erasmus

HOGESCHOOL BRUSSEL

Web Development Advanced

erasmus

HOGESCHOOL BRUSSEL

Formuliervalidatie met regular expressions

VALIDATIE VAN GEGEVENS

- Wij kunnen op onze webpagina formuliervelden voorzien om gegevens in te vullen.
- We kunnen er niet zeker van zijn dat de gebruiker deze juist gebruikt
- Daarom zullen we een aantal checks moeten doen zodat we er min of meer zeker van zijn dat de formuliervelden juist gebruikt werden

erasmus HOGESCHOOL BRUSSEL

php

FORMULIERGEGEVENS VERWERKEN

- In de praktijk zal je tegelijkertijd ook clientside soortgelijke checks uitvoeren.
  - Snellere feedback aan gebruiker
  - Minder load op de server

REGULAR EXPRESSIONS

- Een manier om patronen in strings te gaan opzoeken.
- Wordt gebruikt in verschillende programmeer- en scriptingtalen.
- Implementatie via bibliotheek.
- Wij gebruiken de PCRE/Perl-stijl. Wordt door de meeste programmeertalen ondersteund.

GEBRUIK

- 2 aspecten nodig
  - Een PHP regular expression functie
  - De regular expression zelf (het patroon)

GEBRUIK

- 2 aspecten nodig
  - Een PHP regular expression functie
  - De regular expression zelf (het patroon)

ENKELE PHP FUNCTIES

- `preg_match($patroon, $teOnderzoeken)`
  - Zoekt $patroon in $teOnderzoeken
  - Returnwaarde is één van volgende
    - 1 wanneer $patroon voorkomt in $teOnderzoeken
    - 0 wanneer $patroon niet voorkomt in $teOnderzoeken
    - FALSE wanneer zich een fout voordeed

ENKELE PHP FUNCTIES

```
<?php
  echo preg_match("/auto/", "Dit
  is mijn auto.");
  //Geeft 1

  echo preg_match("/outo/", "Dit
  is mijn auto.");
  //Geeft 0
?>
```

ENKELE PHP FUNCTIES

- `preg_replace($patroon, $vervanging, $teOnderzoeken)`
  - Zoekt $patroon in $teOnderzoeken en vervangt de voorkomens met $vervanging
  - Werkt niet alleen met strings, maar ook met vb arrays

ENKELE PHP FUNCTIES

```
<?php
  echo preg_replace(
     “/auto/”,
     “limousine”,
     “Dit is mijn auto.”
  );

  //Geeft:
  //Dit is mijn limousine.
?>
```

ENKELE PHP FUNCTIES

- `preg_filter`
  - Zelfde als preg_replace, maar deze functie retourneert enkel indien er een match is
  - De reden waarom deze functie bestaat is omdat preg_filter en preg_replace “mixed” inputs hebben. Zij kunnen bijvoorbeeld ook een array als argument binnen krijgen.

ENKELE PHP FUNCTIES

- `preg_grep($patroon, $inputArray)`
  - Geeft een array met de elementen van $inputArray die aan het patroon voldoen

ENKELE PHP FUNCTIES

```
<?php
   $inputArray = array(
      “een”,
      “auto”,
      “drie”,
      “auto”
   );
   $resultaat = preg_grep(“/auto/”, $inputArray);

   //$resultaat[0]=“auto”;
   //$resultaat[1]=“auto”;
?>
```

erasmus HOGESCHOOL BRUSSEL

ENKELE PHP FUNCTIES

- `preg_match_all($patroon, $teOnderzoeken)`
  - Geeft het **aantal matches** terug of FALSE wanneer er zich een fout voordeed
  - Optioneel kan deze functie ook het resultaat teruggeven in een array die verschillende manieren gesorteerd kan worden => bekijk de PHP manual voor alle mogelijkheden

ENKELE PHP FUNCTIES

```
<?php
   echo preg_match_all (
      “/auto/”,
      “Deze auto is mijn auto.”
   );

   //Geeft: 2.
?>
```

ENKELE PHP FUNCTIES

- `preg_split($patroon, $teOnderzoeken)`
  - Splitst $teOnderzoeken op door middel van een patroon

