

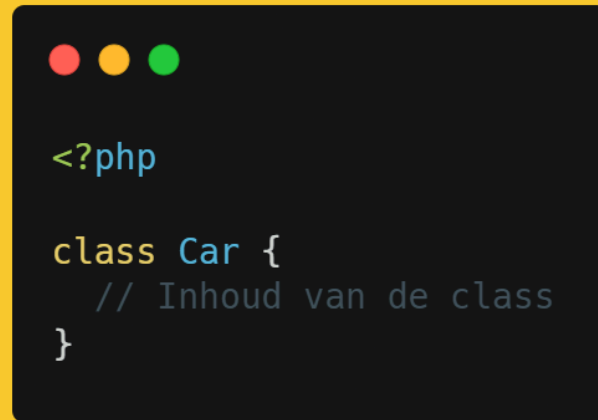
# Versnellers Blok C PHP Les 1

# Leerdoelen

- PHP Classes
- Properties in Classes
- PHP functions
- Access Modifiers
- Wat is \$this?

# Wat is een Class in PHP?

- Blauwdruk van een Object
- Structuur
- Herbruikbaar
- Uitbreidbaar
- Logisch



```
<?php

class Car {
    // Inhoud van de class
}
```

# Wat zit er allemaal in een PHP Class?

- Class name (PascalCase)
- Properties (camelCase)
- Functions (camelCase)

# Wat is een Property?

Een Property is een waarde/variabele die opgeslagen wordt in een Class.

Een Property is ALTIJD in camelCase!

```
<?php  
  
class Car {  
    public $brand;  
    public $type;  
    public $color;  
    public $transmissionType;  
    // etc.  
}
```

✓ Dus:

- ✓ \$voornaam, \$aantalStudenten, \$maxSnelheid
- ✗ \$VoorNaam, \$aantal\_studenten, \$Max\_Snelheid

# Wat is een Function?

Een Function is een stuk code wat aangeroepen kan worden.

Een Class kan meerdere functies hebben.

Een Function is **ALTIJD** in camelCase!



```
<?php


class Car {
    public function showGreeting() {
        echo "Hello!";
    }
}
```

# Wat is een Access Modifier?

Een access modifier geeft aan hoe een Property of Function gebruikt mag worden.

Je hebt:

- public
- protected
- private



```
<?php

class Car {
    public $brand;

    protected $speed;

    private $fuelType;
}
```

# Wat doet de public Access Modifier?

Toegankelijk binnen de Class, erbuiten, en in subclasses.

Gebruik je voor Properties en Functions die open benaderbaar moeten zijn.



```
<?php  
  
class Car {  
    public $brand;  
  
    protected $speed;  
  
    private $fuelType;  
}
```



# Wat doet de protected Access Modifier?

Toegankelijk binnen de Class en in subclasses.

Gebruik je voor Properties en

Functions die alleen intern of in een uitbreidende class gebruikt mogen worden.

(Deze gaan we vrij weinig gebruiken, maar hij is belangrijk om te kennen)



```
<?php  
  
class Car {  
    public $brand;  
    protected $speed;  
    private $fuelType;  
}
```

# Wat doet de private Access Modifier?

Alleen toegankelijk binnen de Class zelf.

Gebruik je voor Properties en

Functions die mogelijk gevoelig

kunnen zijn, of problemen kunnen veroorzaken als ze buiten de class worden uitgevoerd.

```
<?php

class Car {
    public $brand;

    protected $speed;

    private $fuelType;
}
```



# Voorbeelden:

```
<?php

// We maken de class aan.
class Car {
    // We zetten een property
    public $brand;

    // We schrijven een function
    public function showBrand() {
        echo $this->brand;
    }
}

// We maken een nieuwe instantie van de Car class, en stoppen deze in $myCar
$myCar = new Car;
// We slaan een waarde op in de $brand property van $myCar;
$myCar->brand = "Toyota";
// We voeren de showBrand functie uit, die de brand van deze auto op het scherm gaat tonen.
$myCar->showBrand();
```

# Voorbeelden:

```

<?php

class Car {
    public $brand;

    public function showBrand() {
        echo $this->brand;
    }
}

$myCar = new Car;
$myCar->brand = "Toyota";

// We kunnen classes ook hergebruiken
$mySecondCar = new Car;
// Deze nieuwe instantie krijgt een andere waarde dan de eerste.
$mySecondCar->brand = "Audi";

// Nu kunnen we beide apart tonen.
$myCar->showBrand( );
$mySecondCar->ShowBrand( );
```

# Wat is \$this?

\$this wijst naar de huidige instantie van de Class. Je vind deze dus **ALTIJD** binnen een Class, nooit erbuiten.

