**PHP Tentamen Sheet**

Door: Mark Meerlo

Opstel datum: 04/11/2016

(Aan dit document kunnen geen rechten worden ontleend. Kan hier en daar een fout bevatten of incompleet zijn.)

**Conventies:**

**1. Naming**

* 1. Use meaningful names.

Use descriptive names for all identifiers (names of classes, variables and methods. Avoid ambiguity. Avoid abbreviations.

Gebruik namen die omschrijven waarvoor ze staan (bijvoorbeeld classes, variabelen & methoden) vermijdt het gebruik van dubbelzinnige woorden en afkortingen.

Simple mutator methods should be named set*Something*(…).

Simple accessor methods should be named get*Something*(…).

Accessor methods with Boolean return values are often called is*Something*(…), for example isEmpty().

1.2 Class names start with a capital letter.

* 1. Class names are singular nouns.

(enkelvoudige zelfstandige naamwoorden).

1.4 Methods and variable names start with lowercase letters.

All the – class, method and variable names – use capital letters in the middle to increase readability of compound identifiers, e.g. numberOfItems.

1.5Constants are written in UPPERCASE.

Constants occasionally use underscores to indicate compound identifiers:

MAXIMUM\_SIZE

**2. Layout**

2.1 One level of indentation is four spaces. (4spcs[ ] TAB[ ])

2.2 All statements within a block are intended one level.

2.3 Braces for classes and methods are alone on one line.

The braces for class and method blocks are on separate lines and are at the same indentation level, for example:

public int getAge() { **statements** }

2.4 Also for all other blocks, are alone on one line.

The braces for all other blocks are on separate lines and are at the same indentation level, for example:

while(condition)

{ **statements** }

if(condition)

{ **statements** }else

{ **statements** }

2.5 Always use braces in control structures

Braces are used in if-statements and loops even if the body is only a single statement.

2.6 Use a space before the opening brace of a control structure's block.

2.7 Use a space around operators.

2.8 Use a blank line between methods (and constructors).

Use blank lines to separate logical blocks of code. This means at least between methods, but also between logical parts within a method.

**3. Extra conventies:**

3.1 Uitbundig gebruik maken van comments: //COMMENT.

3.2 Zet bij elke code identificatie als volgt:

<!DOCTYPE html>

<?php

/\*

\* Filename :

\* Assignment :

\* Created :

\* Description:

\* Programmer :

\*

\*/

?>

<html>

<head>

<title></title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

</head>

<body>

</body>

</html>

**Voorbeelden:**

**Bijlage 1: Toelichting Chapters 2 en 13**

Op blz. 53 en 383 worden verscheidene voorbeelden gegeven aangaande de naamgeving van variabelen en functies. Echter, waar zij ervoor kiezen de ‘underscore’ te gebruiken, hanteren wij de zogenaamde ‘camel casing’ conventie, die besproken wordt op pagina 52:

“*De naam van een variabele begint altijd met een kleine letter (meteen na het*

*$-teken). Als de variabele uit meer woorden bestaat dan beginnen de volgende woorden steeds met een hoofdletter.”.*

Voorbeelden:

$teller

$mijnWoonplaats

$totaalAantalStudenten

**Bijlage 2: Toelichting Chapter 8**

Op *blz. 250* en volgend wordt het while- en het do-while statement behandeld. Bij elk van de voorbeelden gaat het om een zogenaamde tellende lus (die ook heel eenvoudig met een for-statement is te programmeren). Daarom volgt hier nog een voorbeeld met een niet tellende lus:

Stel we hebben een array *$woorden[]*; deze array bevat een groot aantal woorden. Een variabele *$woord* bevat een enkel woord. We moeten uitzoeken of de inhoud van *$woord* voorkomt in de array *$woorden*. Indien dit het geval is moet de index beschikbaar komen, zo niet, dan is de waarde -1.

function zoekWoord($woord, $woorden)

{

$gevonden = false;

$i = 0;

while($i < count($woorden) && !gevonden)

{

if($woord == $woorden[$i])

{

$gevonden = true;

}

else

{

$i++;

}

}

if($gevonden)

{

Return $i;

}

else

{

Return -1;

}

}

**Tentamenstof:**

1. **Bijlage 6**

6.1

Maak gebruik van de print statement.

print “Tekst”; - Print de tekst als plain text.

print “<h1>Tekst</h1>”; - Print de tekst als cursieve (h1) text.

6.2

Maak gebruik van de ‘echo’ statement.

echo “Tekst”; - Echoot de tekst als plain text.

echo “<h1>Tekst<h1>”; - Echoot de tekst als cursieve (h1) text.

6.4

Rekenen met variabelen.

<?php

$Celcius = 20;

$Fahrenheit = 0;

$Fahrenheit = $Celcius \* 9 / 5 + 32;

echo $Fahrenheit;

?>

6.8

Rekenen met de Modulo.

