



Esitietokysely -->



## Luento 2: Algoritmit, tilat ja maailmat

Ohjelmoinnillista ajattelua  
arjessa -luentosarja



Ohjelmoinin  
llinen ajattelu

```
object to mirror  
mirror_mod.mirror_object  
  
    operation == "MIRROR_X":  
        mirror_mod.use_x = True  
        mirror_mod.use_y = False  
        mirror_mod.use_z = False  
  
    operation == "MIRROR_Y":  
        mirror_mod.use_x = False  
        mirror_mod.use_y = True  
        mirror_mod.use_z = False  
  
    operation == "MIRROR_Z":  
        mirror_mod.use_x = False  
        mirror_mod.use_y = False  
        mirror_mod.use_z = True  
  
    selection at the end - add  
    mirror_ob.select= 1  
    mirror_ob.select=1  
    context.scene.objects.active  
    ("Selected" + str(modifier))  
    mirror_ob.select = 0  
    bpy.context.selected_objects  
    data.objects[one.name].se  
  
    print("please select exact  
-- OPERATOR CLASSES --
```

# Ohjelmointi

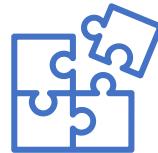
# PRADA-malli



Pattern Recognition  
(Hahmontunnistus)



Abstraction  
(Abstraktio)



Decoupling  
(Paloittelu)



Algorithms  
(Algoritmit)



# Monimutkaisuus

- Yksinkertainen
- Monimutkainen
- Helppo
- Vaikea



Algoritmit, tilat ja maailma (14.3.)

# Kertaus - mistä puhuttiin viimeksi?

- Ohjelmallinen ajattelu on ongelmanratkaisua, ohjelmojan laseilla katsottuna
- PRADA-malli
  - Hahmontunnistus (engl. Pattern Recognition)
  - Abstaktio (engl Abstraction)
  - Paloittelu (engl. Decoupling)
  - Algoritmit (engl. Algorithms)
- Monimutkaisuus
  - Yksinkertainen v. monimutkainen
  - Helppo v. vaikea
  - Tekeminen v. käyttäminen



# Käsitteitä

---

- Algoritmi
- Heuristiikka
- Tila
- Maailma (tai maailman tila)



# Tila

- Tila on kuin valokuva
  - Otettu hetkessä, pysäyttäen kyseisen hetken tuotokseksi (engl. Artifact)
  - Seuraava tila on erilainen, kuten seuraava valokuvakin
- Tila kuvailee yhtä tai useampaa asiaa
  - Yksittäinen tila
  - Monen tilan joukko, niin sanottu koostettu tila (engl. Composite State)
- Valokuvana tila voi
  - Kuvata yhtä tiettyä asiaa kuvassa
  - Kaikkea kuvassa - tällöin puhutaan mahdollisesti "maailman tilasta"



# Maailma ja maailman tila

---

- Ei tarkoita *koko* maailmaa
- Tarkoittaa ongelmanratkaisun kannalta *suurinta mielenkiintoista kokonaisuutta, joka sisältää kaikki ongelmanratkaisun kannalta mielenkiintoiset tilat*
- Maailma ja *ongelma-avaruus* ovat suurin piirtein synonyymejä



# Mikä on algoritmi?

---

- Algoritmi on "rajallinen joukko tarkkoja ohjeita" (wikipediaan mukaan)
- Algoritmilla on yleensä tavoite
  - Tavoite on yleensä (wikipediaan mukaan)
    - ratkaista tietty ongelma
    - Suorittaa joukko laskutoimituksia
- Todellisuudessa algoritmi yleensä tekee myös asioita, joita emme välttämättä halua sen tekevän, mutta joiden tekeminen ei oikeastaan kinnosta meitä.
  - Tällöin puhutaan "sivuvaikutuksista".
  - Yleensä vain tyydymme olemaan väliittämättä näistä sivuvaikutuksista, koska ne eivät ole oleellisia ongelman tai ratkaisun kannalta.



