

# □ JS: Partikkel-gravitasjon



PÅ BOKMÅL



LAST NED PDF

## Introduksjon

I denne oppgåva tek me utgangspunkt i animasjonen me laga i oppgåva [Partikkel-animasjon](#) . Viss du ikkje har gjort den oppgåva anbefalar me at du gjer den før du fortset her.

Denne oppgåva går ut på å leggje til gravitasjon på `partikkel`-objektet. Du får presentert kva du skal gjere, men du må sjølv finne ut korleis du skriv kode for å få det til. Til slutt kjem det til å sjå ut omlag som dette:



## Steg 1: Få partikkelen til å falle

Det fyrste me skal gjere er å få partikkelen til å falle, og for å få til det må du programmere det følgjande:

- ☐ Legg til to attributtar i `particle.gravitasjon` som styrer akselerasjonen nedover, og `hastigheit` som er farta den fell med. Du bør velje eit lite tal for `gravitasjon`, til dømes `0.01`. Frå starten bør `hastigheit` vere `0`.
- ☐ Set ein passende startstad for `particle`, helst høgt oppe.

For kvar gong `draw()` køyrer, skal følgjande skje:

- Attributten `gravitasjon` skal endrast med `hastigheit`-attributten.
- `y`-posisjonen til `particle` blir endra med `hastigheit`-attributten.

HINT

Fell partikkelen no? Når partikkelen fell ser du kanskje òg at den fell gjennom skjermen. Difor må me leggje til ein sjekk som gjer at den stoppar nedst.

## Steg 2: Stopp nedst på skjermen

For at partikkelen vår skal stoppe nedst på skjermen treng me enkelt og greit ein sjekk kvar gong me køyrer `draw()`

- ☐ Lag ein sjekk som kontrollerer om `particle` sin `y`-verdi er større enn høgda på `canvas`-objektet. OBS! Du vil sikkert merke at den "ristar" heilt nedst, difor må du òg ta omsyn til storleiken på partikkel-objektet når du sjekkar kor langt ned den er.

### HINT

- ☐ I sjekken må du stoppe partikkelen ved å setje `y`-verdien. Kan du tenke deg kva `y`-verdien bør vere?

No har du eit fallande objekt!

## Utfordringar

- ☐ Når partikkelen treff kanten, snu gravitasjonen slik at partikkelen går oppover.
- ☐ Klarar du å få partikkelen til å gå til venstre eller høgre òg?
- ☐ "Steng" alle kantane i `canvas` ved hjelp av fleire `if`-setningar.

## Steg 3: Få partikkelen til å flyge!

No som me har fått gravitasjon på partikkelen vår kan me leggje til styring på partikkelen ved hjelp av tastar. Me skal programmere slik at når du trykkar `pil opp` så gir du partikkelen negativ akselerasjon, og når me slepp att blir den påverka av gravitasjonen ned att. For å gjere det brukar me noko som heiter `onkeyup`, `onkeydown` og `keyCode`.

For å kunne utføre ein operasjon når me trykkar på ein tast må me ha denne koden:

```
window.onkeydown = function(inputKey) {
  var key = inputKey.keyCode ? inputKey.keyCode : inputKey.which;

  if (key == TAST) {
    // Kode som skal utførast når TAST blir trykka.
  }
}
```

- ☐ Bruk [keycode.info](https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/keyCode) til å finne ut kva `tastekode` tasten du vil bruke har.

## Forklaring

- `window.onkeydown` er ein funksjon som sjekkar om ein tast blir trykka ned, og viss den gjer det blir funksjonen køyrt med `tastekode` lik `inputKey`.
- `var key = inputKey.keyCode ? inputKey.keyCode : inputKey.which;` er ein enklare måte å skrive dette på:

```
if(inputKey.keyCode) {  
    var key = inputKey.keyCode;  
} else {  
    var key = inputKey.which;  
}
```

Grunnen til at du må bruke `keyCode` og `which` er fordi ikkje alle nettlesarar støttar `keyCode`, men dei brukar `which` i staden.

- I `if (key == TAST)` må du skrive kode for at gravitasjonen skal vere negativ. Prøv med ulike tal.
- ☐ For at du skal få att gravitasjonen når du slepp tasten må du lage ein funksjon for `window.onkeyup`.
- ☐ Les meir om `window.onkeyup` og meir her:

[w3school](https://www.w3schools.com/jsref/jsref_onkeyup.asp)

Viss du står heilt bom fast kan du sjå her: [Forslag til kode](#).

Lisens: CC BY-SA 4.0