- Properties ophalen: `$this->brand;`
- Functies uitvoeren: `$this->showBrand();`

# Meer voorbeelden:



```
<?php
```

```
class Car {  
    public $brand;  
  
    public function showBrand() {  
        echo $this->brand;  
    }  
}
```

```
$myCar = new Car;
```

```
// Zowel de brand property als de showBrand functie zijn public.  
// Dus mogen ze buiten de class aangepast en aangeroepen worden.
```

```
$myCar->brand = "Toyota";
```

```
$myCar->showBrand( );
```

# Meer voorbeelden:

```

<?php

class Car {
    private $brand;

    private function showBrand() {
        echo $this->brand;
    }
}

$myCar = new Car;

// De Property en Function zijn nu private en kunnen niet
// buiten de class aangeroepen worden. Dit geeft dus een error.
$myCar->brand = "Toyota";
$myCar->showBrand();

```

# Meer voorbeelden:

```
<?php

class Car {
    // We maken de property public
    public $brand;

    // We schrijven een private function die alleen binnen de class gebruikt mag worden.
    private function getBrand() {
        return $this->brand;
    }

    // We maken een public function aan die wel buiten de class gebruikt mag worden
    public function showBrand() {
        // We roepen een private function aan binnen deze class.
        echo $this->getBrand();
    }
}

$myCar = new Car;

// De property en Function zijn public, dus dit is geen probleem meer.
$myCar->brand = "Toyota";
$myCar->showBrand();

// Dit geeft nog steeds een error.
$myCar->getBrand();
```



# POP QUIZ

# Welke property is correct aangemaakt?

```
<?php  
  
class Car {  
    public $VarName;  
    public $var_name;  
    public $varName;  
    public $Car_name;  
}
```

A

\$VarName

\$var\_name

B

C

\$varName

\$Var\_name

D

# Welke Class is correct aangemaakt?

```
<?php
class classname {
}
class Classname {
}
class className {
}
class ClassName {
}
```

A

classname

Classname

B

C

className

ClassName

D

# Welke Function is correct aangemaakt?

```
<?php  
  
class ClassName {  
    public function functionName() {  
    }  
  
    public function FunctionName() {  
    }  
  
    public function functionname() {  
    }  
  
    public function function_name() {  
    }  
}
```

A

functionName

FunctionName

B

C

functionname

function\_name

D



```
<?php
```

```
class FunctionClass {  
    public function firstFunction() {  
  
    }  
  
    protected function secondFunction() {  
  
    }  
  
    private function thirdFunction() {  
  
    }  
}
```

```
$var = new FunctionClass;  
$var->secondFunction();
```

# Gaat dit goed, of krijgen we een error?

```
<?php  
  
class FunctionClass {  
    public function firstFunction() {  
    }  
  
    protected function secondFunction() {  
    }  
  
    private function thirdFunction() {  
    }  
}  
  
$var = new FunctionClass;  
$var->secondFunction();
```

A

Dit gaat goed

Error

B

# Welke functie mag wel uitgevoerd worden

```
<?php  
  
class FunctionClass {  
    public function firstFunction() {  
    }  
  
    protected function secondFunction() {  
    }  
  
    private function thirdFunction() {  
    }  
}  
  
$var = new FunctionClass;  
$var->secondFunction();
```

A

firstFunction

thirdFunction

B

# Wat zit er ook alweer in \$this

```
<?php
class FunctionClass {
    public $var;

    public function firstFunction() {
        echo $this->var;
    }
}
```

A

Alle properties van de class

Alle properties en functions van de huidige instantie

B

C

Alle properties en functions van alle bestaande classes

Alle functions van de class

D



# Mag dit?

```
<?php  
  
class FunctionClass {  
    public $var;  
  
    public function firstFunction() {  
        echo $this->var;  
    }  
}  
  
$var = new FunctionClass;  
$this->firstFunction();
```

A

Ja

Nee

B

# Opdracht!

Maak een .php file met het volgende:

- Een class genaamd Book.
- In de Book class wil ik de public properties voor de titel, genre, en de datum van uitgave.
- In de Book class wil ik een functie die het volgende op het scherm toont:  
"Het Boek [titel] is een [genre] verhaal die uit kwam op [datum]".
- Maak een instantie van de class, en voer de functie uit.