# Sivuhuomio: sivuvaikutuksista

- Tarkemmin sivuvaikutus tarkoittaa ohjelmoinnissa mitä tahansa maailman tilaa muokkaavaa, tai siitä riippuvaa tietoa tai muutosta ohjelman suorituksessa.
- Liittyy ajatukseen, että puhdas, siis sivuvaikutuksista puhdas, koodi tuottaa aina saman vastauksen, jokaisella kerralla kun sitä kutsutaan.
  - Sivuvaikutuksellinen koodi tuottaa eri vastauksen, koska se riippuu maailman tilasta, tai se muokkaa maailman tilaa.
- Sivuvaikutuksien ymmärtäminen on tärkeää virheiden metsästyksen kannalta.
  - Ilman sivuvaikutuksia oleva asia on aina sama
  - Sivuvaikutuksellinen tarkoittaa suoritukseen liittyvää muutosta
    - Siis, itsessään maailman tilan muokkaaminen on sivuvaikutus
- Käytännössä melkein aina ohjelmoinnissa halutaan tuottaa sivuvaikutuksia, koska ilman niitä sovelluksen tuotos olisi abstraktia.
  - Mutta sivuvaikutuksia halutaan tuottaa hallitusti.



# Algoritmi

---

- Koostuu:
  - Alkutila tai alkuarvot
  - Ohjeet (järjestyksessä)
  - Ehtoja
  - Toistoa
- Ohjeet saattavat sisältää alialgoritmeja
  - joita voidaan kuvata, tai olla kuvaamatta



