

MERKKIJONOT

Ohjelmoinnin peruskurssi yläkouluun

© 2019 Oppimisanalytiikan keskus / Erkki Kaila, Heidi Kaarto ja Heidi Laine, kuvat Johan Sjövall



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

KERTAUS

- Merkkijono voi olla tyhjä tai koostua yhdestä tai useammasta merkistä.
- Merkkijonoihin tallennetaan yksittäisiä merkkejä, sanoja, lauseita tai vastaavia.
- Indeksi kertoo merkin järjestysnumeron jonossa.



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

KERTAUS

- Merkkien indeksointi alkaa nolasta.
- Ensimmäinen merkki merkkijonossa on indeksin nolla kohdalla, toinen indeksin yksi ja niin edelleen.

| | | | | |
|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 |
| V | i | L | L | E |



KERTAUS

- Merkkijonojen yhdistäminen tapahtuu + -operaattorin avulla:

```
nimi = "Pekka" + " " + "Python"  
print(nimi) # tulostaa Pekka Python
```

- Merkin tai alijonon poimiminen merkkijonosta tapahtuu []-operaattorilla:

```
merkkijono = "Python"  
print(merkkijono[4]) # tulostaa o  
print(merkkijono[1:3]) # tulostaa yt
```



MERKKIJONOMETODIT

- Metodit ovat Pythonissa **valmiina olevia ohjelmia**, joita voidaan käyttää apuna ohjelman kirjoittamisessa.
- Pythonissa on useita metodeja, joilla merkkijonoja voidaan käsitellä.



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

LUKUJEN JA MERKKIJONOJEN YHDISTÄMINEN

- Merkkijonoihin voidaan yhdistää muun tyyppisiä arvoja käyttämällä `str()`-metodia.
- Metodi **muuttaa parametrina annetun arvon merkkijonoksi:**

```
ika = 18
```

```
ika_jono = str(ika) + " vuotta" # "18 vuotta"
```



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

LUKUJEN JA MERKKIJONOJEN YHDISTÄMINEN

- Ruudulle tulostaessa voidaan käyttää myös **pilkkoa** erottamaan eri tyyppisiä arvoja:

```
ika = 17 # Lukutyyppinen arvo  
print("Hän on", ika, "vuotta vanha.") # toimii
```



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

MERKKIJONON PITUUS

- **Merkkijonon pituus** voidaan palauttaa metodin `len()` avulla.
- Metodin **sulkujen sisälle** kirjoitetaan merkkijono tai sen merkkijonomuuttujan nimi, jonka sisältämän merkkijonon pituus halutaan selvittää.
- Metodi **palauttaa kokonaisluvun**, joka kertoo merkkijonon merkkien määrän.



MERKKIJONON PITUUS

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

| Merkkijono | Pituus |
|------------------|--------|
| "Moi" | 3 |
| "Hei kaikki" | 10 |
| " " | 1 |
| "Tämä on lause." | 14 |
| "" | 0 |



ESIMERKKI

```
mjono = "abcdef"  
print(len(mjono)) # tulostaa 6  
  
mjono2 = mjono[1] + mjono[3]  
print(len(mjono2)) # tulostaa 2  
  
pituus = len("ab" + "cd")  
print(pituus * 2) # tulostaa mitä?
```

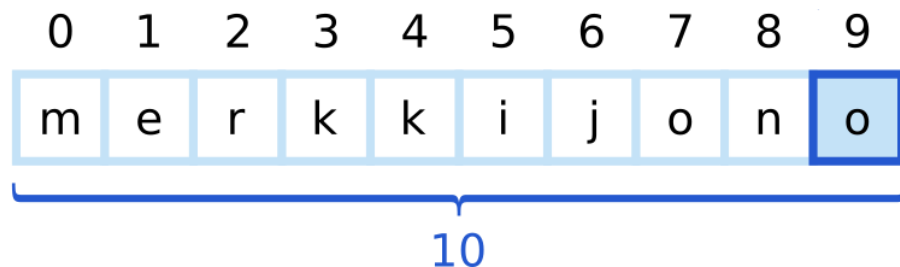


MERKKIJONON PITUUS

- Merkkijonon viimeinen merkki löytyy indeksin $\text{len}(\text{merkkijono}) - 1$ kohdalta.

```
s = "merkkijono"
```

```
n = len(s) # 10
```



```
s = "merkkijono"
```

```
n = len(s) - 1 # 9
```

```
k = s[n] # s[9] = "o"
```



