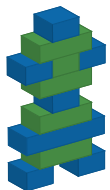


Johdatus kurssille ja ohjelmointiin

Ohjelmoinnin perusteet

1



Tervetuloa kurssille!

- Opettaja:
- Sähköposti:
- ViLLE-asioissa tukea saa myös ViLLE Teamilta:
villeteam@utu.fi

Kurssin rakenne

- n. 17 viikkoa, 2 x 45 minuuttia / viikko
- Kurssi koostuu
 - Oppitunneista
 - Tutoriaaleista
 - Harjoitustehtävistä
- Viimeisellä viikolla sähköinen tentti

Materiaalit

- Jaetaan ViLLEn kautta
- Luentokalvot
- Harjoitustehtävät
- Tutoriaalit
- Viikkotehtävät

VILLE

- Sähköinen, tehtäväpohjainen oppimisjärjestelmä
- Rekisteröitymislinkki:
ville.cs.utu.fi/ohjelmointi
- Kirjautumisavain:
ohj456

ViLLE (2)

- Viikkotehtävät (luennon jälkeen) tehdään yksin, tutoriaalit ja harjoitustehtävät pareittain
- Kaikilla tehtäväkierroksilla on aikaraja
- Aikarajan puitteissa tehtäviä saa palauttaa niin monta kertaa kuin haluaa

Mitä ohjelmointi on?

- Tietokoneen ohjaamista **ohjelmointikielen** avulla.
- Ohjelmointikielellä kirjoitetaan **ohjelmia**
- Ohjelman kirjoittaja = **ohjelmoija**.

Ohjelmointikielet

- Useita erilaisia, suosittuja tällä hetkellä esimerkiksi Java, Python, C#, C++, JavaScript jne.
- Matalan tason ja korkean tason kieliä
- Käännetyt ja tulkatut kielet

Python

- Kurssilla on käytössä **Python**-ohjelmointikieli
- Python on kohtuullisen helppo kieli oppia. Siitä huolimatta se on melko tehokas kieli.
- Kurssilla käytössä **versio 2.7**

Python (2)

- Python-tulkin voi ladata ilmaiseksi osoitteesta <https://www.python.org/downloads/>
- Tulkin mukana tulee yksinkertainen ohjelmointiympäristö IDLE. **Muista ladata versio 2!**
- Lisäksi Python-ohjelmia voi ajaa ViLLEssä suoraan

Miltä Python-ohjelma näyttää?

```
# Ohjelma kysyy käyttäjän nimen  
# ja tulostaa sen lopusta alkuun  
nimi = raw_input("Nimi:")  
nimi_rev = nimi[::-1]  
print "Väärinpäin:" + nimi_rev
```

Mikä on ohjelma?

- Ohjelma koostuu peräkkäisistä käskyistä eli lauseista
- Python määrittelee käytettävät käskyt
- Python-tulkki suorittaa ohjelman käsky kerrallaan alkaen ensimmäisestä

Peräkkäisyys

- Peräkkäisyys tarkoittaa, että ohjelma suoritetaan yksi käsky (eli **lause**) kerrallaan, alkaen ensimmäisestä
- Kun lause on suoritettu, siirrytään seuraavaan lauseeseen. Tätä jatketaan kunnes koko ohjelma on suoritettu.

Syntaksi ja semantiikka

- Ohjelmointikielessä määritellään
- **Syntaksi:** eli kielioppi, se miltä **käskyt näyttävät**
- **Semantiikka:** se, mitä **käskyt tarkoittavat**

Ensimmäinen käsky: print

- Ensimmäinen käsiteltävä käsky on **print**
- Sen avulla voidaan **tulostaa** tietoa ruudulle (konsoliin, ViLLEn tulostusikkunaan, Python IDLE:n ikkunaan jne.)

Syntaksi: print

```
print <tulostettava_tieto>
```


Syntaksi: print

```
print <tulostettava_tieto>
```

Käsky

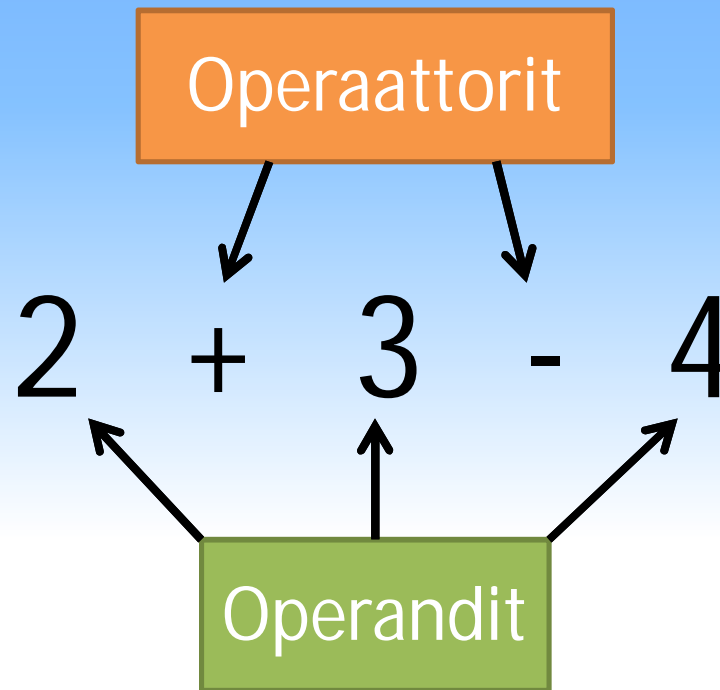
"Parametri"

Tulostus ruudulle

- Print-lauseella voidaan tulostaa minkä tahansa lausekkeen arvo ruudulle
- Lauseke voi olla yksittäinen arvo tai merkkijono, tai koostua useasta **operandista** ja **operaattorista**

Mitä ovat "operandit" ja "operaattorit"?