ENKELE PHP FUNCTIES

```
<?php
   $resultaat = preg_split(
      “/auto/”,
      “Deze auto is mijn auto.”
   );
   //$resultaat[0]= “Deze”
   //$resultaat[1]= “is mijn”
   //$resultaat[2]= “.”
?>
```

erasmus HOGESCHOOL BRUSSEL

ENKELE PHP FUNCTIES

- `preg_quote($teOnderzoeken)`
  - Neemt $teOnderzoeken en zet een backslash voor elk karakter dat onderdeel is van de regular expression syntax
  - Handig als je een zoekstring hebt die vb door een gebruiker wordt ingegeven en die speciale karakters kan bevatten

ENKELE PHP FUNCTIES

```
<?php
  echo preg_quote("Deze auto is
  mijn auto.");

  //geeft: Deze auto is mijn
  auto\.
?>
```

ANDERE PREG FUNCTIES

- Bekijk de PHP manual voor een overzicht van de andere preg functies of voor meer info over optionele parameters van de uitgelegde functies

GEBRUIK

- 2 aspecten nodig
  - Een PHP regular expression functie
  - De regular expression zelf (het patroon)

PCRE SYNTAX

- Welke regular expression patronen kunnen we gebruiken in onze preg-functies?
- Hier bekijken we de belangrijkste onderdelen van de syntax. Zoek op het internet een goede reference voor een uitgebreid overzicht.
- Er bestaan tools waarmee je gemakkelijk een regular expression kan uitproberen.

BASIS

- Het begin en einde wordt aangeduid met een slash /
- /auto/
  - zoekt naar voorkomens van het woord auto
- ^ duidt op het begin van de te onderzoeken string
- $ duidt op het einde van de te onderzoeken string

BASIS

```
<?php
   preg_match(“/auto/”, “Dit is
   mijn auto.”);
   //Gevonden
?>
```

erasmus
HOGESCHOOL BRUSSEL

BASIS

```
<?php
  preg_match("/^auto/", "Dit is
  mijn auto.");
  //Niet gevonden, want auto
  staat niet in het begin van de
  string

  preg_match("/^Dit/", "Dit is
  mijn auto.");
  //Wel gevonden
?>
```

erasmus
HOGESCHOOL BRUSSEL

BASIS

```
<?php
  preg_match("/auto$/", "Dit is
  mijn auto.");
  //Niet gevonden, want .

  preg_match("/auto$/", "Dit is
  mijn auto");
  //Wel gevonden
?>
```

KARAKTERKLASSE

```
<?php
  echo preg_replace(
     “/[0123456789]/”,
     “1”,
     “847654398435468”
  );
  //Geeft: 1111111111111111
?>
```

KARAKTERKLASSE

- Je kan de tekens die je wilt zoeken met een karakterklasse ook verkort opschrijven
- Voorbeelden:
  - [0123456789] = [0-9]
  - [012abc] = [0-2|a-c] = [0-2a-c]

KARAKTERKLASSE

```
<?php
  echo preg_replace(
     “/[0-9]/”,
     “1”,
     “847654398435468”
  );
  //Geeft: 1111111111111111
?>
```

KARAKTERKLASSE

- Je kan ook specificeren dat de tekens binnen een bepaalde karakterklasse juist niet mogen gematcht worden. Dit doe je met een ^ **BINNEN** de []
- Voorbeeld:
  - [^012] => komt overeen met alle tekens, behalve 012

KARAKTERKLASSE

```
<?php
  echo preg_replace(
     “/[^0-2]/”,
     “1”,
     “abcdefg 0123456789”
  );
  //Geeft: 1111111101211111111
?>
```

KARAKTERKLASSE

- LET OP: als de ^ BUITEN de vierkante haakjes staat, dan moet de karakterklasse aan het begin van de te onderzoeken string voorkomen.

KARAKTERKLASSE

```
<?php
  echo preg_replace(
     “/^[0-2]/”,
     “9”,
     “0123456789”
  );
  //Geeft: 9123456789
?>
```

SPECIALE KARAKTERS

- Als je wilt testen op speciale karakters die deel uitmaken van de regular expression syntax, kan je deze karakters escapen door middel van een backslash \
- Voorbeeld:
  - [\\\$] => test op backslash of dollarteken

SPECIALE KARAKTERS

- Volgende karakters hebben een speciale betekenis, en moeten eventueel geescaped worden

\	*
^	+
$	(
.	)
|	[
?	]

QUANTIFIERS

- Geeft een minimum en een maximum aantal voorkomens mee, gescheiden door een komma
- Kunnen toegepast worden op gewone karakters, karakterklasses, subpatronen, …
- Voorbeeld:
  - {1,2} => minimum 1 en maximum 2 voorkomens
  - {1,} => minimum 1 voorkomen
  - {1} => exact 1 voorkomen