<?php

$Datum = 14062016;

$Dag = 0;

$Maand = 0;

$Jaar = 0;

// 10 % 3 = 1

//Hoe vaak past de 3 in 10?

// 10 / 3 = 3.33

// 3 \* 3 = 9

// 10 – 9 = 1.

**$Dag = $Datum / 1000000 % 100;**

// 14062016 / 1000000 % 100 = 14

//14062016 / 1000000 = 14.062016

//Hoe vaak past de 100 in 14.062016?

//14.062016 / 100 = 0

//14.062016 – 0 = 14

//14.062016 – (0 \* 14)

//14.<DIT> wordt weggegooid.

**$Maand = $Datum / 10000 % 100;**

//14062016 / 10000 % 100 = 6

//14062016 / 10000 = 1406.2016

//Hoe vaak past de 100 in 1406.2016?

// 1406.2016 / 100 = 14

// 1406.2016 - (100 \* 14)

// 1406.2016 - 1400

**$Jaar = $Datum % 10000;**

//14062016 % 10000 = 2016

//Hoe vaak past de 10000 in de 14062016?

// 14062016 / 10000 = 1406.2016

// 10000 \* 1406 = 14060000

// 14062016 – 14060000 = 2016

?>

<!DOCTYPE html>

<html>

<head>

<title>Opdracht 8</title>

</head>

<body>

<fieldset>

<?php

echo $Dag."/".$Maand."/".$Jaar;

?>

</fieldset>

</body>

</html>

<!DOCTYPE html>

<html>

<body>

<?php

echo date('H:i:s', time());

?>

</body>

</html>

6.9

Tijd / Datum weergeven

De ‘:’, ‘/’ & ‘.’ Tussen de tekens veranderen de weergave.

H : Uur.

i : Minuut.

s : Seconde.

Tijd weergeven

<!DOCTYPE html>

<html>

<body>

<?php

echo date('H:i:s', time());

?>

</body>

</html>

**Output:**

21:30:12

HH:MM:SS

Datum weergeven

d : Dag.

m : Maand.

y : Jaar.

<!DOCTYPE html>

<html>

<body>

<?php

echo date('d/m/Y', time());

?>

</body>

</html>

**Output:**

04/11/2016

DD/MM/YYYY

1. **Bijlage 7**

7.2

Switch(case) & if/elseif/else structuren.

Or / || = OR

And / && = AND

Xor = 1 van de 2 is juist, niet beide.

if / else if / else:

if($appel < $peer)

{

echo “De appel is kleiner als de peer.”;

}

else if($appel == $peer)

{

echo “Ze zijn beiden even groot!”;

else

{

echo “De peer is kleiner als de appel.”;

}

Switch case:

$waarde = 5;

Switch ($waarde)

{

case “1”;

case “2”;

case “3”;

echo “het is gelijk aan 1,2 of 3.”

//Stopt de case, voorwaarde is voldaan.

Break;

case “4”;

case “5”;

echo “Het is gelijk aan 4 of 5.”;

//Zonder break gaat het door tot de eerstvolgende break;

//break;

Case “6”;

echo “Het is gelijk aan 6.”;

break;

//De default geeft iets weer als de $waarde ongeldig / leeg is.

default:

echo “Ongeldige invoer.”

break;

}

**Output:**

Het is gelijk aan 4 of 5. Het is gelijk aan 6.

7.6

Loops

For:

For($i = 0; $i <= 2; $i++)

{

echo 'Dit is nr.' . $i . "<br>";

}

**Output:**

Dit is nr.0  
Dit is nr.1

While:

$i = 0;

while($i <= 2)

{

echo 'Dit is nr.' . $i . "<br>";

$i++;

}

**Output:**

Dit is nr.0  
Dit is nr.1

Do while:

$i = 0;

do{

echo 'Dit is nr.' . $i . "<br>";

$i++;

}while($i <= 2);

**Output:**

Dit is nr.0  
Dit is nr.1

Dit is nr.2

7.8

Nested Loops

Alle variabelen die in een functie staan, kunnen niet buiten die functie worden opgeroepen als er geen global instaat.

