



# Object Oriented Programming

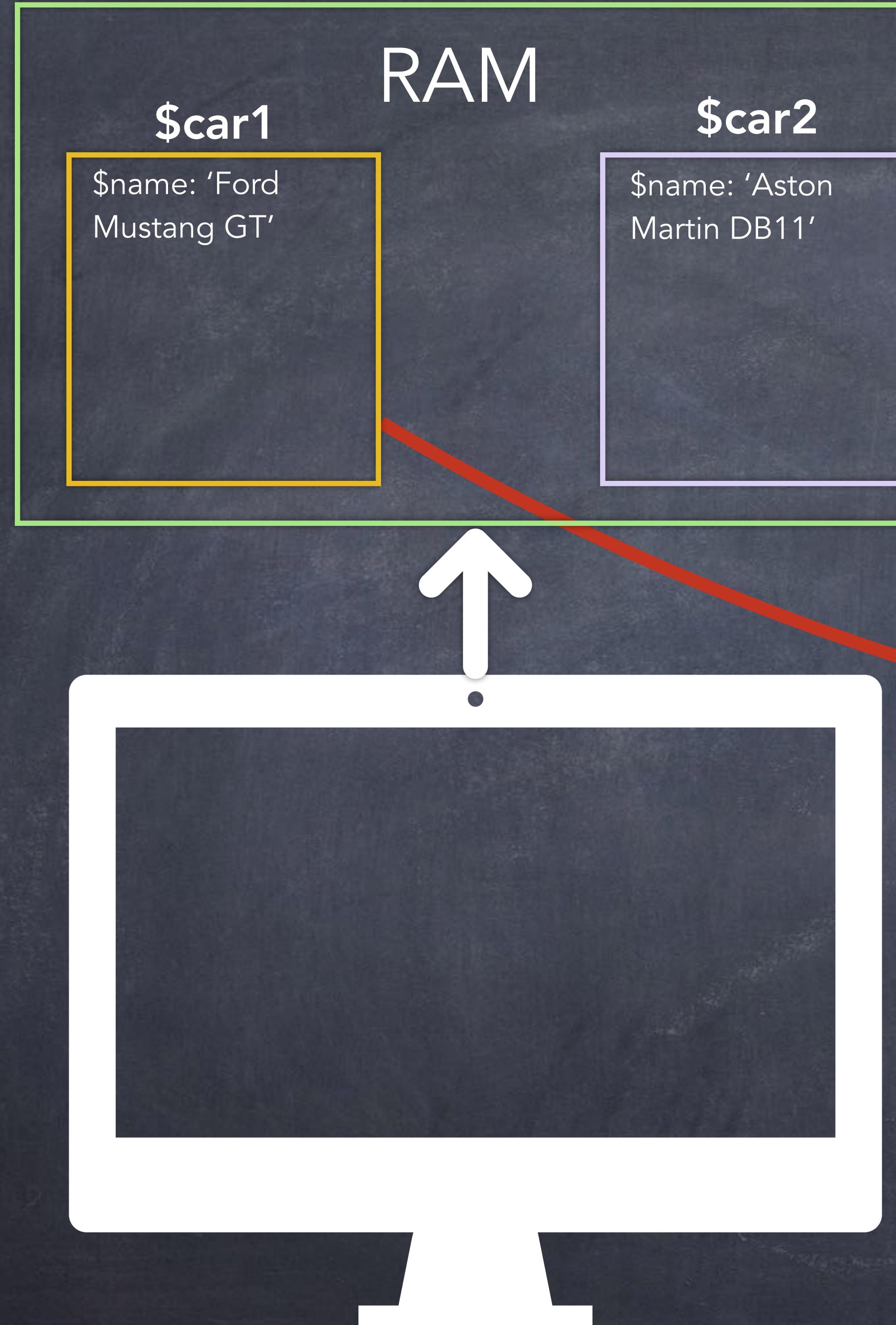
OOP Les 2

# Lesdoelen vandaag

- Je begrijpt de lesstof van les 1 beter
- Je leert al iets meer visueel te denken en minder in code
- Je begrijpt wat `$this` is en doet

# Les 1 (doelen van voorgaande les)

- ➊ Je weet in basis wat Object Oriented Programming (OOP) is
- ➋ Je weet wat classes en objecten zijn
- ➌ Je kunt zelf een simpele class schrijven



```
<?php  
require_once('Car.php');  
  
$start = 0;  
$finish = 25;  
  
$car1 = new Car();  
$car1->setName('Ford Mustang GT');  
$car1->driveTo($finish);  
  
$car2 = new Car();  
$car2->setName('Aston Martin DB11');  
$car2->driveTo($finish);
```

\$car1

RAM

class: Car  
\$name: 'Ford Mustang GT'

\$car2

class: Car  
\$name: 'Aston Martin DB11'

```
<?php  
class Car  
{
```

```
    protected $name;  
    protected $position;
```

```
    public function setName($new_name)  
{  
        $this->name = $new_name;  
    }
```

```
    public function driveTo($finish)  
{  
        while ($this->position < $finish) {  
            $this->position++;  
            echo $this->name . ' is at position ' . $this->position . "\n";  
        }  
    }  
}
```

**\$this** in de Class is een 'magische' variabele die verwijst naar het **huidige object**

Met andere woorden:

Zeg ik in de class Car (die **generiek** voor alle auto's is) in de functie setName(\$new\_name) de code:

`$this->name = $new_name;`

Zorgt dat ervoor dat bij:

`$car1->setName('Ford Mustang GT')` de naam van \$car1 wordt aangepast, en niet van \$car2.



Lees het nog eens na: slides\_jochem/oop\_les\_2.pdf

# ClassName

## Eigenschappen / Properties

eigenschap1 (\$name, \$color, ...)

eigenschap2 (\$speed, \$size, ...)

## Functionaliteit / Gedrag / Functies

setName(\$new\_name)

setSpeed(\$new\_speed)

moveTo(\$target)

getPosition()









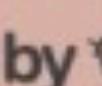


designed by  freepik.com

## STUDENT CHARACTER

---



designed by  freepik.com



