

# ΔΟΜΗ

---

Διδάσκοντες: Π. Αγγελάτος, Δ. Ζήνδρος

Επιμέλεια διαφανειών: Δ. Ζήνδρος

Σχολή Ηλεκτρολόγων Μηχανικών  
και Μηχανικών Υπολογιστών



# Στόχος της ώρας

- Δόμηση κώδικα
- Συνεργασία σε ομάδες
- Χωρισμός σε αρχεία
- Στυλ κώδικα
- Front-end και back-end
- Το πρότυπο MVC

# Τίποτα δεν γίνεται από έναν

- Facebook:
  - **4** ιδρυτές
  - 1700 άτομα σήμερα
- Google:
  - **2** ιδρυτές
  - 23000 άτομα σήμερα
- Microsoft: **2** ιδρυτές
- Apple: **3** ιδρυτές
- YouTube: **3** ιδρυτές

...πρέπει να μάθουμε να συνεργαζόμαστε!

# Δόμηση κώδικα

- Καθαρότερος κώδικας **για εμάς**
  - Επεξεργασία μετά από **ένα μήνα** ή ένα χρόνο
- Καθαρότερος κώδικας για τους άλλους
  - Κι άλλοι διαβάζουν τον κώδικά σας
  - Πρέπει να τον καταλάβουν
  - Πρέπει να τον αλλάξουν
  - Πρέπει να τον ξανακαταλάβουμε

...χρειαζόμαστε μία κοινή συνισταμένη

# Δόμηση κώδικα

- Πλεονεκτήματα ορθής δόμησης: Maintainability
- Ο κώδικας μπορεί να αλλάξει εύκολα
- Είναι προσαρμόσιμος
- Είναι επεκτάσιμος
- Είναι ευανάγνωστος
- Εκφράζει την σκέψη μας


Programs must be written for **people** to read, and only incidentally for machines to execute.

# Στυλ κώδικα

- Το στυλ κώδικα ορίζει την μορφή του κώδικα
- Πού έχει κενά και πού όχι
- Πόσα κενά έχει
- Πότε γίνεται indentation
- Πότε γίνονται αλλαγές γραμμών
- κ.ό.κ.

# Δύο διαφορετικά στυλ

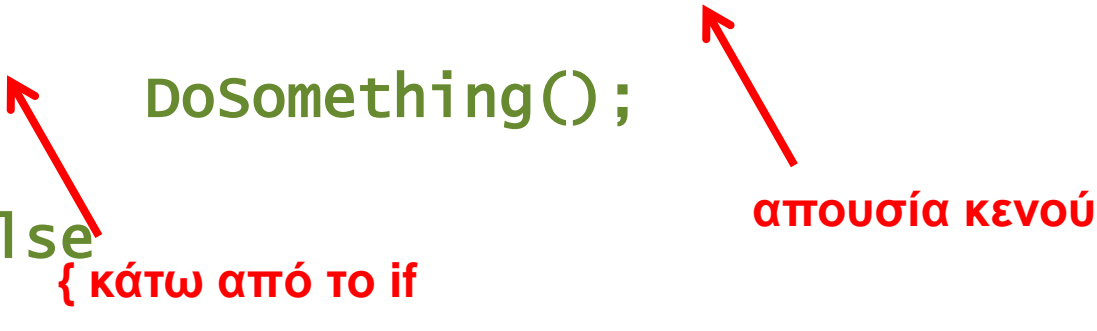
```
if ( isset( $foo[ 'bar' ] ) ) {  
    doSomething();  
}  
else {  
    doSomethingElse();  
}
```



κενό { δίπλα στο if

---

```
if(isset($Foo['bar']))  
{  
    DoSomething();  
}  
else  
{  
    DoSomethingElse();  
}
```



απουσία κενού { κάτω από το if



# Στυλ κώδικα

- Ασυνέπεια:

```
if ( $a ) {  
    doSomething();  
}  
else { doSomethingElse(); }
```