<?php

**//Lengte / herhaal variable alle loops.**

$length = 10;

**//Benoem een functie**

function lusI(){

global $length;

$lusses = $length;

**//Eerste loop**

while($lusses >= 0){

$amount = $lusses;

$lusses--;

echo"<br>";

**//Tweede loop**

while($amount > 0){

$amount--;

echo "\*";

}

}

}

**//Benoem een functie**

function lusII(){

global $length;

$check = $length + 1;

**/Eerste loop.**

for($lusses = $check; $lusses >= 0;){

$lusses--;

echo"<br>";

**//Tweede loop**

for($amount = $lusses; $amount > 0;){

$amount--;

echo"\*";

}

}

}

**//Benoem een functie**

function lusIII(){

global $length;

$lusses = $length;

**//Eerste loop**

do{

$amount = $lusses;

$lusses--;

echo"<br>";

**//Tweede loop**

do{

$amount--;

echo "\*";

}

**//Loop voorwaarde**

while($amount > 0);

}

**//Loop voorwaarde**

while($lusses > 0);

}

**//Functies worden aangeroepen.**

lusI();

lusII();

lusIII();

?>

**Output(2x):**

\*\*\*\*\*\*\*\*\*\*  
\*\*\*\*\*\*\*\*\*  
\*\*\*\*\*\*\*\*  
\*\*\*\*\*\*\*  
\*\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*  
\*\*\*  
\*\*  
\*

7.9

Nested Loops (getallen)

$length = 6;

function lusI(){

global $length;

$lusses = $length;

while($lusses >= 0){

$amount = $lusses;

$lusses--;

$count = 0;

echo"<br>";

while($amount > 0){

$amount--;

$count++;

echo $count . " ";

}

}

}

function lusIII(){

global $length;

$lusses = $length;

do{

$amount = $lusses;

$lusses--;

$count = 0;

echo"<br>";

do{

$amount--;

$count++;

echo $count . " ";

}

while($amount > 0);

}

while($lusses > 0);

}

**Output(2x):**

1 2 3 4 5 6   
1 2 3 4 5   
1 2 3 4   
1 2 3   
1 2   
1

1. **Bijlage 7**

8

String bewerkingen:

**(Deel I)**

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title></title>

</head>

<body>

<?php

**// String? --> Afgeleide van array, ieder teken (character)** staat op een bepaalde positie!

$hogeschool = "<p>Stenden";

$hogeschool .= " is een hogeschool.</p>";

echo $hogeschool;

**// strlen() --> Retourneert het aantal chars (karakters).**

$nummer = 45654;

echo "<p>" . strlen($nummer) . "</p>";

**// str\_word\_count() --> Retourneert het aantal woorden.**

$film2 = "Alice in Wonderland";

echo "<p>" . str\_word\_count($film2) . "</p>";

**// ucfirst() --> Eerste woord, alleen de eerste letter UpperCase.**

**// lcfirst() --> Eerste woord, alleen de eerste letter LowerCase.**

**// ucwords() --> Alle woorden, eerste letter UpperCase.**

**// strtolower() --> Alle woorden lowercase.**

**// strotoupper() --> Alle woorden uppercase.**

$quote = "May ThE FOrCe bE WIth yOU!";

echo ucfirst($quote) . "<br />";

echo lcfirst($quote) . "<br />";

**//2e bewerking 1e bewerking**

echo ucwords(strtolower($quote)) . "<br />";

?>

</body>

</html>

**Output:**

Stenden is een hogeschool.

5

3

May ThE FOrCe bE WIth yOU!  
may ThE FOrCe bE WIth yOU!  
May The Force Be With You!

**(Deel II)**

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title></title>

</head>

<body>

<?php

**//substr() --> Retourneert chars, hierbij kan een**

**startwaarde en eindwaarde worden meegegeven.**

**//Als de waarde negatief is dan wordt**

**achteraan de variabele begonnen.**

$php = "hypertext preprocessor";

echo substr($php, 4) . "<br />";

echo substr($php, 0, 9) . "<br />";

echo substr($php, -12) . "<br />";

echo substr($php, -7, 5) . "<br />";

**// strrev() --> Omdraaien string**

echo "<p>" . strrev($php) . "</p><br />";

**// strpos() --> Retourneert de positie (getal) van de**

**meegegeven variabele.**

**// strchr() --> Retourneert het gedeelte van de**

**string dat begint vanaf de aangegeven variabele.**

**// strrchr() --> Retourneert het gedeelte van de**

**string dat begint vanaf de aangegeven variabele,**

**//maar begint nu achteraan met zoeken.**

$email = "student@stenden.com";

echo strpos($email, '@') . "<br />";

echo strchr($email, 's') . "<br />";

echo strrchr($email, 's') . "<br />";

?>

</body>

</html>

**Output:**

rtext preprocessor  
hypertext  
preprocessor  
ocess

rossecorperp txetrepyh

7  
student@stenden.com  
stenden.com

**(Deel III)**

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title></title>

</head>

<body>

<?php

**// str\_replace() --> Vervangt het aangegeven gedeelte van een**

**variabele.**

**// str\_ireplace() --> Idem, maar deze functie is case**

**'i'nsensitive.**

$email = "student@stenden.com";

$newemail = str\_replace("student", "naam", $email);

echo $newemail . "<br />";