---

## Alkutila tai alkuarvot

Jostain pitää lähteä

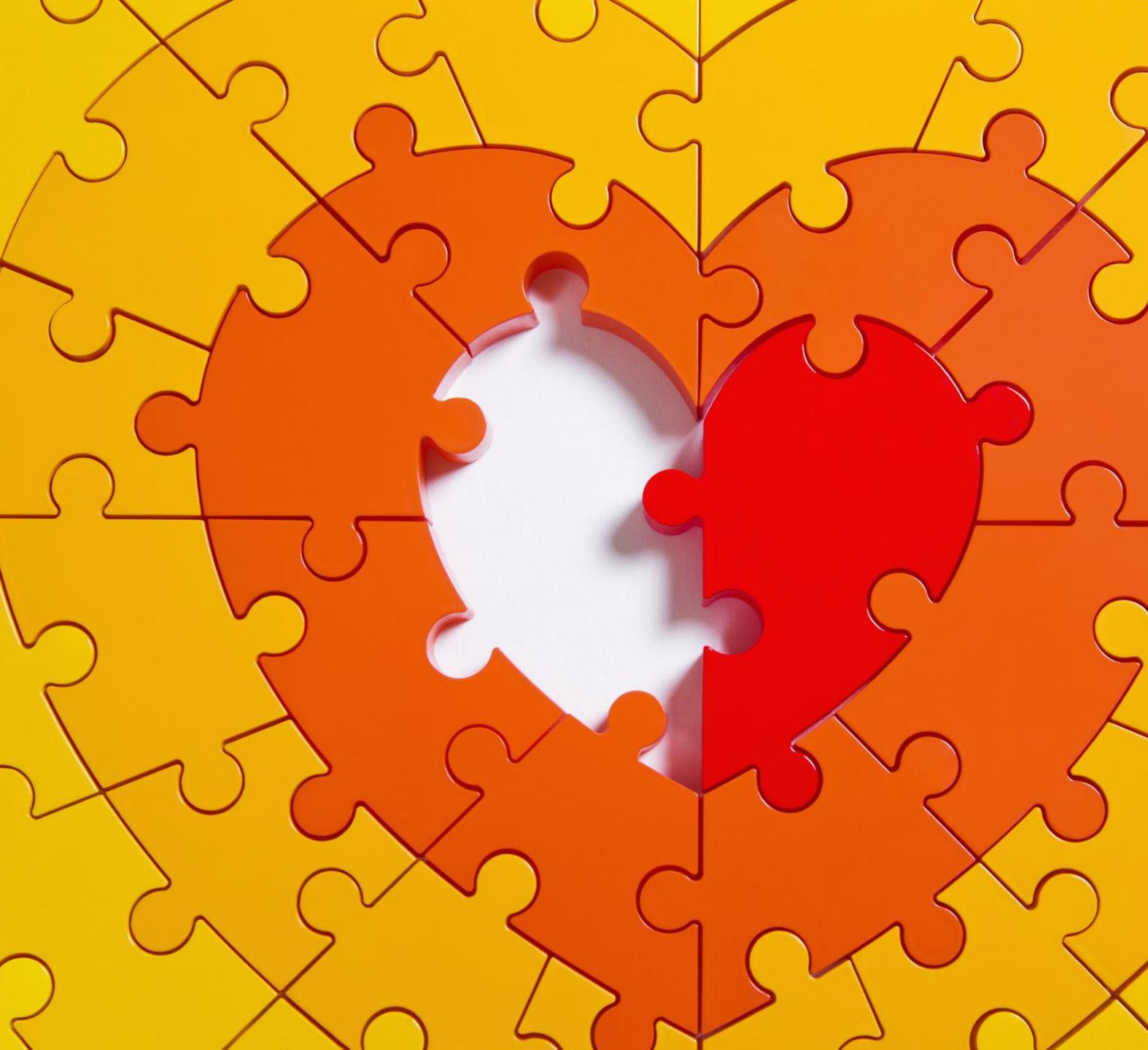




---

## Ohjeet

- Mitä tehdään
- Mitä tehdään sen jälkeen
- Miten tehdään



## — Alialgoritmit

- Asia voidaan määrittää halutulla tarkkuudella
- Alialgoritmit voivat sisältää uusia sisäkkäisiä alialgoritmeja
- Joskus tärkeää olla hyvinkin tarkka, ja kuvata kaikki yksityiskohdat ja vaiheet
- v. paloittelu, abstraktio

---

## Ehdot

- Mitä jos tila on
  - yhdenlaista
  - toisenlaista
- Mitä jos tila onkin eri kuin oletettiin?
- Mitä jos tila onkin juuri se mitä oletettiin?
- Ollaan valmiita reagoimaan erilaisiin tilanteisiin
- Ehdon täytymistä kuvataan yleensä totuusarvolla





## Totuusarvo

- Väittämä tai ehto:
  - On totta
  - Ei ole totta
- Juu tai ei
- 1 tai 0
- ehto täytyy yhdellä tai useammalla arvolla, tai se ei täyty yhdelläkään
- Ohjelmoinnin yhteydessä puhutaan *Boolean logiikasta*
- Lopulta kielellinen kysymys/sopimus, jonka voisi esittää myös satunnaisella jaottelulla "tomaatti vai kananmuna?" Tärkeätä siis kaksi eri tilaa, ei niiden nimet.