# Στυλ κώδικα

- Δεν έχει σημασία ποιο στυλ χρησιμοποιείτε
- Δεν έχει σημασία αν έχετε πολλά κενά ή λίγα
- **Διαλέξτε** ένα στυλ και **κρατήστε** το
- Μία ομάδα → Ένα στυλ
- Ένα λογισμικό → Ένα στυλ
- Συνέπεια! Οτιδήποτε άλλο είναι αντιπαραγωγικό

# Στυλ κώδικα

- Αν διαφωνείτε στο στυλ...
- Πιο σημαντικό να γράφετε **στο ίδιο στυλ** με τους συναδέλφους σας
- Παρά να γράφετε σε διαφορετικά στυλ στην ίδια ομάδα
- Σε ένα ήδη υπάρχον project, χρησιμοποίησε το στυλ που βλέπεις ήδη

# Στυλ κώδικα

- Υπάρχουν έτοιμες προτάσεις
  - Mozilla
  - Google
  - και άλλες. Αναζητήστε για “Coding Style”
- Βρείτε μία που σας ταιριάζει

# Συνεργασία σε ομάδες

- Απαιτείται καταμερισμός εργασιών
- Αυτό σημαίνει **σωστή δόμηση**
- Χωρισμός σε αρχεία
- Όλοι πρέπει να συμφωνήσουμε:
  - Τι είδους κώδικα γράφουμε πού;
- Πού θα γραφεί ο κώδικας...
  - PHP, για την σύνδεση στη βάση δεδομένων;
  - HTML, για την φόρμα δημιουργίας λογαριασμού;
  - SQL, για την δημιουργία λογαριασμού;
- **Κάθε** πράγμα πρέπει να έχει **μία** θέση

# Χωρισμός σε αρχεία

- Πρώτη ιδέα:
  - Αντί να έχω ένα μεγάλο αρχείο, έχω πολλά μικρότερα
- Ευκολότερη συνεργασία
- Ο καθένας δουλεύει στα αρχεία που τον αφορούν