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

ESIMERKKI

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



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

ALIJONON ETSIMINEN

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

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



**TURUN
YLIOPISTO**

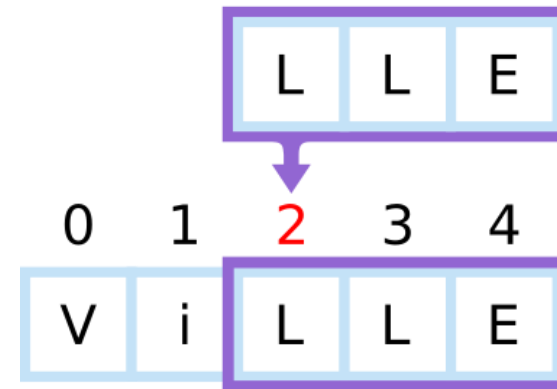
Oppimisanalytiikan keskus

ALIJONON ETSIMINEN

- Metodi palauttaa kokonaislukuna alijonon 1. indeksin:

```
nimi = "ViLLE"
```

```
indeksi = nimi.find("LLE") # 2
```



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

ESIMERKKI

```
mjono = "Python"  
print(mjono.find("Py")) # tulostaa 0  
print(mjono.find("hon")) # tulostaa 3  
print(mjono.find("ton")) # tulostaa -1  
  
sanoja = "ei ole helppoa ei"  
print(sanoja.find("ei")) # tulostaa 0
```



ALIJONON KORVAAMINEN

- Metodi `replace()` **luo uuden merkkijonon**, jossa kaikki **ensimmäisen alijonon esiintymät on korvattu toisella alijonolla.**

`<merkkijono>.replace(<korvattava>, <korvaava>)`



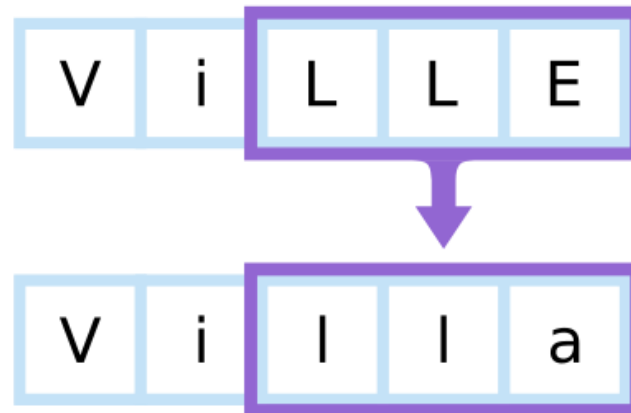
**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

ALIJONON KORVAAMINEN

```
nimi = "ViLLE"
```

```
vaihto = nimi.replace("LLE", "lla") # "Villa"
```



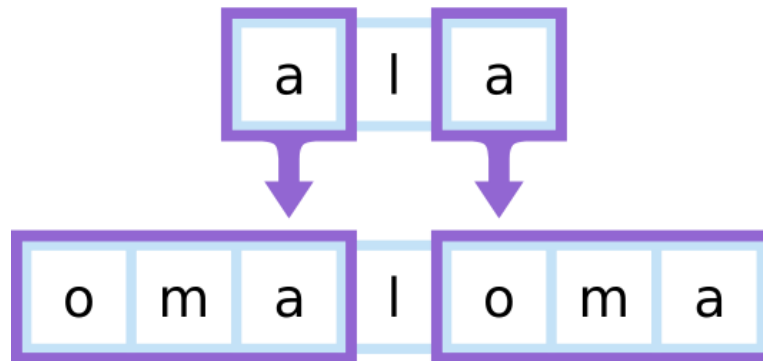
**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