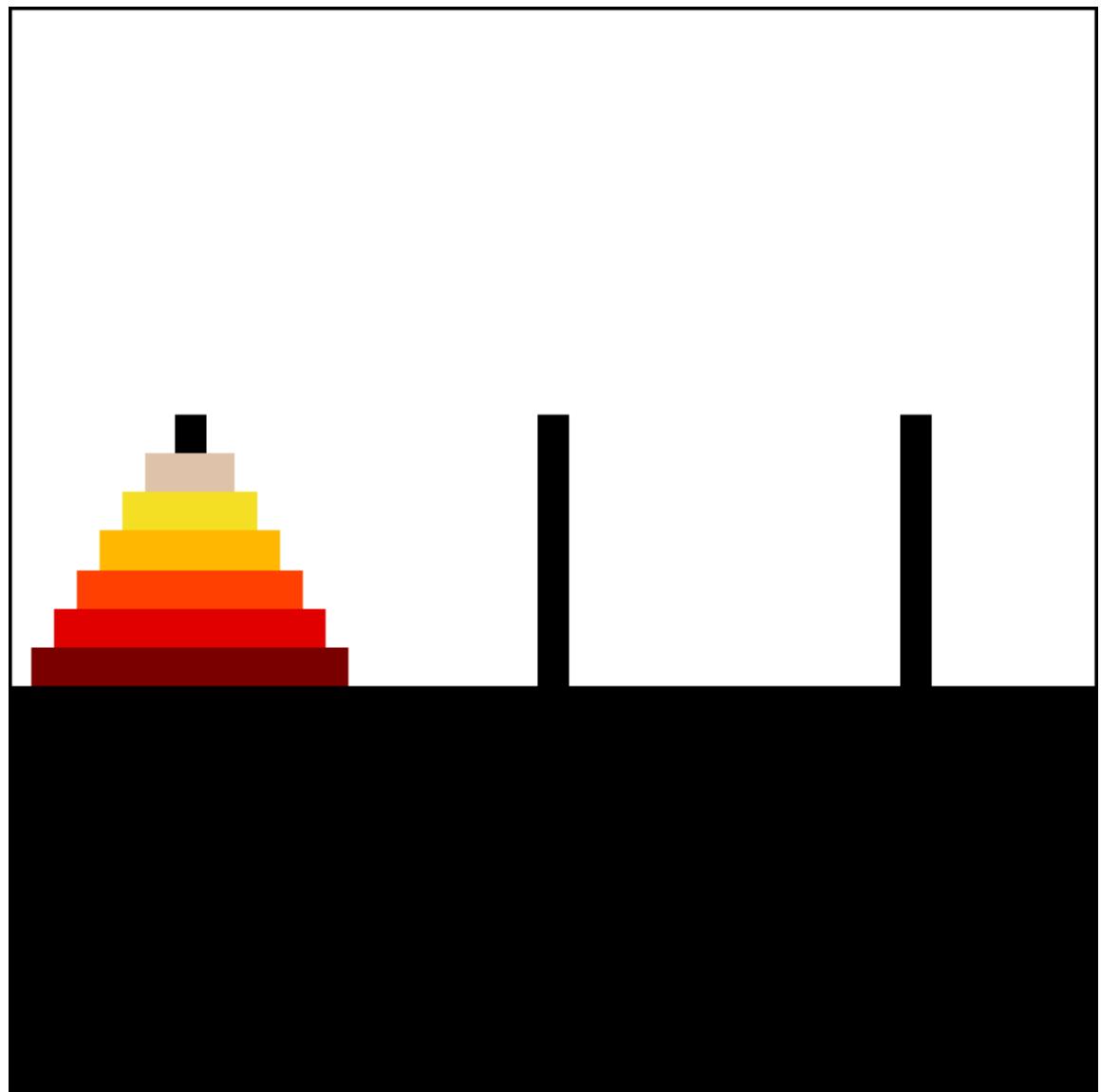


---

## Toisto

- Mahdollistaa monimutkaisten ohjeiden kirjoittamisen lyhyesti
- Mahdollistaa loputtoman toiston
- Yleensä eri toistokerroilla halutaan tehdä jotain asiaa eri tavalla kuin edellisellä toistokerralla

# Hanoin tornit



kuva:

[https://en.wikipedia.org/wiki/File:Iterative\\_algorithm\\_solving\\_a\\_6\\_disks\\_Tower\\_of\\_Hano](https://en.wikipedia.org/wiki/File:Iterative_algorithm_solving_a_6_disks_Tower_of_Hano)

