

OOP Inheritance

1 Inheritance (Overerving)

? Leerdoelen

- Je weet wat inheritance (overerving) betekent in OOP.
- Je kunt een class maken die eigenschappen en methodes **erft** van een andere class.
- Je ziet waarom inheritance handig is: minder herhaling, meer overzicht.

? Uitleg

Wat is inheritance?

Inheritance betekent dat je een nieuwe class maakt die eigenschappen en methodes **overneemt** van een andere class. Dit heet overerving. De 'ouderclass' noem je ook wel de *superclass*, en de 'kindclass' de *subclass*.

Voorbeeld

```
<?php
class Dier {
    public $naam;

    public function adem() {
        echo $this->naam . " ademt.\n";
    }
}

class Hond extends Dier {
    public function blaf() {
        echo $this->naam . " zegt: Woef!\n";
    }
}
```

```
$rex = new Hond();  
$rex->naam = "Rex";  
$rex->adem(); // komt uit Dier  
$rex->blaf(); // komt uit Hond  
?>
```

□ De class `Hond` **erft** de methode `adem()` van `Dier`, maar voegt ook zijn eigen gedrag toe: `blaf()`

?? Opdracht – Maak je eigen dier met overerving

1. Bestand: dier.php

Maak een class `Dier` met:

- een property `$naam`
- een methode `beweeg()` die `"{$this->naam} beweegt."` toont

2. Bestand: vogel.php

Maak een class `Vogel` die `extends Dier`:

- een methode `vlieg()` die `"{$this->naam} vliegt!"` toont

3. Bestand: test.php

- Maak een object `$mus` van de class `Vogel`
- Stel zijn naam in op `"Mus"`
- Roep zowel `beweeg()` als `vlieg()` aan

? Reflectie

- Wat gebeurt er als je `vlieg()` oproept op een `Dier`-object?
- Kun je uitleggen waarom `Vogel` de methode `beweeg()` kan gebruiken zonder die zelf te schrijven?

? Inleveren

- `dier.php`, `vogel.php`, `test.php`
- Reflectie in `reflectie-les7-<jouw-naam>.txt`

Revision #1

Created 2025-06-29 19:09:40 UTC by Max

Updated 2025-06-29 19:09:59 UTC by Max