**// substr\_replace() --> Vervangt het aantal chars aangegeven door**

**de indexwaarden.**

$newemail2 = substr\_replace($newemail, "leerling", 0, 4);

echo $newemail2 . "<br />";

**// String -> Array**

$cat = "longcat is long";

$arraySplit = str\_split($cat, 6);

echo "<pre>";

print\_r($arraySplit);

echo "</pre>";

echo "<pre>";

var\_dump($arraySplit);

echo "</pre>";

**// explode() --> Breekt een string op in array-elementen op**

**het aangegeven teken/char.**

$arrayExplode = explode(" ", $cat);

echo "<pre>";

var\_export($arrayExplode);

echo "</pre>";

**// Array -> String**

$catsArray = array("Bob", "Bob2", "Bob3");

print\_r($catsArray);

echo "<br />";

**// implode() --> Maakt een string uit de elementen van een**

**array.**

$cats = implode(" ", $catsArray);

echo $cats;

?>

</body>

</html>

**Output:**

naam@stenden.com  
leerling@stenden.com

Array

(

[0] => longca

[1] => t is l

[2] => ong

)

array(3) {

[0]=>

string(6) "longca"

[1]=>

string(6) "t is l"

[2]=>

string(3) "ong"

}

array (

0 => 'longcat',

1 => 'is',

2 => 'long',

)

Array ( [0] => Bob [1] => Bob2 [2] => Bob3 )   
Bob Bob2 Bob3

8.1

String bewerken vanuit een tekstveld:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

<form action="" method="POST">

<textarea name="textValue" placeholder="Plaats

hier tekst..."></textarea>

<input type="submit" value="Submit"/>

</form>

<?php

if(isset($\_POST["textValue"]))

{

$textValue = $\_POST["textValue"];

$arrayExplode = explode(" ", $textValue);

$end = "";

$begin = "";

$pos = 0;

foreach($arrayExplode as $word)

{

//echo $word ."<br>";

}

echo "<br>";

foreach($arrayExplode as $word)

{

if(strlen($word) >= 4)

{

$pos = (strlen($word) - 1);

// echo $word;

// echo $pos . "<br><br>";

$end = substr($word, $pos);

// echo $end . "<br>";

$begin = substr($word, 0, 1);

// echo $begin . "<br>";

$print = substr\_replace($word, $end,

0,1);

$final = substr\_replace($print, $begin,

$pos,1);

echo $final . "<br>";

}

else

{

echo $word . " <br>";

}

}

}

?>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

<form action="#" method="POST">

<p> <input type="text" placeholder="Name" name='name'></p>

<p><input type="password" placeholder="password" name="pass"></p>

<p><input type="submit" value="submit"></p>

</form>

<?php

if (isset($\_POST['name']))

{

$name = strip\_tags($\_POST['name']);

$pass = strip\_tags($\_POST['pass']);

$length = strlen($pass);

echo "<p>Je naam is " . $name . "</p>";

echo "<p>Je wachtoord is " . $length . " karakters lang.</p>";

}

?>

</body>

</html>

1. **Bijlage Index**

<html>

<head>

<meta charset="UTF-8">

<title>Inhouds Opgave</title>

</head>

<body>

<h1>Inhouds Opgave</h1>

<?php

**//waarin $y het aantal opdrachten is**.

$y = 10;

for ($x = 1; $x <= $y; $x++){

echo "<a href='http://localhost/**<TENTAMENLOCATIE>**/Opdracht$x.php'>Opdracht $x</a><br>";

}

?>

</body>

</html>

1. **Maanden van het jaar:**

1.Januari

2.Februari

3.Maart

4.April

5.Mei

6.Juni

7.Juli

8.Augustus

9.September

10.Oktober

11.November

12.December

1. **Van string naar int:**

**$string = “32”;**

**$int = intval($string);**

**Radio Buttons**

**<form>**

**<input type="radio" name="gender" value="male" checked> Male<br>**

**<input type="radio" name="gender" value="female"> Female<br>**

**<input type="radio" name="gender" value="other"> Other**

**</form>**

**<?php**

**if (isset($\_POST['submit']))**

**{**

**$naam = strip\_tags($\_POST['naam']);**

**$mail = strip\_tags($\_POST['mail']);**

**if(!empty($naam))**

**{**

**if(!empty($mail) && strpbrk($mail , '@' ))**

**{**

**echo "<div id= 'respons'>" . $naam . " bedankt voor je bericht! We zullen zo spoedig mogelijk een antwoord versturen naar " . $mail . ". </div>";**

**}**

**Else**

**{**

**echo "Het 'E-mail Adres' veld is leeg, of het adres is mogelijk ongeldig.";**

**}**

**}**

**Else**

**{**

**echo "Het 'Naam + Achternaam' veld is leeg.";**

**}**

**}**

**else{**

**}**

**?>**