# Συναρτήσεις

- Επαναχρησιμοποίηση κώδικα
- Συχνά μπορούν να χρησιμοποιηθούν σε αρχεία

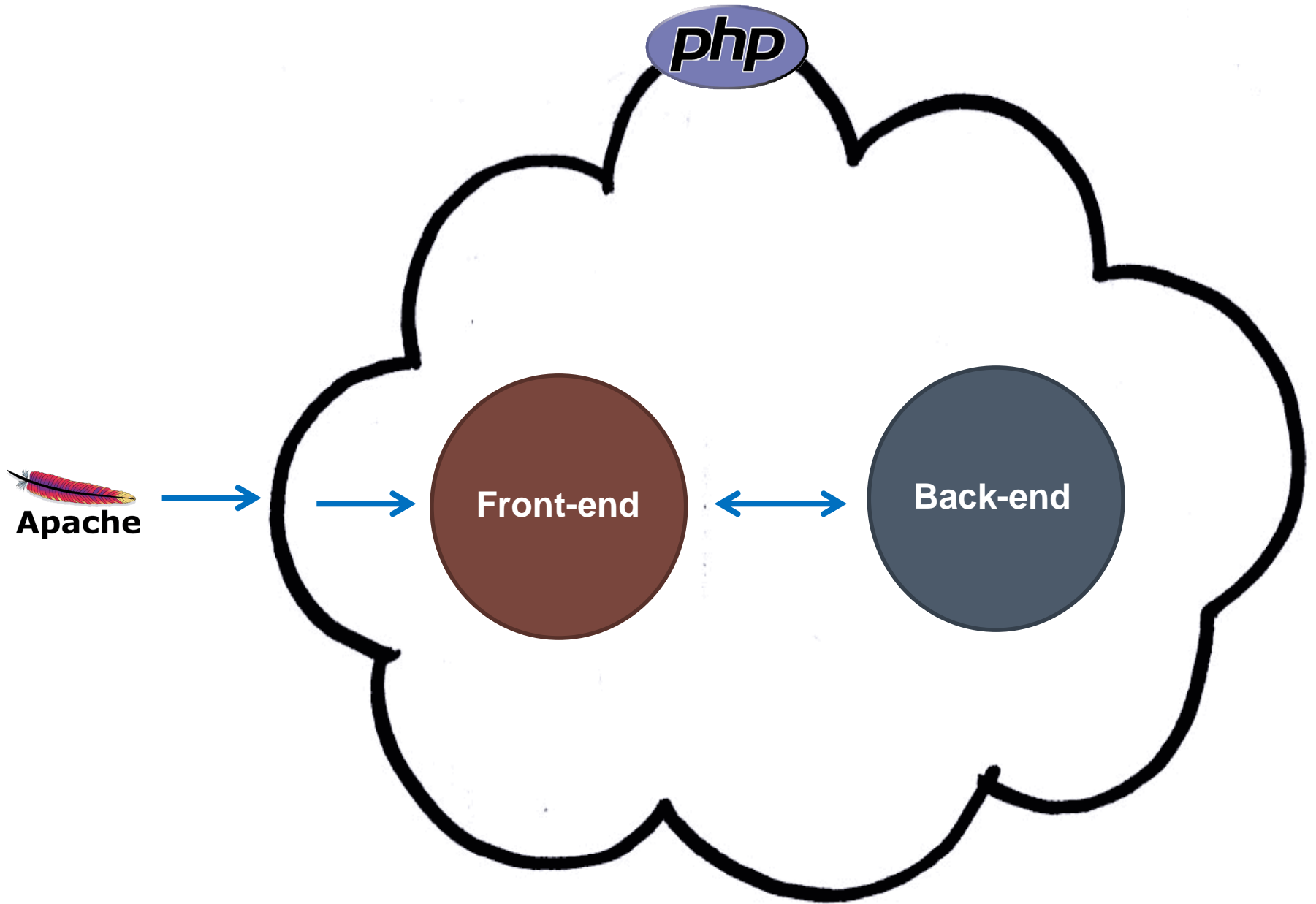
# Front-end και back-end

- Η πρώτη μεγάλη ιδέα!
- Front-end:
  - Κώδικας που αναφέρεται στην διεπαφή χρήστη
  - PHP που παράγει άμεσα HTML
  - Στατικό HTML
  - CSS
  - Javascript
  - Εικόνες
  - κ.ό.κ.



# Front-end και back-end

- Back-end:
  - Κώδικας που επεξεργάζεται τα δεδομένα
  - PHP που δεν παράγει HTML
  - Συναρτήσεις επεξεργασίας δεδομένων
  - Συναρτήσεις αποθήκευσης δεδομένων
  - SQL



# Front-end και back-end

- Χωρίζουμε σε ξεχωριστά αρχεία
  - Το front-end
  - Το back-end
- Στο front-end δεν υπάρχει SQL
- Στο back-end δεν υπάρχει HTML/CSS
- Το **front-end** περιγράφει την παραγωγή της **διεπαφής** **χρήστη**
- Το **back-end** περιγράφει την **επεξεργασία, αποθήκευση, ανάκτηση** δεδομένων

# Front-end και back-end

- Ας τα χωρίσουμε!
- Το back-end συχνά αναφέρεται και ως **models**
- Συχνό φαινόμενο σε web εφαρμογές:
  - **Φάκελος models** που περιέχει τον back-end κώδικα

# Decoupling

- Κάθε τμήμα κώδικα γνωρίζει μόνο όσα **χρειάζεται** να γνωρίζει
- Κάθε είδους «γνώση» υπάρχει μόνο σε **ένα** σημείο του κώδικά μας
- Front-end γνωρίζει:
  - Ότι χρησιμοποιούμε HTML
  - Ποια έκδοση της HTML χρησιμοποιούμε
- Back-end γνωρίζει:
  - Ότι χρησιμοποιούμε MySQL
  - Ποια έκδοση της MySQL χρησιμοποιούμε
  - Ποιο είναι το σχήμα μας

