*Project Barroc-IT*

Applicatie- en mediaontwikkelaar  
periode 5  
Groep 5

Conventie rapport  
  
  
Groepsleden:  
  
1- Ahmad Khaled

2- Pieter Jan Kolijn

3- Laura Kruidhof

4- Mohamed Jelle

**Inhoudsopgave:**

1 Conventies3

1.1 HTML-Conventies3

1.2 CSS-Conventies3

1.3 PHP-Conventies3

1.3.1 Classes3

1.3.2 Variabelen4

1.3.3 Methoden4

1.3.4 Bestandsnamen4, 5

# 1. Conventies:

## 1.1 HTML-Conventies:

- Altijd met kleine letters werken.

- Denken aan een goed beschrijvende naamgeving.

- Comments zetten om de code wat overzichtelijker te houden.

- Denk aan het gebruik van tab.

- Boilerplate gebruiken.

- Je mag maar 1 h1 tag gebruiken.

- Denk eraan om met id’s te werken.

## 1.2 CSS-Conventies:

- Met comments werken om het overzichtelijk te houden.

## 1.3 PHP-Conventies:

1.3.1 Classes:De naam van een class is altijd een zelfstandig naamwoord in het enkelvoud. De eerste letter van een class naam is altijd een hoofdletter, de rest van de letters is altijd lowercase. Mocht de naam uit meerdere woorden bestaan dan worden deze gescheiden door een underscore. De eerste letter van opeenvolgende woorden is ook een hoofdletter.  
  
Voorbeelden van correcte class namen:  
- User  
- Message  
- Html\_Table  
- Message\_Store  
  
  
Voorbeelden van incorrecte class namen:  
- user (eerste letter geen hoofdletter)  
- Document\_PDF (te veel hoofdletters, DF van PDF moet lowercase)

1.3.2 Variabelen:Namen van properties/variabelen bestaan uit alfanumerieke tekens, alhoewel het gebruik van getallen in een naam afgeraden wordt. Ze beschrijven duidelijk de inhoud van de variabele. Mocht een variabele uit meerdere woorden bestaan, dan wordt de camelCase notatie gehanteerd, underscores horen daar niet thuis.  
  
Namen van private of protected properties worden altijd voorafgegaan door een underscore. Dit is de enige situatie waarin underscores in een variabele naam voorkomen.  
  
Voorbeelden van correcte property/variabele namen:  
  
- public $username  
- public $homeAddress  
- protected $\_databaseConnection  
- private $\_resultSet  
  
  
1.3.3 Methoden:De naam van een method beschrijft de handeling die de method uitvoert. Ze zijn volgens de camelCase notatie opgebouwd bestaan tevens enkel uit alfanumerieke tekens. Ook hier geldt dat private en protected methods voorafgegaan worden door een underscore.  
  
Voorbeelden van correcte method namen:  
  
- public function filterInput()  
- public function draw()  
- protected function \_calculateTotal()  
- private function \_countMessages()

1.3.4 Bestandsnamen:De naamgeving van je PHP-bestanden staat los van de naamgeving in je PHP-code, maar wil ik toch behandelen. Het is gebruikelijk om voor iedere class een apart PHP-bestand aan te maken. Met oog op een functionaliteit die ik in een later hoofdstuk zal behandelen, is het verstandig om de volgende conventie aan te houden: class\_naam.class.php. We zorgen dus dat de naam van de class terugkomt in de bestandsnaam van het bestand waarin die class staat. De class naam wordt daarbij geheel lowercase geschreven.

Voorbeelden van bestandsnamen:  
  
- user.class.php  
- message.class.php  
- html\_table.class.php  
- message\_store.class.php  
  
  
Zoals ik eerder al zei is naamgeving vooral een kwestie van persoonlijke smaak en bewust programmeren. Ook zijn er meer conventies dan alleen die van Zend, dus ik nodig je zeker uit om verder te kijken als deze je niet bevalt. Uiteindelijk komt het erop neer dat je in ieder geval voor jezelf een consistente naamgeving in je scripts moet gebruiken, anders zie je op een gegeven moment door de bomen het bos niet meer.  
  
Het wordt tijd om weer eens naar wat code te gaan kijken. In het volgende hoofdstuk bespreek ik het gebruik van de constructor van een class.