MERKKIJONOT

Ohjelmoinnin peruskurssi yläkouluun

© 2021 Oppimisanalytiikan keskus / Erkki Kaila, Heidi Kaarto ja Heidi Laine, kuvat Johan Sjövall

TURUN YLIOPISTO
Oppimisanalytiikan keskus

MERKKIJONO

- Merkkijonot ovat yksi Pythonin tyypeistä.
- **Merkkijono** on *nollasta, yhdestä tai useammasta merkistä koostuva jono*. Ne voivat olla siis esimerkiksi yksittäisiä merkkejä tai lauseita.
- Merkkijonon merkit voivat olla *mitä tahansa tietokoneen tukemia merkkejä* kuten kirjaimia, numeroita tai erikoismerkkejä.

MERKKIJONO

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

```
""
"Moi!"
"Syntymäpäiväjuhlat: 23.3. klo 15:00"
```

MERKKIJONOMUUTTUJAT

- Merkkijonomuuttujalle annetaan luomisen yhteydessä tunniste ja arvo kuten muillekin muuttujille.
- Syntaksi:

```
<nimi> = <merkkijono>
```

- Esimerkki:

```
tervehdys = "Hei kaikille."
```

TURUN YLIOPISTO
Oppimisanalytiikan keskus

ESIMERKKI

```
tervehdys = "Hei vaan kaikille."
print(tervehdys) # Tulostus: Hei vaan kaikille.
tervehdys = "Mitä kuuluu?"
print(tervehdys) # Mitä tämä tulostaa?
```

ESIMERKKI

```
tervehdys = "Hei vaan kaikille."
print(tervehdys) # Tulostus: Hei vaan kaikille.
tervehdys = "Mitä kuuluu?"
print(tervehdys) # Mitä tämä tulostaa?
```

Mitä kuuluu?

MERKKIJONO-OPERAATTORIT

- Pari Pythonin matemaattisista operaattoreista toimivat myös merkkijonojen kanssa:

Operaattori	Merkitys	Esimerkki
+	merkkijonojen yhdistäminen	`m = "abc"` `m2 = m + "def" # abcdef`
*	merkkijonojen monistaminen	`m = "abc"` `m2 = m * 3 # abcabcabc`

ESIMERKKI

```
etunimi = "Paula"
sukunimi = "Python"
# Yhdistetään nimet ja laitetaan välilyönti väliin.
kokoNimi = etunimi + " " + sukunimi

merkki = "|"
merkit = merkki * 5
print(merkit) # Tulostus: |||||
```

TURUN YLIOPISTO
Oppimisanalytiikan keskus

MERKKIJONOJEN YHDISTÄMINEN

- + -operaattorin avulla voidaan siis yhdistää kaksi tai useampia merkkijonoja yhdeksi uudeksi merkkijonoksi.
- **Operaattori toimii kuitenkin vain merkkijonojen kanssa**: sillä ei voi yhdistää muuntyyppisiä (esim. lukutyyppisiä) arvoja merkkijonoihin.

ESIMERKKI

- Tämä toimii:

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

- …mutta tämä antaa virheilmoituksen:

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

X

INDEKSI

- **Indeksit** kertovat merkkijonon *merkkien järjestysnumerot* eli sen, miltä paikalta jonossa merkki löytyy.
- *Merkkien indeksointi alkaa **nollasta**:* ensimmäinen merkki on siis indeksin nolla kohdalla, toinen indeksin yksi kohdalla ja niin edelleen.

0	1	2	3	4
V	i	L	L	E

YKSITTÄISTEN MERKKIEN POIMIMINEN

- Merkkijonon merkkejä voidaan käsitellä myös *yksittäin poimimalla ne merkkijonosta*.
- Ohjelmassa voi olla esimerkiksi tarve poimia henkilön nimen alkukirjain tai lauseen lopussa oleva välimerkki.

