = "Python"
ali = mjono[0:2] # merkit 0 ja 1
print ali # Tulostaa Py
ali2 = mjono[1:5] # merkit 1-4
print ali2 # Tulostaa ytho
print "abcde"[2:4] # Tulostaa cd
```

Alijonojen indeksoinnista (3)

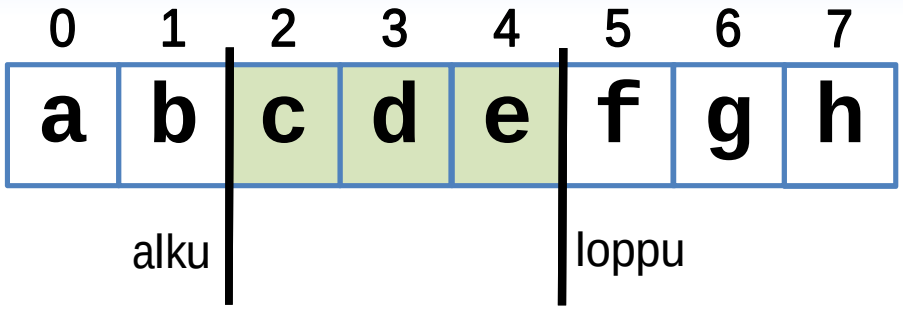
- Syy alku- ja loppuindeksien toimintaan on paitsi historiallinen (näin on useimmissa muissakin ohjelmointikielissä), luultavasti myös se, että näin alijonon pituuden saa laskettua kaavalla

loppuindeksi - alkuindeksi

Muistisääntö

- Alijonon indeksoinnin muistamista helpottaa, jos ajattelee alku- ja loppuindeksin merkin vasemmalle puolelle piirrettäviksi viivoiksi:

```
"abcdefgh"[2:5]
```



Merkkijonometodeista

- Pythonissa on useita metodeja, joiden avulla merkkijonojen käsittely on helppoa.

- Metodeiden avulla voidaan esimerkiksi etsiä merkkijonosta alijonoja tai korvata niitä toisilla alijonoilla.

Alijonon etsiminen

- Metodi `find` palauttaa tiedon siitä, mistä indeksistä alkaen annettu alijono löytyy merkkijonosta. Metodi palauttaa -1, jos alijonoa ei löydy merkkijonosta ollenkaan.

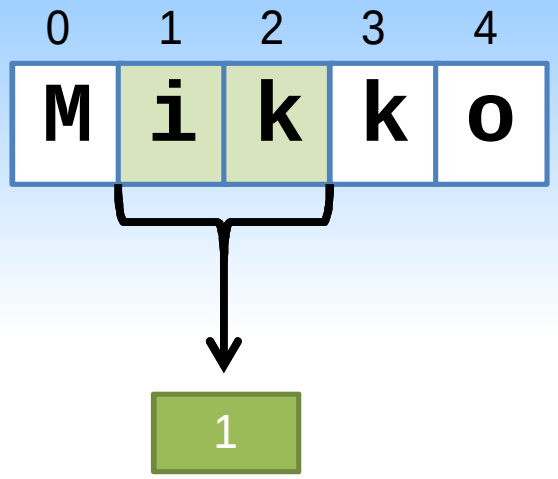
Syntaksi:

```
<merkkijono>.find(<alijono>)
```

Alijonon etsiminen

```
mjono
```

```
mjono.find("ik")
```



Alimerkkijonon etsiminen (2)

- Metodi palauttaa **kokonaislukuna** alijonon 1. indeksin:

```
mjono = "Python"
print mjono.find("Py") # Tulostaa 0
print mjono.find("hon") # Tulostaa 3
indeksi = mjono.find("ton")
print indeksi # Tulostaa -1
```

Alimerkkijonon korvaaminen

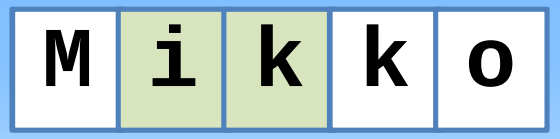
- Metodi `replace` palauttaa uuden merkkijonon, jossa kaikki ensimmäisen alijonon esiintymät on korvattu toisella alijonolla.

Syntaksi:

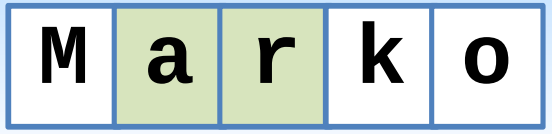
```
<merkkijono>.replace(<alijono1>, <alijono2>)
```

Alijonon korvaaminen

```
mjono
```



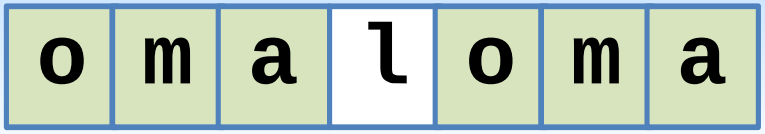

```
mjono.replace("ik","ar")
```



Alijonon korvaaminen (2)

```
mjono
```


```
mjono.replace("a","oma")
```



Esimerkkejä

```
mjono = "Python"
# Seuraava tulostaa Jython
print mjono.replace("P","J")
mjono2 = "kissa"
mjono3 = mjono2.replace("s","k")
print mjono3 # Tulostaa kikka

kor = "k"
mjono4 = mjono3.replace(kor,"")
print mjono4 # Mitä tulostaa?
```

Kysymys?

- Mitä seuraava ohjelma tulostaa? Miksi?

```
mjono = "Python"
mjono.replace("h","s")
print mjono
```

Vastaus

- Ohjelma tulostaa alkuperäisen merkkijonon **Python.** Replace-metodi **ei muuta** alkuperäistä merkkijonoa, vaan **luo uuden** jossa alijonot on korvattu.

- Jos tulos halutaan talteen, se pitää siis tallentaa johonkin muuttujaan.

Eli...

- Esimerkiksi näin:

```
mjono = "Python"
mjono2 = mjono.replace("h","s")
```

- ...tai tallentaen korvattu merkkijono samaan muuttujaan:

```
mjono = "Python"
mjono = mjono.replace("h","s")
```

Syötteiden lukeminen näppäimistöltä

- Usein ohjelmissa on tärkeää lukea käyttäjältä tietoa.

- Pythonissa on kaksi erillistä funktiota syötteiden lukemiseen:
  - **`input` lukujen lukemista** varten, ja
  - **`raw_input` merkkijonojen lukemista** varten.

Lukujen lukeminen: input

- Funktiolle **input** annetaan tietona käyttäjälle näytettävä kysymys, ja se palauttaa käyttäjän syöttämän luvun:

```
ika = input("Kuinka vanha olet?")
print "Olet siis",ika,"vuotta."
```

Merkkijonojen lukeminen: raw_input

- Funktio raw_input toimii muuten samalla tavalla, mutta sillä voidaan lukea merkkijonoja:

```
nimi = raw_input("Anna nimesi:")
print "Terve, " + nimi + "!"
```

Esimerkkejä

```
luku1 = input("Anna luku 1:")
luku2 = input("Anna luku 2:")
print "Lukujen keskiarvo:",
print (luku1 + luku2) / 2.0

nimi = raw_input("Anna koko nimesi:")
print "Nimessäsi on", len(nimi), "merkkiä."
print "Etunimesi on ",nimi[0:nimi.find(" ")]
```