# Mikä on heuristiikka?

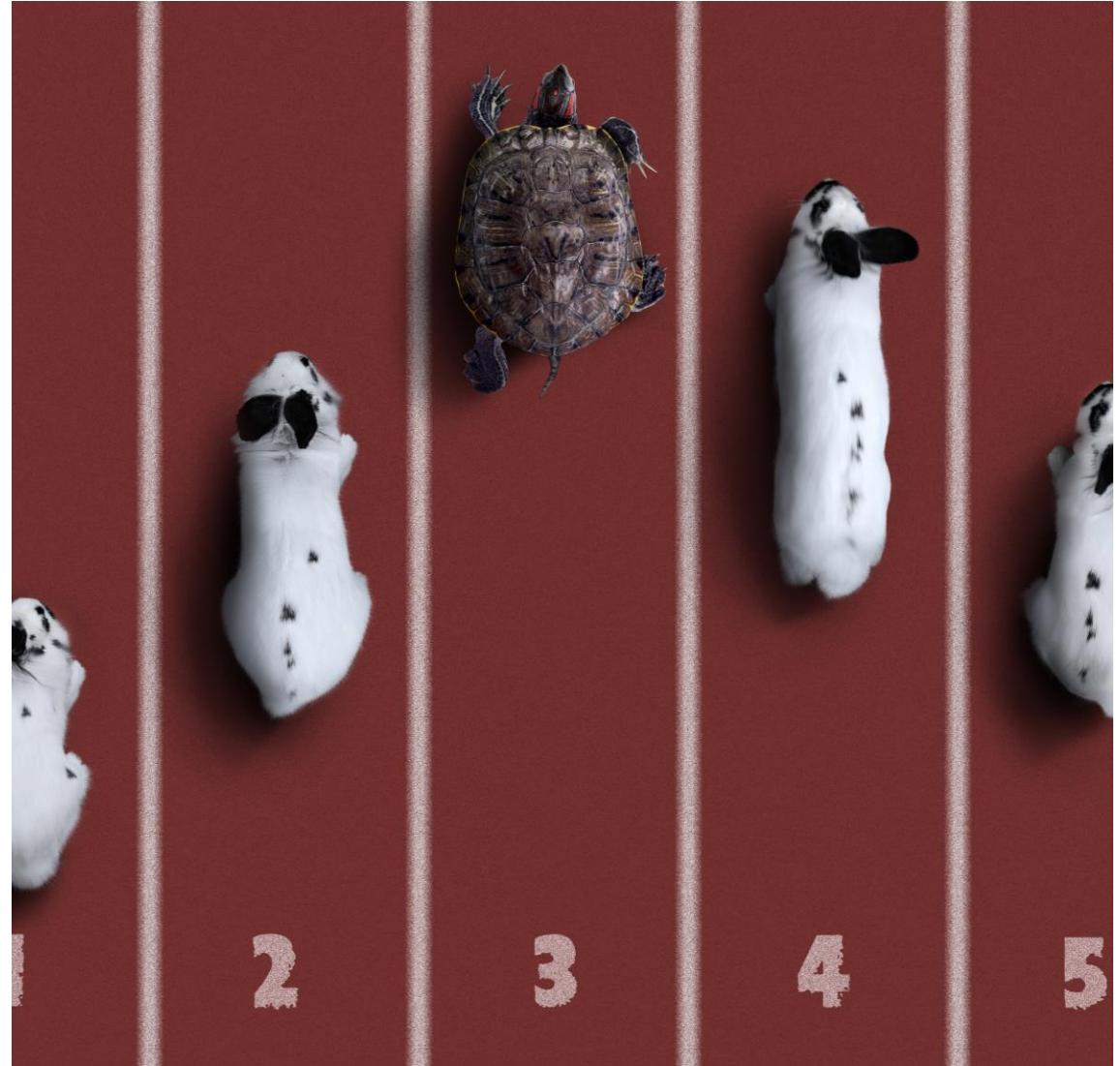
---

- Heuristiikka on
  - Tilapäinen, tai avustava tapa lähestyä ratkaisun löytämistä
  - Se on yleensä epätarkka, optimoitamaton, tai muutoin puutteellinen tapa lähestyä ratkaisua.
  - yleensä kuitenkin riittävä ohjeistus maaliin pääsemiseksi,
    - vaikkakaan ei yleensä kovin tarkka, kaunis tai edes riittävä määritelmänsä puolesta.
- "Ollaan menossa oikeaan suuntaan"



# Algoritmi vai heuristiikka?

- Algoritmi on tarkka kuvaus ratkaisusta, kun taas heuristiikka on pikemminkin arvaus mahdollisista ratkaisun pääkohdista.
- Algoritmi on lopputuote, kun taas heuristiikka on lähempänä lähtökohtaa, josta voidaan lähteä algoritmia rakentamaan.
- Algoritmi on esimerkiksi valmis resepti, kun taas heuristiikka on arvaus, että soppa saadaan, kunhan vain muistetaan lisätä perunaa.
- Algoritmin suorittaminen tuottaa lopputuloksen (joka voi olla haluttu), kun taas heuristiikka voi olla oikean (tai väärän) suuntainen.





