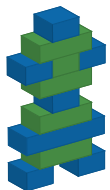


Merkkijonot

Ohjelmoinnin perusteet

2



Merkkijono

- Merkkijono on nollasta, yhdestä tai useammasta **merkistä** koostuva **jono**.
- Merkkijonoihin tallennetaan siis yksittäisiä merkkejä, sanoja, lauseita tai vastaavia.

Merkit merkkijonossa

- Merkkijonon merkit voivat olla mitä tahansa tietokoneen tukemia merkkejä

Esimerkiksi

- kirjaimia (a, b, c..., A, B, C...),
- numeroita (1, 2, 3...) tai
- erikoismerkkejä (. , ; : * @ jne...)

Merkkijono Pythonissa

- Pythonissa merkkijono kirjoitetaan lainausmerkkien sisään. Tyhjä merkkijono luodaan kirjoittamalla lainausmerkit peräkkäin ilman sisältöä.

```
nimi = "Paula Python"
```

```
etunimi = "Mikko"
```

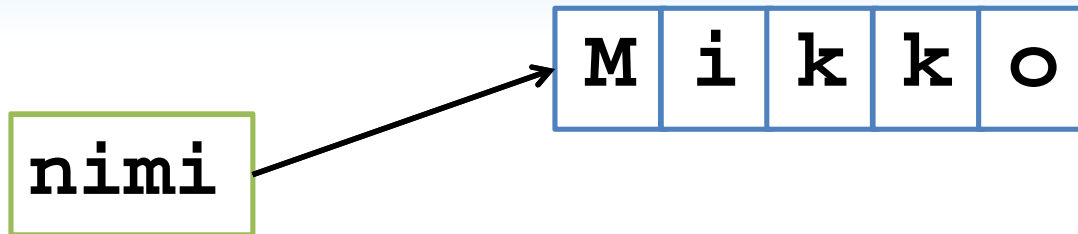
```
tyhja = ""
```

```
eka = "a"
```

Merkkijono on olio

- Kun merkkijono luodaan Pythonissa, se luodaan muistiin ja muuttujaan tallennetaan viittaus merkkijonoon muistissa.

```
nimi = "Mikko"
```



Merkkijono on olio (2)

- Luomisen jälkeen merkkijonomuuttujaa voidaan käyttää lukemaan merkkijono muistista. Käytännössä tämä toimii samalla tavalla kuin lukuja käytettäessä:

```
nimi = "Paula Python"  
print nimi # Tulostaa Paula Python
```

Merkkijono-operaattorit

- Muutamia Pythonin operaattoreita toimivat myös merkkijonojen kanssa:

Operaattori	Merkitys	Esimerkki
+	Katenoi (eli yhdistää) kaksi merkkijonoa yhdeksi	<pre>s1 = "abc" s2 = "def" s3 = s1 + s2 # abcdef</pre>
*	Monistaa merkkijonoa annetun määrän	<pre>s1 = "abc" s2 = s1 * 2 print s2 # abcabc</pre>

Operaattorit (2)

- Esimerkkejä:

```
etunimi = "Paula"  
sukunimi = "Python"  
# Yhdistä nimet, välilyönti väliin  
kokoNimi = etunimi + " " + sukunimi  
  
merkki = "*" #  
merkit = merkki * 10  
print merkit # Tulostaa *****
```


Merkkijonojen yhdistäminen eli katenointi

- + -operaattorin avulla voidaan siis yhdistää kaksi tai useampia merkkijonoja yhdeksi uudeksi merkkijonoksi.
- Operaattori toimii kuitenkin vain merkkijonojen kanssa, sillä ei voi yhdistää lukutyyppejä arvoihin.

Esimerkki

- Tämä toimii:

```
nimi = "Pekka" + " " + "Python"
```

- ...mutta tämä antaa virheilmoituksen:

```
ika = 18 + " vuotta"
```

Lukujen ja merkkijonojen katenointi

- Merkkijonoihin voidaan yhdistää muun tyyppisiä arvoja käyttämällä `str`-funktiota. Funktio muuntaa parametrina annetun arvon merkkijonoksi:

```
ika = 18
```

```
ika_mjono = str(ika) + " vuotta"
```

Lukujen ja merkkijonojen katenointi (2)

- Print-lauseessa voidaan myös käyttää pilkkua erottamaan eri tyyppisiä arvoja:

```
ika = 18  
mjono = "Ikä on"  
print mjono, ika, "vuotta."
```

Merkit merkkijonossa

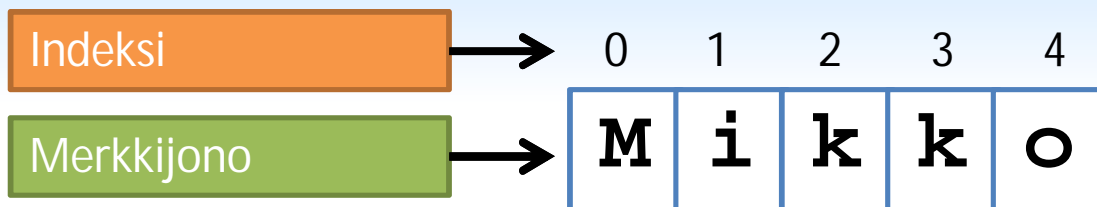
- Koska merkkijono on merkeistä koostuva jono, voidaan jonon merkkejä käsitellä myös yksittäin.
- Ohjelmassa voi olla esimerkiksi tarve poimia henkilön nimen alkukirjain tai lauseen lopussa oleva välimerkki.

Indeksi

- **Indeksi** on ohjelmoinnissa tärkeä käsite.
- Sillä tarkoitetaan jonkin **alkion järjestysnumeroa**.
- Esimerkiksi merkin indeksi merkkijonossa kertoo **miltä paikalta jonossa merkki löytyy**: ensimmäisen merkin indeksi on pienempi kuin toisen merkin ja niin edelleen.

Merkkien indeksointi

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



Merkin poimiminen merkkijonosta

- Yksittäinen merkki merkkijonosta voidaan palauttaa [] -operaattorin avulla.
- Syntaksi:

```
<merkkijono>[ <merkin indeksi> ]
```


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.

Merkkijonon pituudesta

- Merkkijonon pituuteen lasketaan kaikki sen sisältämät merkit, myös välilyönnit ja erikoismerkit:

Merkkijono	Pituus
"Moi "	3
"Hei kaikki "	9
" "	1
"Tämä on lause."	14
" "	0

Esimerkkejä

```
mjono = "abcdef"
```

```
print len(mjono) # Tulostaa 6
```

```
mj2 = mjono[1] + mjono[3]
```

```
print len(mj2) # Tulostaa 2
```

```
pituus = len("ab" + "cd")
```

```
print pituus * 2 # Mitä tulostaa?
```

Merkkien poimiminen(3)

- Koska merkkijonon ensimmäinen merkki on indeksin nolla kohdalla, on merkkijonon viimeinen merkki vastaavasti indeksin pituus - 1 kohdalla:

```
mjono = "Python"  
eka = mjono[0]  
vika = mjono[len(mjono) - 1]  
print eka, vika # Tulostaa P n
```