

# Pong-peli Pythonilla

Tämän ohjeen avulla rakennat Pong-pelin Pythonilla käyttäen turtle-kirjastoa. Peli tehdään kolmessa vaiheessa, ja jokaisen vaiheen askeleissa peli rakentuu pala palalta. Etene askel kerrallaan ja kokeile pelin toimivuutta aina askeleen lopussa ennen kuin jatkat seuraavaan.

Käytä IDLE-ympäristöä ja tee tälle pelille oma tiedosto (esim. pong.py). Tallenna tiedosto OneDriveen.

**Vinkki:** Tallenna ohjelma usein *Ctrl+S*-näppäinyhdistelmällä. *OneDrive synkronoi tiedoston pilveen, jolloin työ ei jää vain koulun tietokoneelle.*

## Vaihe 1: Peli-ikkuna ja mailat

Aloitetaan pelin rungosta: ikkuna ja kaksi mailaa, jotka liikkuvat näppäimillä.

### Askel 1: Peli-ikkuna

Tuo turtle-kirjasto ja luo peli-ikkuna. `wn.tracer(0)` sammuttaa Pythonin automaattisen piirtoanimaation. Ikkunan päivittäminen toteutetaan myöhemmässä vaiheessa `wn.update()`-komennolla, jolloin peli näyttää sulavalta eikä nykivältä.

```
import turtle

wn = turtle.Screen()
wn.title("Pong")
wn.bgcolor("black")
wn.setup(width=800, height=600)
wn.tracer(0)
```

### Askel 2: Pelaajan A maila

Luodaan ensimmäinen maila. `turtle.Turtle()` luo uuden olion. `penup()` nostaa kynän ylös, jotta olio ei piirrä viivaa liikkeessaan, ja `shapsize` venyttää valmiin neliön mailan muotoiseksi. Lopuksi pelaajan A maila siirretään ikkunan vasempaan reunaan.

```
maila_a = turtle.Turtle()
maila_a.speed(0)
maila_a.shape("square")
maila_a.color("white")
maila_a.shapesize(stretch_wid=5, stretch_len=1)
maila_a.penup()
maila_a.goto(-350, 0)
```

### Askel 3: Pelaajan B maila

Toisen pelaajan maila: nyt nimeksi maila\_b ja siirretään se ikkunan oikeaan reunaan.

```
maila_b = turtle.Turtle()
maila_b.speed(0)
maila_b.shape("square")
maila_b.color("white")
maila_b.shapesize(stretch_wid=5, stretch_len=1)
maila_b.penup()
maila_b.goto(350, 0)
```

### Askel 4: Pallo (ei vielä liiku)

Luodaan seuraavaksi pallo-olio. Tehdään pallostä aluksi valkoinen neliö.

```
pallo = turtle.Turtle()
pallo.speed(0)
pallo.shape("square")
pallo.color("white")
pallo.penup()
pallo.goto(0, 0)
```

### Askel 5: Mailojen liikutusfunktiot

Luodaan funktiot mailojen pystysuuntaista liikuttamista varten. Komento ycor() palauttaa olion y-koordinaatin ja sety() asettaa uuden. Näin mailaa liikutetaan pystysuunnassa 20 pikseliä kerrallaan.

```
def maila_a_up():
    y = maila_a.ycor()
    y += 20
    maila_a.sety(y)

def maila_a_down():
    y = maila_a.ycor()
    y -= 20
    maila_a.sety(y)

def maila_b_up():
    y = maila_b.ycor()
    y += 20
    maila_b.sety(y)

def maila_b_down():
    y = maila_b.ycor()
    y -= 20
    maila_b.sety(y)
```

## Askel 6: Näppäinohjaus

Komento `wn.listen()` kääsee ikkunaan kuuntelemaan näppäinpainalluksia. Komento `onkeypress(funktio, näppäin)` yhdistää näppäimen ja funktion, ja tätä kutsutaan tapahtumaohjaukseksi: funktiota kutsutaan aina kun näppäintä painetaan.

```
wn.listen()
wn.onkeypress(maila_a_up, "w")
wn.onkeypress(maila_a_down, "s")
wn.onkeypress(maila_b_up, "Up")
wn.onkeypress(maila_b_down, "Down")
```

## Askel 7: Pääsilmutka

Peli tarvitsee jatkuvasti pyörivän silmukan, joka päivittää ikkunan. Ilman sitä ohjelma sulkeutuisi heti.

```
while True:
    wn.update()
```

**Kokeile:** Aja koodi. Näkyviin pitäisi tulla musta ikkuna, jossa on kaksi valkoista mailaa reunoilla ja pallo keskellä. Näppäimet W/S liikuttavat vasenta mailaa, ja nuolet ↑/↓ oikeaa.

**Kustomoi oma pelisi:** Nyt kun perusta on pystyssä, voit muokata peliä omannäköiseksi:

- Taustaväri: `wn.bgcolor("darkblue")`, kokeile eri värien nimiä (esim. "purple", "hotpink") tai hex-koodia (esim. "#1a1a2e")
- Mailan väri: `maila_a.color("red")`, mailojen ei tarvitse olla samanväriset
- Pallon muoto: `shape("triangle")`, `shape("circle")` tai `shape("turtle")` toimivat "square"-muodon sijaan
- Koko: `shapesize()`-arvoja suurentamalla tai pienentämällä, esim. `pallo.shapesize(2)` saa isomman pallon