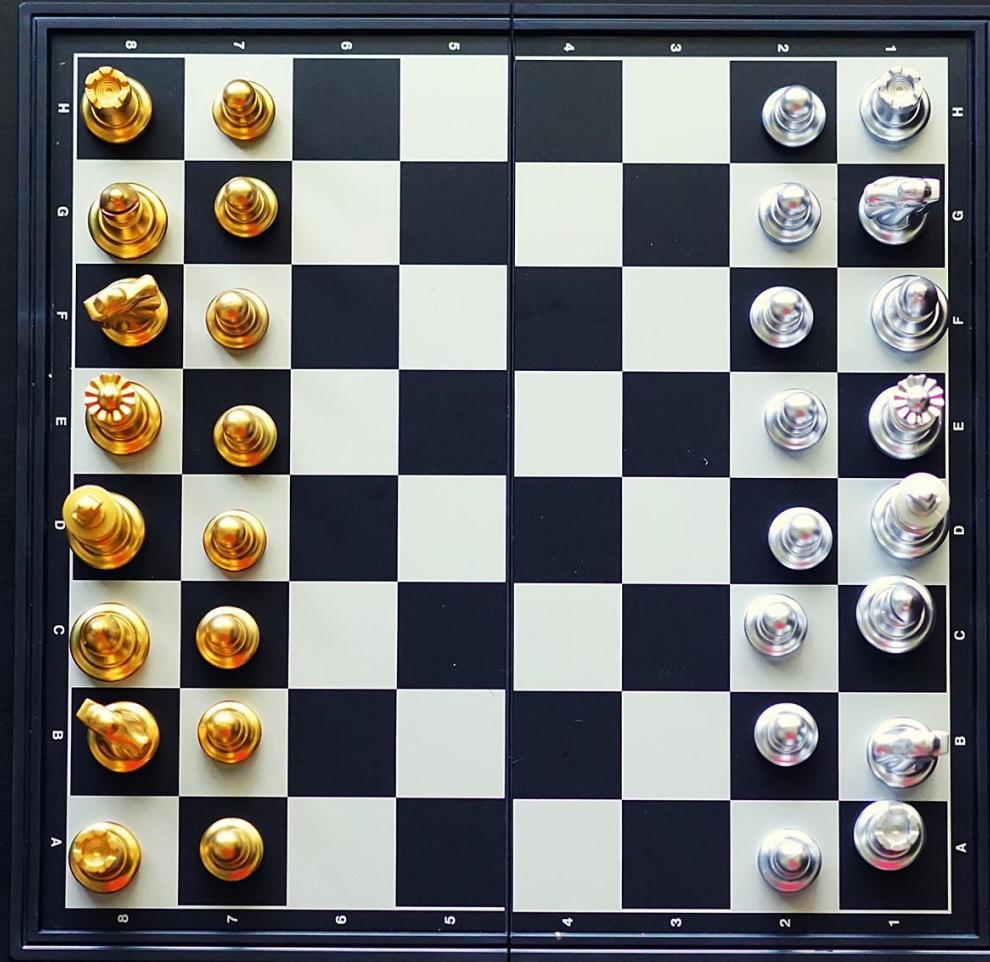
Esimerkkejä

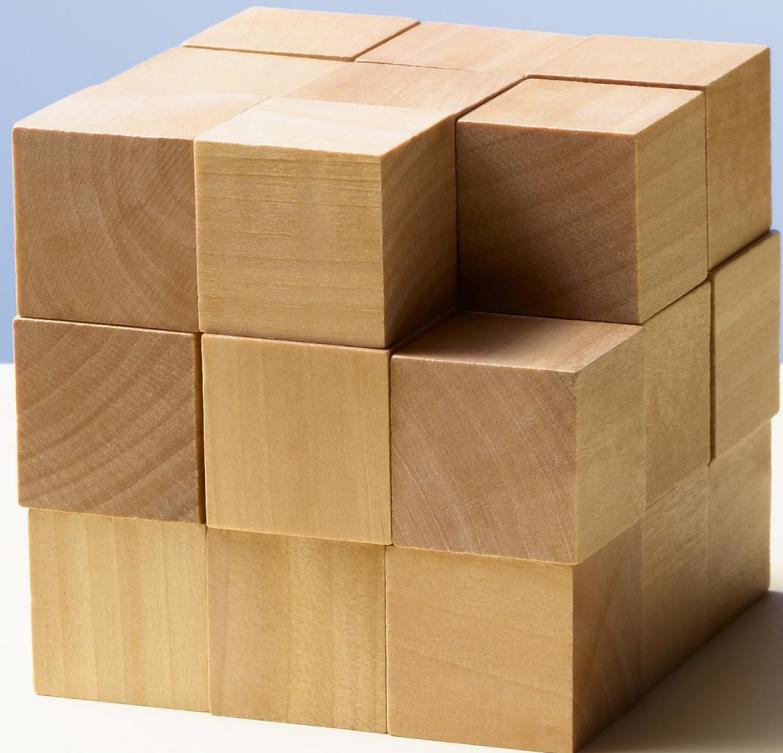


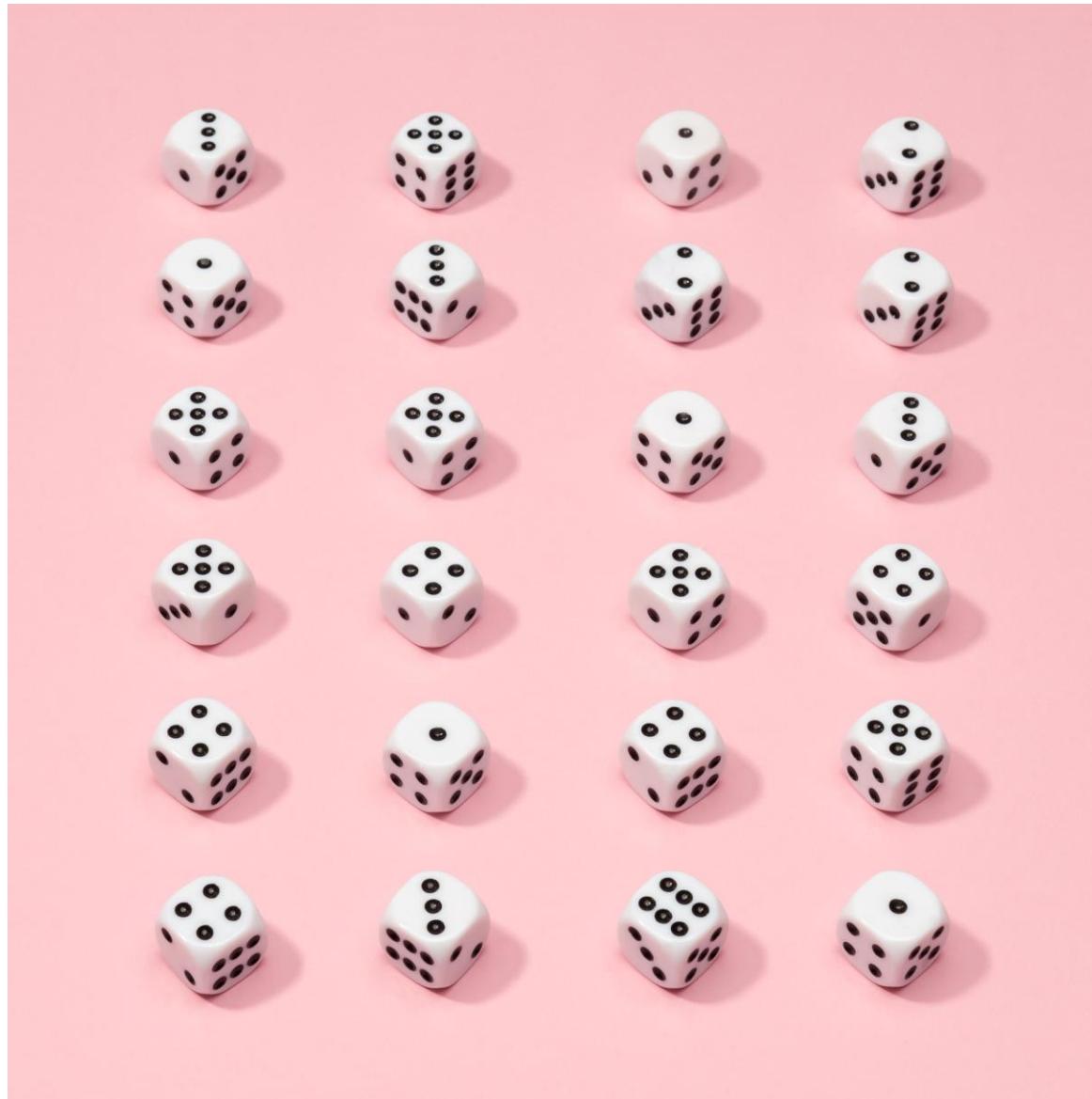
# Sivuhuomiona: Sokraattinen väittely

- Väite + perustelu (- vastaväite)
- Väite on
  - jotaan, jolla on totuusarvo
  - yleensä lyhyt
  - Esim. "*Shakissa siirto voidaan esittää algoritmina*"
- Perustelu on
  - jotaan, joka tarjoaa pohjan väitteelle
  - Yleensä väitettä pidempi, mutta voi olla ytimekkäänä lyhyt
  - Esim. ", *koska se voidaan määrittää ehdollisina ja järjestettyinä ohjeina.*"
- Vastaväite (+ perustelu)
  - Tarkoituksena kumota aiempi väite
  - Esim. "*siirto olisi parempi esittää alialgoritmina, (koska se on vain osa pelin algoritmia)*"
    - Vastaväite voi myös epäonnistua, niin kuin väitekin. Silloin siihen voi kohdistaa toisen vastavätteen.
    - Esim. "*Siirtoa ei tarvitse esittää alialgoritmina, koska sitä voidaan haluta käsitellä erillään itse pelistä.*"

