

# Ohjelmoinnin peruskurssi Y1

CS-A1111

Olioiden metodeista vielä vähän lisää

## Oppimistavoitteet: tämän videon jälkeen

- ▶ Tiedät, miten olioiden tiedot voi tulostaa.
- ▶ Osaat käyttää olioita yhdessä toisten olioiden kanssa.

## Merkkijonoesitys oliosta

- ▶ Ohjelmissa halutaan hyvin usein tulostaa jonkin olion kaikkien kenttien arvot.
- ▶ Tulostaminen helpottuu, jos luokkaan määritellään metodi, joka tekee oliosta merkkijonoesityksen.
- ▶ Metodi palauttaa merkkijonon, joka sisältää olion kenttien arvot tai muuta haluttua tietoa oliosta.
- ▶ Tämän metodin nimeksi annetaan `__str__`. Tällöin olion tiedot voi tulostaa (esimerkiksi pääohjelmassa) käyttämällä suoraan olioon viittaavan muuttujan nimeä, esimerkiksi

```
print(kurssilainen1)
```

## Merkkijonoesitys Opiskelija-olioista

```
def __str__(self):
    mjono = self.__nimi + ", " + \
            self.__opiskelijanumero + \
            ", tenttiarvosana: " + \
            str(self.__tenttiarvosana) + \
            ", harjoitusarvosana: " + \
            str(self.__harjoitusarvosana)
    return mjono
```

## Opiskelijoiden tietojen tulostus, koodi

```
print("1. opiskelijan tiedot:")
print(kurssilainen1)
print("Kurssiarvosana:",
      kurssilainen1.laske_kokonaisarvosana())

print("2. opiskelijan tiedot:")
print(kurssilainen2)
print("Kurssiarvosana:",
      kurssilainen2.laske_kokonaisarvosana())
```

## Olion kenttänä voi olla mitä vaan

- ▶ Olion kenttänä voi olla vaikka toinen olio tai lista, ihan mitä vaan tarvitsee.
- ▶ Esim. jos kurssin harjoitustyöt vois tehdä myös parityönä, voisi Opiskelija-oliossa olla kenttä `pari`, johon tallennettaisiin toisen opiskelijan tiedot, jos opiskelijalla on pari:
- ▶ Opiskelijan arvostelutietoihin voisi tallentaa vaikka tenttitehtävien pistemäärät listana.
- ▶ Tällöin voitaisiin opiskelijaa luotaessa lisätä kentät:

```
self.__pari = False  
self.__tenttipisteet = []
```

## Lisäykset Opiskelija-olioon

```
def lisää_pari(self, pari):
    self.__pari = pari

def kerro_pari(self):
    if not self.__pari:
        return False
    else:
        return self.__pari.kerro_nimi()

def lisää_tentti(self, tehtavat):
    self.__tenttipisteet = tehtavat

def kerro_tenttipisteet(self):
    return self.__tenttipisteet
```



## Olio metodin parametrina

- ▶ Luokan metodeissa voidaan antaa toinen olio parametrina toiselle oliolle.
- ▶ Esim. jos pitää vertailla kahden opiskelijan tenttiarvosanaa, voitaisiin kutsua luokassa määriteltyä tämän kertovaa metodia tyyliin `kurssilainen1.onko_parempi_tenttiarvosana(kurssilainen2)`

## Parametriolion kenttien arvon selvittäminen

- ▶ Parametrina saadun olion kentät voidaan selvittää joko pistenotaation avulla, mutta tämä toimii vain saman luokan olioille:

```
def onko_parempi_tenttiarvosana(self, toinen):  
    return self.__tenttiarvosana > toinen.__tenttiarvosana
```

- ▶ Toinen vaihtoehto on käyttää luokan metodeita – tämä toimii aina:

```
def onko_parempi_harjoitusarvosana(self, toinen):  
    return #oikeasti alla oleva on samalla rivillä!  
           self.__tenttiarvosana > toinen.kerro_harjoitusarvosana()
```