

# Ohjelmoinnin peruskurssi Y1

CS-A1111

# Poikkeusten käsittely

## Oppimistavoitteet: tämän videon jälkeen

- ▶ Osaat käsitellä ohjelmassa erilaisia virhetilanteita niin, että ohjelma ei kaadu esimerkiksi silloin, jos käyttäjä antaa pyydetyn luvun sijasta tekstiä.

# Poikkeukset

- ▶ Ohjelmaa suoritettaessa voidaan törmätä virhetilanteisiin.
- ▶ Osa virheistä johtuu ohjelmointivirheistä, mutta osaan ohjelmoija ei voi vaikuttaa (esim. väärän tyyppinen syöte).
- ▶ Virhetilanteiden käsittely if-else-rakenteen avulla tekee ohjelmasta helposti sekavan.
- ▶ Python tarjoaa virhetilanteiden käsittelyyn oman mekanismin, *poikkeukset*.
- ▶ Poikkeus voidaan käsitellä try-except-rakenteen avulla.

## try-except-rakenne

```
try:  
    # Jono kaskyja, joista jokin tai jotkin  
    # voivat aiheuttaa poikkeuksen.  
except poikkeuksen_tyyppi:  
    # Kaskyja, jotka jotenkin selvittavat  
    # virhetilanteen, jos on aiheutunut  
    # poikkeuksen_tyyppi-tyyppinen poikkeus.
```

## Virheelliseen syötteeseen varautuminen

- ▶ Ohjelma yrittää muuttaa käyttäjän syöteen luvuksi, mutta syöte on väärää tyyppiä  $\Rightarrow$  aiheutuu `ValueError`-tyyppinen poikkeus.
- ▶ Vaihtoehtoja poikkeuksen käsittelemiseksi:
  - ▶ Käyttäjälle annetaan `except`-osassa selväsanainen virheilmoitus.
  - ▶ Käyttäjältä pyydetään uutta syötettä niin kauan, että hän antaa luvun.

## Esimerkki: naulamuunnos

```
def main():
    NAULAKERROIN = 0.4536
    print("Muutan nauloina annetun massan kilogrammoiksi.")
    try:
        syote = input("Anna massa nauloina: ")
        naulat = int(syote)
        kilot = NAULAKERROIN * naulat
        print("Massa on {:.3f} kg".format(kilot))
    except ValueError:
        print("Virhe: et antanut nauvoja kokonaislukuna.")

main()
```

## Syötteen pyytäminen uudelleen

- ▶ Parempi versio pyytää käyttäjältä nauvoja niin kauan, että hän antaa kokonaisluvun.
- ▶ Uutta pyyntöä ei sijoiteta except-osaan, vaan koko try-except-osa sijoitetaan toistokäskyn sisään.
- ▶ Toistokäskyn suoritusta jatketaan niin kauan, että on saatu luettua kelvollinen syöte.



## Naulamuunnos: uusi koodi

```
def main():
    NAULAKERROIN = 0.4536
    print("Muutan nauloina annetun massa kilogrammoiksi.")
    luku_onnistui = False
    while not luku_onnistui:
        try:
            syote = input("Anna massa nauloina: ")
            naulat = int(syote)
            kilot = NAULAKERROIN * naulat
            print("Massa on {:.3f} kg".format(kilot))
            luku_onnistui = True
        except ValueError:
            print("Virhe: naulat pitää olla kokonaisluku.")
            print("Yrita uudelleen!")

main()
```

## Huomatuksia poikkeuksista

- ▶ try-except-rakenne voi sisältää useita except-osia erityyppisiä poikkeuksia varten. Tällöin poikkeuksen sattuessa siirrytään ensimmäiseen except-osaan, jonka poikkeuksen tyyppi vastaa aiheutunutta poikkeusta.
- ▶ Poikkeuksien tyypit löytyvät pythonin dokumentaatiosta:  
<https://docs.python.org/3/library/exceptions.html>
- ▶ Ohjelmoija voi myös itse aiheuttaa poikkeuksen (virhetilanteen sattuessa) raise-käskyllä (ei käsitellä tällä kurssilla).
- ▶ Ohjelmoija voi määritellä itse poikkeuksia (ei käsitellä tällä kurssilla).