ALIJONON KORVAAMINEN

```
mjono = "ala"
```

```
vaihto = mjono.replace("a", "oma") # "omaLoma"
```



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

ESIMERKKI

```
mjono = "Python"  
print(mjono.replace("P", "J")) # tulostaa Jython  
elain = "kissa"  
sana = elain.replace("s", "k") # kikka  
  
kirjain = "k"  
sana = sana.replace(kirjain, "")  
print(sana) # tulostaa mitä?
```



KYSYMYS

- Mitä seuraava ohjelma tulostaa? Miksi?

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



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

VASTAUS

- Ohjelma tulostaa alkuperäisen merkkijonon Python.
- `replace`-metodi **ei muuta alkuperäistä merkkijonoa**, vaan luo uuden merkkijonon, jossa alijonot on korvattu.
- Jos uusi merkkijono halutaan talteen, se pitää siis tallentaa johonkin muuttujaan.



ELI...

- ...esimerkiksi näin:

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

- ...tai korvaten muuttujan merkkijono uudella merkkijonolla:

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



SYÖTTEIDEN LUKEMINEN

- Usein ohjelmissa on tärkeää **lukea käyttäjältä tietoa**.
- Pythonissa tämä toteutetaan `input`-lauseella.

```
input(<kysymys tai ohje>)
```

- Käyttäjän **syöte täytyy tallentaa muuttujaan**, jotta sitä voidaan käyttää ohjelmassa myöhemmin.



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

ESIMERKKI

```
nimi = input("Mikä on nimesi?") # syöte: Ville  
print("Hei", nimi) # tulostaa Hei Ville
```



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

SYÖTTEIDEN LUKEMINEN

- Käyttäjän antama tieto välitetään ohjelmalle aina **merkkijonotyyppisenä**.
- Käyttäjän syöttämä tieto voidaan muuttaa toisen tyyppiseksi **muunnosfunktioiden** avulla.



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

MUUNNOSFUNKTIOT

| Funktio | Selitys | Esimerkki |
|----------------------|---|---|
| <code>str()</code> | Muuntaa minkä tahansa arvon merkkijonoksi (<i>string</i>) | <code>lukupono = str(15) # "15"</code> |
| <code>int()</code> | Muuntaa desimaalilukuja tai numeroita (ei pistettä) sisältäviä merkkijonoja kokonaisluvuiksi (<i>integer</i>) | <code>luku = int("38") # 38</code> <code>desimaali = 4.65</code> <code>kokoLuku = int(desimaali) # 4</code> |
| <code>float()</code> | Muuntaa kokonaislukuja tai numeroita (ja pisteen) sisältäviä merkkijonoja desimaaliluvuiksi (<i>float</i>) | <code>desimaali = float(42) # 42.0</code> <code>desimaali2 = float("53.2") # 53.2</code> <code>desimaali3 = float("5") # 5.0</code> |



MUUNNOSFUNKTIOT

- Funktio on metodin alalaji: niitä käsitellään myöhemmin tarkemmin.
- Metodi `str()`, joka esiteltiin aiemmin, on siis yksi muunnosfunktioista. Se muuttaa minkä tahansa tyyppisen arvon merkkijonoksi.
- Muiden muunnosfunktioiden kanssa täytyy olla tarkkana. **Merkkijonon voi muuntaa luvuksi vain, jos se sisältää numeroita.**



```
ika = input("Mikä on ikäsi?") # Merkkijonosyöte: 17
# Muunnetaan syöte kokonaisluvuksi.
ika = int(ika)
# Nyt muuttujalla voidaan laskea.
uusiIka = ika + 5
# Muutetaan uusi ikä merkkijonoksi tulostuksessa.
print("Viiden vuoden päästä olet " + str(uusiIka) + " vuotta
vanha.")
# Tulostaa: Viiden vuoden päästä olet 22 vuotta vanha.
```



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus