



Aalto University
School of Science

Ohjelmointia

Oliot

Mitä oliot ovat?

- Oliot ovat tietorakenteita
 - joihin tallennetaan tietoa ja
 - joiden avulla tuota tallennettua tietoa voidaan käsitellä
- Tiedon esitystapa ja –muoto määrittää tiedon käsittelyä
 - mitä on helppoa ja luontaista tehdä
 - mitä on vaikeaa ja monimutkaista tehdä
- Pythonissa kaikki varsinainen tieto on esitettyinä oliona
 - myös funktiot ovat Pythonissa olioita
- Oliot ovat yksilöitä
 - mutta samanlaisilla olioilla on sama tyyppi

<https://docs.python.org/3/tutorial/classes.html>

Oliot ja nimet, nimiavaruus ja näkyvyysalue

- Oliot ovat viitattavissa nimien kautta
 - Jos olioon ei päästä käsiksi minkään nimen kautta, se voidaan itse asiassa poistaa koko järjestelmästä
 - Nimet ovat tietyssä yhteydessä sidottu (bound) tiettyyn olioon
 - Samalla oliolla voi olla monta nimeä ja sama nimi voi eri paikoissa viitata eri olioon (huom. nimiavaruus ja näkyvyys)
- Nimiavaruus on sidonta (mapping) nimistä olioihin
 - Nimiavaruudessa tietty nimi on yksikäsitteinen
 - Nimiavaruuksia on useita (erityisesti attribuutit: olio.arvo)
 - Nimet eivät näy nimiavaruuksien välillä
- Näkyvyysalueet (scope): nimiavaruuden tekstuaalinen laajuus
 - Missä tuo sidonta on näkyvillä (esimerkiksi funktioiden argumentit)
 - Näkyvyysalueita on useampia sisäkkäin ja ajallisesti peräkkäin

Luokat (class) ja oliot (object)

- Luokkien avulla määritetään tiedon esitystapa ja –muoto
 - Eli siis millainen olio on kyseessä
 - Luokka määrittelee olion tiedot ja mahdolliset operaatiot
 - On siis hyvä tapa **lähteä liikkeelle luokkarakenteesta**
 - Pythonissa luokat toimivat olioiden tyyppeinä, esim.
sinuhe = {Book} <book.Book object at 0x10f99e3d0>
number = {int} 3
- Yksittäistä oliota sanotaan luokkansa instanssiksi
 - Uuden olion tekeminen luo oliolle oman nimiavaruuden
 - Attribuuteilla viitataan oliossa olevaan tietoon, myös sen metodeihin
 - Olion instanssi luodaan käyttämällä olion konstruktoria, esim
lista = [1, 2, 3]
brothers = Book("Aleksis Kivi" "7 brothers", 1870)

Esim: Book

```
class Book:
```

```
    def __init__(self, author_name,
book_name, publishing_year):
        self.author = author_name
        self.title = book_name
        self.year = publishing_year
        self.edition = 1
        self.publisher = ""
```

```
        [...]
    def compare_year(self, other_book):
        if self.edition < other_book.get_year():
            return True
        else:
            return False
```

```
    def __str__(self):
        return self.author + ": " + self.title
```

```
import book # or from stuff import book
```

```
def main():
    brothers = book.Book("Aleksis Kivi",
                          "Seven brothers", 1870)
    sinuhe = book.Book("Mika Waltari",
                       "Sinuhe The Egyptian",
                       1945)
```

```
    print(brothers.get_title())
    print(sinuhe.get_title())
    print("Sinuhe is published \
          before 7 brothers:",
          sinuhe.compare_year(brothers))
```

```
main()
```