# Decoupling

- Front-end **δεν** γνωρίζει:
  - Αν χρησιμοποιούμε **αρχεία** ή **βάση δεδομένων** για αποθήκευση
  - Αν αποθηκεύω το όνοματεπώνυμο ως όνομα + επώνυμο
  - Αν τα δεδομένα προέρχονται από ανάκτηση ή από υπολογισμό
- Back-end **δεν** γνωρίζει:
  - Ότι παράγουμε HTML
  - Αν χρησιμοποιούμε XHTML 1.0 Strict ή όχι
  - Ότι ο χρήστης είναι άνθρωπος και όχι αράχνη

```
<h2>Τι είπαν οι άλλοι</h2>
```

```
<ol>
```

```
<?php
```

```
    $res = mysql_query(
```

```
        "SELECT
```

```
            username, text
```

```
        FROM
```

```
            shouts CROSS JOIN users
```

```
            ON shouts.userid = users.userid
```

```
        ORDER BY
```

```
            created DESC;"
```

```
    );
```

```
    while ( $row = mysql_fetch_array( $res ) ) {
```

```
        ?><li><strong><?php
```

```
        echo $row[ 'username' ];
```

```
        ?>:</strong>
```

```
        <span><?php
```

```
        echo $row[ 'text' ];
```

```
        ?></span></li><?php
```

```
    }
```

```
?>
```

```
</ol>
```

front-end

back-end

front-end



Σαλάτα από τον [Chiot's Run](#)



```
<h2>Τι είπαν οι άλλοι</h2>
```

```
<ol>
```

```
<?php
```

```
    $res = mysql_query(
```

```
        "SELECT
```

```
            username, text
```

```
        FROM
```

```
            shouts CROSS JOIN users
```

```
            ON shouts.userid = users.userid
```

```
        ORDER BY
```

```
            created DESC;"
```

```
    );
```

```
    while ( $row = mysql_fetch_array( $res ) ) {
```

```
        ?><li><strong><?php
```

```
        echo $row[ 'username' ];
```

```
        ?>:</strong>
```

```
        <span><?php
```

```
        echo $row[ 'text' ];
```

```
        ?></span></li><?php
```

```
    }
```

```
?>
```

```
</ol>
```

front-end

back-end

front-end

# Πώς θα χωρίσουμε αυτό τον κώδικα;

- Το HTML στο front-end
- Η SQL στο back-end

models/shouts.php:

```
$res = mysql_query(  
    "SELECT  
        username, text  
    FROM  
        shouts CROSS JOIN users  
    ON shouts.userid = users.userid  
    ORDER BY  
        created DESC;"  
);
```

back-end



```
<h2>Τι είπαν οι άλλοι</h2>
```

```
<ol>
```

```
<?php
```

```
    include 'models/shouts.php';
```

```
    while ( $row = mysql_fetch_array( $res ) ) {
```

```
        ?><li><strong><?php
```

```
        echo $row[ 'username' ];
```

```
        ?>:</strong>
```

```
        <span><?php
```

```
        echo $row[ 'text' ];
```

```
        ?></span></li><?php
```

```
    }
```

```
?>
```

```
</ol>
```

έλλειψη διαφάνειας αναφοράς

back-end

back-end

front-end ?

```
<h2>Τι είπαν οι άλλοι</h2>
```

```
<ol>
```

```
<?php
```

```
include 'models/shouts.php' ;
```

```
$shouts = GetShouts() ;
```

← διεπαφή (API) front-end/back-end

```
foreach ( $shouts as $shout ) {
```

```
    ?><li><strong><?php
```

```
    echo $shout[ 0 ] ;
```

```
    ?>:</strong>
```

```
    <span><?php
```

```
    echo $shout[ 1 ] ;
```

```
    ?></span></li><?php
```

```
}
```

```
?>
```

```
</ol>
```

models/shouts.php:

```
function GetShouts() {  
    $rows = array();  
    $res = mysql_query(  
        "SELECT username, text  
        FROM  
            shouts CROSS JOIN users  
        ON shouts.userid = users.userid  
        ORDER BY created DESC;"  
    );  
    while ( $row = mysql_fetch_array( $res ) ) {  
        $rows[] = array(  
            $row[ 'username' ], $row[ 'text' ]  
        );  
    }  
    return $rows;  
}
```