- Esimerkki:



Lausekkeen evaluointi

- Python laskee (eli **evaluoi**) print-lauseessa annetun lausekkeen, ja tulostaa ruudulle lopputuloksen
- Esimerkki:

```
print 15  
print 2 + 3  
print 2 * 4 + 10
```

Lausekkeen evaluointi (2)

- Python laskee lausekkeet operaattoreiden laskujärjestyksen mukaisesti
- Yleensä tämä laskujärjestys noudattaa matematiikasta tuttua kaavaa (eli ensin sulkeiden sisältä, kertolaskut ennen yhteenlaskua jne.)

Operaattorit

- Eräitä Pythonin tukemia matemaattisia operaattoreita:

Operaattori	Pythonissa	Matemaattinen vastine
Yhteenlasku	$x + y$	$x + y$
Vähennyslasku	$x - y$	$x - y$
Kertolasku	$x * y$	$x \cdot y$
Jakolasku	x / y	$x \div y$ tai $\frac{x}{y}$
Potenssi	$x ** y$	x^y

Esimerkkejä

```
print 15 + 8
```

```
print (4 + 2) / 3
```

```
print 2 ** 4 + 10 - (3 + 2)
```

Jakolaskusta

- Oletuksena Python laskee jakolaskun kokonaisluvulla pyöristämällä tuloksen alaspäin
- Esimerkiksi

```
print 5 / 2 # Tulostaa 2
```

```
print 11 / 3 # Tulostaa 3
```

```
print (2 + 4) / 5 # Tulostaa 1
```


Jakolaskusta (2)

- Matemaattisesti oikeampaa jakolaskua voidaan hyödyntää käyttämällä **desimaalilukua** joko jakajana tai jaettavana:

```
print 5.0 / 2 # Tulostaa 2.5
```

```
print 5 / 2.0 # Tulostaa 2.5
```

```
print 5.0 / 2.0 # Tulostaa 2.5
```

```
print 5 / 2 # Tulostaa 2
```

Desimaaliluvut

- Huomaa, että Pythonissa (samoin kuin useimmissa muissa ohjelmointikielissä) desimaalierottimena käytetään **pistettä** pilkun sijasta:

```
print 2.0 + 1 # Tulostaa 3.0
```

```
print 2,0 + 1 # Tulostaa mitä?
```

Merkkijonojen tulostaminen

- Mikäli halutaan tulostaa ruudulle tekstiä print-lauseella, täytyy tulostettava teksti ympäröidä lainausmerkeillä:

```
print "moi!" # Tulostaa moi!  
print "Tervepä terve."
```

Muuttujat

- Ohjelmointikielessä voidaan määritellä **muuttujia** tiedon tallentamiseksi.
- Muuttujilla on **nimi** (eli tunniste) ja **arvo**
- Muuttujan määrittelyn jälkeen sitä voidaan käyttää lausekkeiden osana samoin kuin numeroita tai merkkijonoja.

Miksi muuttuja?

- Muuttujaan **tallennetaan** ohjelmassa tarvittavaa tietoa.
- Muuttujan arvoa voidaan **käyttää** lausekkeissa.
- Muuttujaan tallennettu **arvo** voi nimensä mukaisesti **muuttua**.

Syntaksi: muuttujan määrittely

<tunniste> = <lauseke>

Muuttujan nimestä

- Muuttujan nimi saa sisältää kirjaimia, numeroita ja alaviivan (_). Nimen pitää alkaa kirjaimella.

Nämä kelpaavat	Nämä eivät kelpaa
<code>luku1</code>	<code>1luku</code>
<code>oma_nimi</code>	<code>oma nimi</code>
<code>eka_arvo</code>	<code>1. Arvo</code>
<code>prosenttiOsuus</code>	<code>%osuus</code>

Muuttujan nimestä (2)

- Huomaa, että pienet ja isot kirjaimet ovat merkityksellisiä Pythonissa
- Esimerkiksi muuttujat `eka` ja `Ǝka` ovat siis eri muuttujia.

Esimerkkejä muuttujien määrittelystä

```
a = 10
```

```
luku = 5 * 10 + 4
```

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

Muuttujien käyttö

- Kun muuttuja on määritelty, sen arvoa voidaan muuttaa ja sitä voidaan käyttää lausekkeiden osana:

```
a = 10
print a # Tulostaa 10
b = a + 3
print b # Mitä tulostaa?
a = a + 1
print a # Mitä tulostaa?
```

Muuttujien nimistä

- Muuttujat kannattaa nimetä niiden käyttötarkoituksen mukaan. Näin ohjelmaa on helpompi lukea ja ymmärtää.

Huono nimi	Parempi nimi
n	nimi
a1	arvo1
sn	sukunimi