QUANTIFIERS

```
<?php
   echo preg_replace("/[0-9]{1}/", "z",
   "a1b22c333d4444");
   //Geeft: azbzzczzzdzzzz

   echo preg_replace("/[0-9]{2}/", "z",
   "a1b22c333d4444");
   //Geeft: a1bzcz3dzz

   echo preg_replace("/[0-9]{3}/", "z",
   "a1b22c333d4444");
   //Geeft: a1b22czdz4
?>
```

erasmus HOGESCHOOL BRUSSEL

QUANTIFIERS

```
<?php
   echo preg_replace("/[0-9]{1,2}/",
   "z", "a1b22c333d4444");
   //Geeft: azbzczzdzz

   echo preg_replace("/[0-9]{1,}/",
   "z", "a1b22c333d4444");
   //Geeft: azbzczdz
?>
```

QUANTIFIERS

- Sommige quantifiers hebben een verkorte notatie die je als alternatief kan gebruiken

Verkort	Betekenis
*	{0,}
+	{1,}
?	{0,1}

SUBPATRONEN

- Met ronde haakjes kan je subpatronen maken
- Op subpatronen kan je quantifiers, condities en modifiers toepassen

SUBPATRONEN

```
<?php
   echo preg_replace(
      “/([a-z]{1,}\.)/”,
      “0”,
      “voornaam.achternaam@gmail.com”
   );

   //Geeft: 0achternaam@0com
?>
```

SUBPATRONEN

```
<?php
    echo preg_replace(
        “/(rood|blauw)/”,
        “z”,
        “rood groen blauw geel”
    );
    //Geeft: z groen z geel

    echo preg_replace(
        “/(rood|blauw){2,}/”,
        “z”,
        “rood groen blauw geel”
    );
    //Geeft: rood groen blauw geel
?>
```

erasmus HOGESCHOOL BRUSSEL

MODIFIERS

- Een karakter die op het einde van het patroon wordt toegevoegd en het hele patroon beïnvloeden

i	Maakt het matchen case-**in**sensitive
m	De onderzochte string mag uit meerdere lijnen bestaan
s	Zorgt ervoor dat **.** ook met newlines overeenkomt

MODIFIERS

```
<?php
   preg_match(“/Auto/i”, “Dit is mijn auto.”);
   //Wel gevonden

   preg_match(“/auto/i”, “Dit is mijn Auto”);
   //Wel gevonden

   preg_match(“/Auto/”, “Dit is mijn auto.”);
   //Niet gevonden

   preg_match(“/auto/”, “Dit is mijn Auto”);
   //Niet gevonden
?>
```

erasmus
HOGESCHOOL BRUSSEL

php

erasmus

HOGESCHOOL BRUSSEL

MySQL en PHP

MYSQL

- DBMS = Database Management Systeem
- PHP kan met veel “merken” van databases samenwerken
  - MySQL
  - MS SQL
  - Oracle
  - …

erasmus HOGESCHOOL BRUSSEL

DATABASES

- Data op een gestructureerde manier bijhouden
- Laat toe om website volledig dynamisch te maken
- De inhoud wordt opgeslagen in de database, en wordt met behulp van een sjabloon weergegeven.

CMS

- Content Management Systeem
- Een gewone gebruiker kan een website volledig onderhouden via gewone HTML formulieren
- Voorbeelden:
  - Drupal
  - Joomla
  - Wordpress
- Minder technische kennis noodzakelijk
- Gebaseerd op een database

erasmus HOGESCHOOL BRUSSEL

php

QUERIES

- Gemakkelijk gegevens opvragen en manipuleren door middel van een query language zoals SQL
- Geavanceerde opdrachten mogelijk
  - Zoals het verbinden van gegevenstabellen.
  - Sorteren van gegevens.
- Meer en performantere mogelijkheden tov opslaan van data in tekstbestanden.

DATABASES: SERVERS

- Kunnen op een fysiek aparte server worden gehost tov de webserver
  - Schaalbaarder
  - Hergebruik
    - in andere applicaties en verschillende delen van de webapplicatie
    - Ook als deze andere applicaties op een ander type OS draaien als de database.
- Databases kunnen aangesproken worden over het netwerk.
- Databases zijn vaak cross platform

DATABASES: BEVEILIGING

- Databases kunnen afgeschermd worden door het toewijzen van security rechten voor verschillende gebruikers.
- Gebruikers hebben een wachtwoord.