YKSITTÄISTEN MERKKIEN POIMIMINEN

- **Yksittäinen merkki** merkkijonosta voidaan poimia [ ]-operaattorin avulla.
- Operaattorin sisälle kirjoitettu numero kertoo, *minkä indeksin kohdalla oleva merkki halutaan poimia*.

YKSITTÄISTEN MERKKIEN POIMIMINEN

- Syntaksi:

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

- Esimerkki:

```
nimi[3]
"Pekka"[4]
```

ESIMERKKI

```
nimi = "Python"
print(nimi[0]) # Tulostus: P
kirjain = nimi[1]
print(kirjain) # Tulostus: y
jono = nimi[0] + nimi[1] + nimi[3]
print(jono) # Mitä tämä tulostaa?
```

ESIMERKKI

```
nimi = "Python"
print(nimi[0]) # Tulostus: P
kirjain = nimi[1]
print(kirjain) # Tulostus: y
jono = nimi[0] + nimi[1] + nimi[3]
print(jono) # Mitä tämä tulostaa?
```

Pyh

TURUN YLIOPISTO
Oppimisanalytiikan keskus

ALIJONOT

- **Alijonolla** tarkoitetaan jonoa, joka *löytyy merkkijonon sisältä*.

Merkkijono	V	i	L	L	E

Alijonoja	V	i	L	i	L	L	L	E

TURUN YLIOPISTO
Oppimisanalytiikan keskus

ALIJONOT

- Alijonoilla on useita käyttötarkoituksia ohjelmissa.
- Esimerkiksi etunimen poimiminen koko nimestä, lauseen ensimmäisen sanan poimiminen tai maalimäärän poimiminen tuloksen ilmoittavasta merkkijonosta "87-23".

ALIJONOJEN POIMIMINEN

- Alijonojen poimimisessa operaattori on sama kuin yksittäisten merkkien poimimisessa eli [ ].
- Operaattorille annetaan kuitenkin kaksi arvoa: **alijonon alku- ja loppuindeksit**.

ALIJONOJEN POIMIMINEN

- Syntaksi:

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

- Esimerkki:

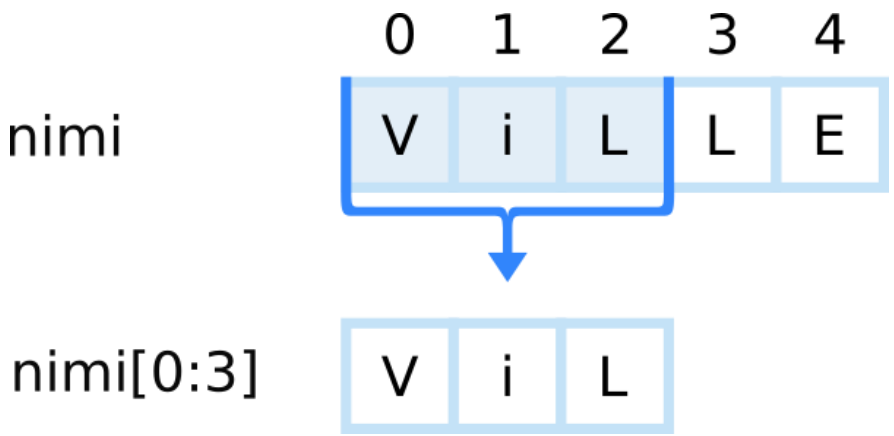
```
kirjaimet = "Pekka"[2:4] # kk

terveyhdys = "Hei kaikille."
alku = tervehdys[0:3] # Hei
```