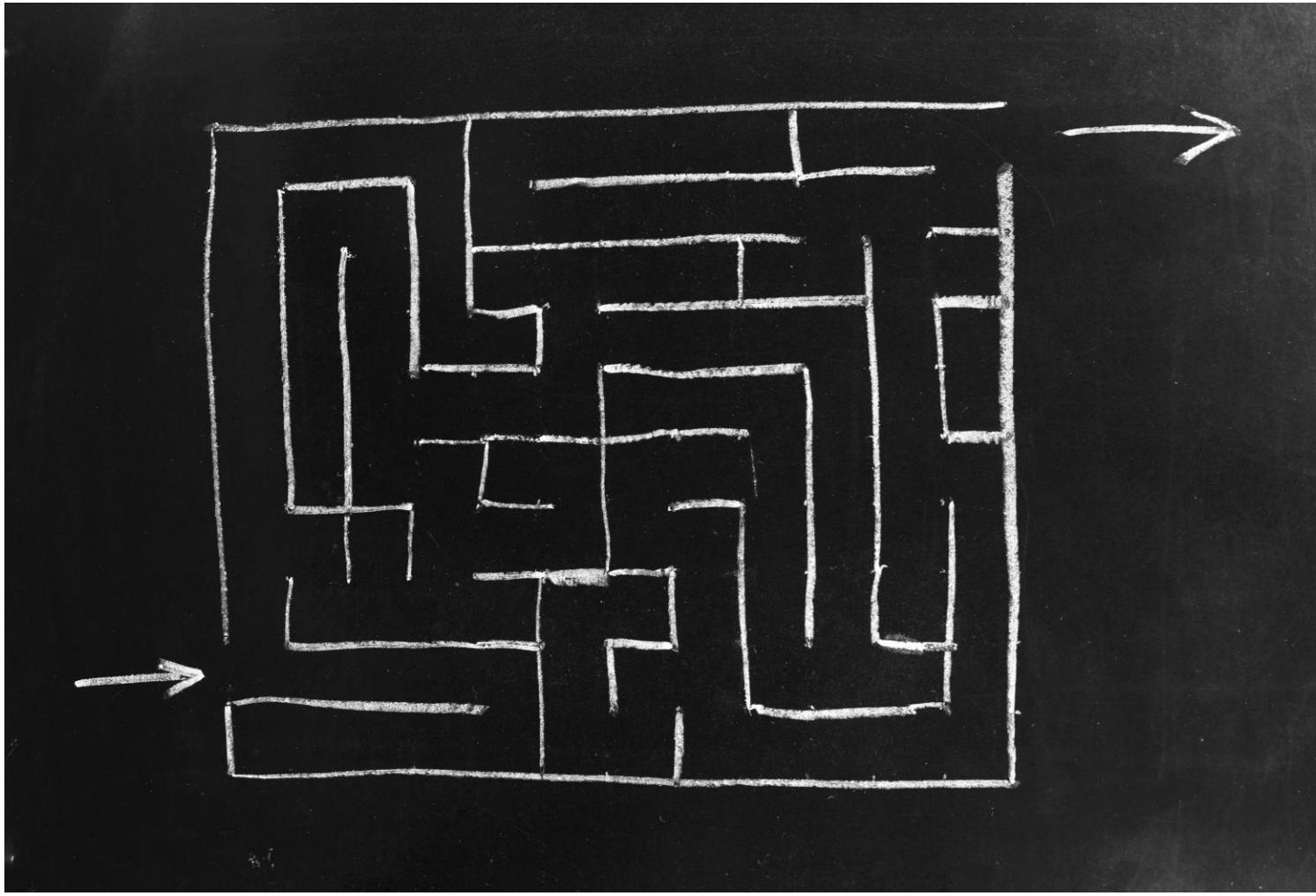


















---

Asioiden nimeämisestä (21.3.)

---



Hahmontunnistus arkisen(kin) ongelmanratkaisun mahdollistajana (4.4.)

A photograph showing a large, chaotic pile of numerous cardboard boxes of various sizes and orientations. The boxes are made of brown corrugated material and are stacked haphazardly, creating a complex, geometric pattern of lines and shapes.

Abstraktioista (11.4.)



Monimutkaisuudesta (18.4.)

# Kuvien alkuperä

- Kuvat, jos ei erikseen mainittu:
  - Powerpointin mukana tulevat "stock photos"-kuvat
  - Unsplash
- Muissa kuvissa merkinnät