MYSQL

- Specifiek “merk” van database
- Gebruikt SQL als query language
- Wordt het meest met PHP gecombineerd
- Gratis
- Performant
- Betrouwbaar
- Wereldwijd gebruikt

MYSQL EN PHP

- Er dient een PHP extensie geïnstalleerd te worden opdat PHP met MySQL kan samenwerken => wij gebruiken mysqli
- Biedt API (Application Programming Interface) aan om te verbinden met een MySQL database.
- Is reeds geïnstalleerd op onze computers

MYSQL EN PHP: GEBRUIKEN

- Eerst verbinding openen met MySQL
- Met constructor van mysqli
- Geef volgende parameters mee:
  - MySQL server adres
  - MySQL gebruikersnaam + wachtwoord
  - MySQL databasenaam
- Returnt object dat connectie met database vertegenwoordigt.

MYSQL EN PHP: GEBRUIKEN

```
<?php
   $mysqli = new mysqli(
      "localhost",
      "user",
      "pass",
      "dbname“
   );
?>
```

MYSQL EN PHP: GEBRUIKEN

- Controleren of verbinding goed gemaakt werd
- Met methode connect_error
  - Geeft foutmelding terug of NULL
  - Foutmelding beschrijft de fout

MYSQL EN PHP: GEBRUIKEN

```
<?php
    $mysqli = new mysqli("localhost", "user", "pass", "dbname");
    if($mysqli->connect_error) {
        die('Connect Error (' . $mysqli->connect_errno . ') '.
        $mysqli->connect_error);
    }

    //hier uw code, bijvoorbeeld
    echo 'Success... ' . $mysqli->host_info . "\n";

    $mysqli->close();
?>
```

erasmus HOGESCHOOL BRUSSEL

MYSQL EN PHP: GEBRUIKEN

- Gebruik van “die” is enkel acceptabel bij het ontwikkelen
  - In extra informatie die gegeven wordt kan gebruiker zien wat er fout is gelopen
  - Eventuele informatie over structuur van database, … kan publiek worden => vulnerability
- Wel acceptabel in productie:
  - Melding naar de gebruiker toe
  - Logging en/of automatische mail naar adminstrator voor verdere afhandeling.

MYSQL EN PHP: GEBRUIKEN

- `$mysqli->host_info`
  - Geeft een string terug die de connectie voorstelt. (oa: server host name)
  - In dit geval:
    - voorbeeld code om de connectie te gebruiken
    - Niet “verplicht” dus bij het gebruikt van MySQL met PHP

MYSQL EN PHP: GEBRUIKEN

- `$mysqli->close()`
  - Sluit een connectie die daarvoor geopend werd
  - Maakt resources terug vrij, vermijdt memory leaks

QUERIES UITVOEREN

```
<?php
  $mysqli = new mysqli("localhost", "user",
  "pass", "dbname");

  //controle connectie

  $result = $mysqli->query("insert into
  producten values ('iPhone', 'Een
  telefoon', 500)");
  $mysqli->close();
?>
```

QUERIES UITVOEREN

- `$mysqli->query`
  - Voert een SQL query uit
  - Returnt
    - FALSE wanneer er zicht een fout heeft voorgedaan
    - een mysqli_result object bij eventuele resultaten
    - TRUE wanneer alles goed is gegaan en er geen resultaten zijn.

QUERIES UITVOEREN

- `$mysqli->affected_rows`
  - Geeft een int terug die het aantal rijen weergeeft dat is aangepast bij het laatste insert/update/replace/delete SQL commando
  - Voor select statements doet dit hetzelfde als `mysqli_result->num_rows`

QUERIES UITVOEREN

- `$mysqli->real_escape_string`
  - Om te vermijden dat strings die je als parameter bij een SQL opdracht meegeeft als SQL code worden geïnterpreteerd
  - Altijd doen!

SELECT STATEMENTS

- 3 manieren om resultaat uit te lezen:
  - fetch_object
  - fetch_array
  - fetch_row

SELECT STATEMENTS

fetch_object

- Kort:
  - zet de waarden van een rij uit de resultatenset om naar een object
- Hoe:
  - Kopieert de kolomwaarden van een rij naar de properties van een overeenkomstig object.
  - De namen van de properties komen overeen met de kolomnamen van de resultaatset uit de database

SELECT STATEMENTS

```
//...
if($result = $mysqli->query( "SELECT * FROM producten")) {
   //controleer of rijen geselecteerd
   if($result->num_rows > 0) {
      $product = $result->fetch_object();
      echo "<p>Naam: $product->productnaam<br/>";
      echo “Prijs: $product->prijs </p>\n";
   }
}
//...
```

SELECT STATEMENTS

fetch_array

- Haalt een rij op uit de resultset en stopt deze rij in een associatieve array.
- De namen van de velden komen overeen met de sleutels van de array.
- Vb: $product[‘productnaam’]

SELECT STATEMENTS

fetch_row

- Haalt een rij op uit de resultset en stopt deze rij in een numerieke array (met indexen).