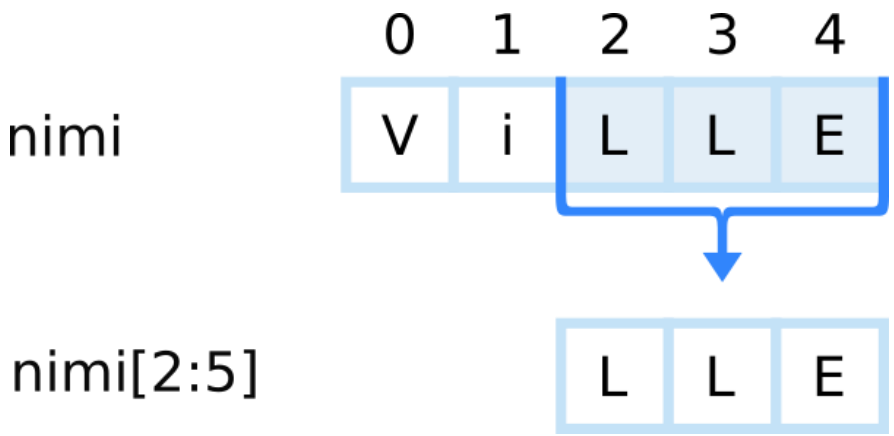
TURUN YLIOPISTO
Oppimisanalytiikan keskus

ALIJONOT

- **Alijonossa on mukana alkuindeksin mukainen merkki, mutta ei loppuindeksin kohdalla olevaa merkkiä.**
- Jos alkuindeksi jätetään pois, Python olettaa sen paikalle nollan. Jos loppuindeksi jää pois, Python olettaa sen paikalle merkkijonon pituuden.

ALIJONOJEN INDEKSIT





TURUN YLIOPISTO
Oppimisanalytiikan keskus

ESIMERKKI

```
nimi = "Python"
jono1 = nimi[0:2] # Merkit 0 ja 1
print(jono1) # Tulostus: Py
jono2 = nimi[1:5] # Merkit 1-4
print(jono2) # Tulostus: ytho
print("abcde"[2:4]) # Mitä tämä tulostaa?
```

ESIMERKKI

```
nimi = "Python"
jono1 = nimi[0:2] # Merkit 0 ja 1
print(jono1) # Tulostus: Py
jono2 = nimi[1:5] # Merkit 1-4
print(jono2) # Tulostus: ytho
print("abcde"[2:4]) # Mitä tämä tulostaa?
```

cd

TURUN YLIOPISTO
Oppimisanalytiikan keskus

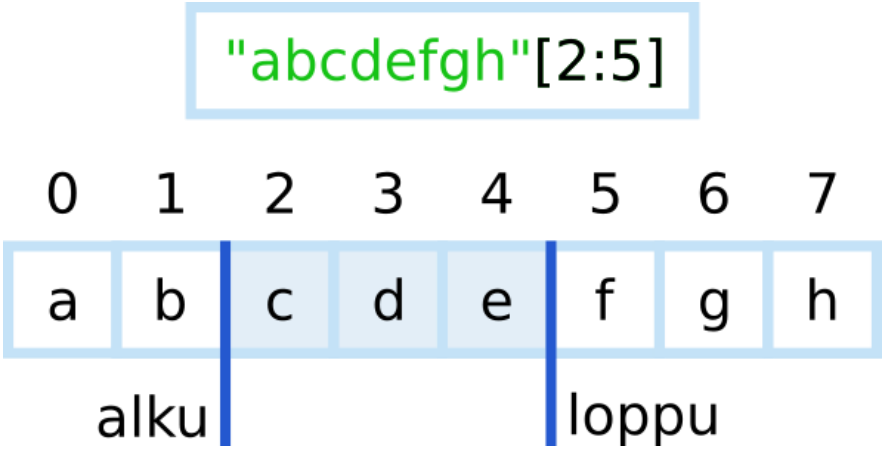
ALIJONOT

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

$$loppuindeksi - alkuindeksi$$

ALIJONOJEN INDEKSIEN MUISTISÄÄNTÖ

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



TURUN YLIOPISTO
Oppimisanalytiikan keskus

TURUN YLIOPISTO

Oppimisanalytiikan keskus