SELECT STATEMENTS

Meerdere lijnen

- Resultatensets die je van de database terugkrijgt bij een select statement kunnen uiteraard meerdere lijnen bevatten.
- Het gebruik van één van de 3 fetch methoden zal automatisch de pointer naar een rij in de resultatenset verplaatsen naar de volgende rij.
- De fetch methodes returnen false wanneer er geen volgende lijn meer is.

SELECT STATEMENTS

```
//...
if($result = $mysqli->query( "SELECT * FROM
producten")) {
   //controleer of rijen geselecteerd
   if($result->num_rows > 0) {
      while($product = $result->fetch_object()) {
         echo "<p>Naam: $product->naam<br/>";
         echo “Prijs: $product->prijs</p>\n";
      }
   }
}
//...
```

erasmus

HOGESCHOOL BRUSSEL

Bestanden en PHP

TEKSTBESTANDEN

- Gegevens in een tekstbestand lezen, wegschrijven en toevoegen
- PHP biedt hier ondersteuning voor onder de vorm van verschillende commando's.
  - fopen
  - fwrite
  - fgets
  - fclose

FOPEN

- Opent een bepaalde resource
- Een resource kan een lokale file zijn of zich ergens op een URL bevinden.
- Verzeker u ervan dat de webserver toegang heeft tot de file en voldoende rechten erop heeft om de acties uit te voeren die je in gedachten hebt.

FOPEN

- `fopen(filename, mode)`
  - De filename is de locatie van het bestand dat je wilt openen
  - Met de mode kan je aangeven welk type access je op the bestand wilt.

FOPEN

- Belangrijkste modes:

Mode	Read or write?	Waar staat pointer?	File leegmaken?
r	Read only	Begin	Nee
r+	Read/write	Begin	Nee
w	Write only	Begin	Ja
w+	Read/write	Begin	Ja
a	Write only	Einde	Nee
a+	Read/write	Einde	Nee

- Fopen heeft nog meer optionele parameters. Bekijk de PHP manual voor meer informatie.

FGETS

- Leest een lijn van de huidige positie in de file
- `fget(resource)`
  - De resource is het return object dat je van fopen hebt teruggekregen
  - De resource mag nog niet opnieuw gesloten zijn door het fclose commando
  - Returnt false wanneer er geen data meer te lezen valt op de huidige positie.
  - Gaat automatisch naar de volgende lijn
- Mogelijke configuratie nodig op Mac's wanneer zij niet goed de nextlinecharacters herkennen (auto_detect_line_endings)

FWRITE

- Schrijft een string naar een file
- `fwrite(resource, string)`
  - De resource is het return object dat je van fopen hebt teruggekregen
  - De resource mag nog niet opnieuw gesloten zijn door het fclose commando
  - De string is de string die weggeschreven moet worden
  - Returnt het aantal bytes dat werd weggeschreven, of FALSE bij een error

FCLOSE

- Files die je met fopen geopend hebt, moeten ook opnieuw afgesloten worden. Dit doe je met fclose
- `fclose(resource)`
  - De resource is het return object dat je van fopen hebt teruggekregen
  - Geeft true terug wanneer alles OK is en false wanneer de handeling niet gelukt is.

ANDERE METHODES

- Er zijn nog andere methodes die je kan gebruiken om files te manipuleren.
- Je kan een referentie terugvinden op de PHP website. http://us3.php.net/manual/en/ref.filesystem.php

erasmus

HOGESCHOOL BRUSSEL

PHP Frameworks

FRAMEWORK

- Bestaat uit klassebibliotheken
- Geschreven door vele ontwikkelaars over de hele wereld
- Bieden veel functionaliteit voor het ontwikkelen van webapplicaties

erasmus
HOGESCHOOL BRUSSEL

VOORDELEN FRAMEWORK

- Tijd
  - Ingebouwde functionaliteiten hergebruiken
- Stabiliteit
  - Vaste structuur, zoals MVC
- Veiligheid

erasmus
HOGESCHOOL BRUSSEL

MVC

- = Model-View-Controller
- = ontwerppatroon dat een complexe, objectgeoriënteerde applicatie opdeelt in drie onderdelen met elk zijn eigen verantwoordelijkheden:
  - Datamodel (Model)
  - Presentatie (View)
  - Logica (Controller)

erasmus
HOGESCHOOL BRUSSEL

MODEL

- definieert de gegevens gebruikt in de applicatie
- Bevatten functies voor ophalen en wegschrijven gegevens
- Laag tussen database en PHP

erasmus
HOGESCHOOL BRUSSEL

VIEW

- Presentatie van de informatie
- Definieert UI elementen
- Bestaat voornamelijk uit HTML, CSS en JS
- ! Verwerking van gegevens worden nooit in view gedaan

erasmus
HOGESCHOOL BRUSSEL

CONTROLLER

- Verwerkt en reageert op gebeurenissen
- Stuurt gegevens naar model

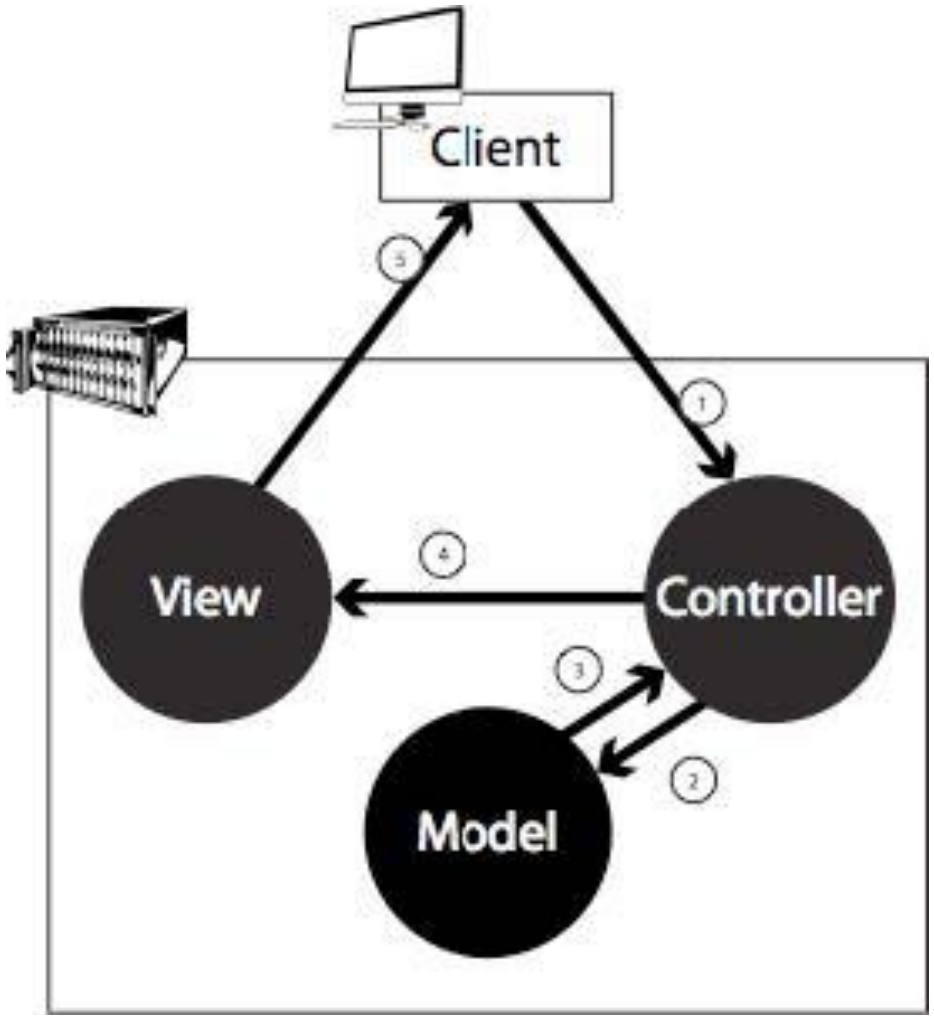
erasmus
HOGESCHOOL BRUSSEL

MVC IN PHP

- HTML uitvoer enkel in de view
- SQL-queries enkel in model

erasmus
HOGESCHOOL BRUSSEL

MVC IN PHP



FRAMEWORKS

- Laravel
- Symfony
- CodeIgniter
- Yii
- Phalcon
- CakePHP
- Zend

erasmus
HOGESCHOOL BRUSSEL