# Front-end και back-end

- Back-end που ζει στον φάκελο “models”:
  - Εμπεριέχει όλη την **λογική** επικοινωνίας με την βάση
  - Αποθήκευση
  - Ανάκτηση
  - Επεξεργασία
  - **Δεν** γνωρίζει για τον τρόπο χρήσης των δεδομένων
  - **Δεν** γνωρίζει από πού προήλθαν τα δεδομένα

# Front-end και back-end

- Front-end:
  - Εμπεριέχει όλη την **λογική** διεπαφής χρήστη
  - Παραγωγή HTML
  - CSS/JS
  - **Δεν** γνωρίζει για τον τρόπο αποθήκευσης των δεδομένων
  - **Δεν** γνωρίζει πώς ανακτούνται τα δεδομένα



# Συνεργασία σε ομάδες

- Σε μία ομάδα ο κάθε προγραμματιστής επιλέγει αν θα ασχοληθεί με το back-end ή το front-end
- Ταχύτερη ανάπτυξη
- Δεν γράφεται το ίδιο πράγμα 2 φορές

```
if ( isset( $_SESSION[ 'username' ] )
    && isset( $_POST[ 'shout' ] ) ) {
    $shout = $_POST[ 'shout' ];
    mysql_query(
        "INSERT INTO
          shouts
        SET
          text = '" . $shout . "',
          userid = " . $_SESSION[ 'userid' ] . ",
          created = NOW();"
    );
    header( 'Location: index.php' );
}
else {
    ?>Πρέπει να έχεις κάνει είσοδο.<?php
}
```

front-end

back-end

front-end

front-end

```
graph TD
    FE1[front-end] --> L1["if ( isset( $_SESSION[ 'username' ] ) && isset( $_POST[ 'shout' ] ) ) { $shout = $_POST[ 'shout' ]; mysql_query( \"INSERT INTO shouts SET text = '\" . $shout . \"', userid = \" . $_SESSION[ 'userid' ] . \", created = NOW();\" ); header( 'Location: index.php' ); } else { ?>Πρέπει να έχεις κάνει είσοδο.<?php }"]
    BE[back-end] --> L2["mysql_query( \"INSERT INTO shouts SET text = '\" . $shout . \"', userid = \" . $_SESSION[ 'userid' ] . \", created = NOW();\" );"]
    FE2[front-end] --> L3["header( 'Location: index.php' );"]
    FE3[front-end] --> L4["?>Πρέπει να έχεις κάνει είσοδο.<?php"]
```

models/shouts.php:

```
function SaveShout( $userid, $text ) {  
    mysql_query(  
        "INSERT INTO  
        shouts  
        SET  
        text = '$text',  
        userid = $userid,  
        created = NOW();"   
    );  
}
```

```
if ( isset( $_SESSION[ 'username' ] )
    && isset( $_POST[ 'shout' ] ) ) {
    include 'models/shouts.php' ;

    SaveShout(
        $_SESSION[ 'userid' ],
        $_POST[ 'shout' ]
    );

    header( 'Location: index.php' );
}
else {
    ?>Πρέπει να έχεις κάνει είσοδο.<?php
}
```

# Τι είναι front-end και τι back-end?

- Εμφάνιση μηνύματος σφάλματος (Δεν έχεις κάνει login)
- Ανάγνωση δεδομένων GET
- Ανάγνωση δεδομένων POST
- Προσθήκη εγγραφής στη βάση δεδομένων
- Εμφάνιση κουμπιού που αποθηκεύει
- Χρήση `mysql_fetch_array` για ανάγνωση δεδομένων
- Εύρεση μέσου όρου βαθμολογιών μαθητών
- Χρωματισμός γραμμάτων διεπαφής με κόκκινο χρώμα

# MVC

- Model-View-Controller
- Καλός ο διαχωρισμός Front-end / Back-end
- Back-end = Model
- Όμως στο Front-end μας μπλέκουμε δύο πράγματα
  - Την παραγωγή της διεπαφής (View)
  - Την συλλογή δεδομένων και τις απαραίτητες κλήσεις (Controller)

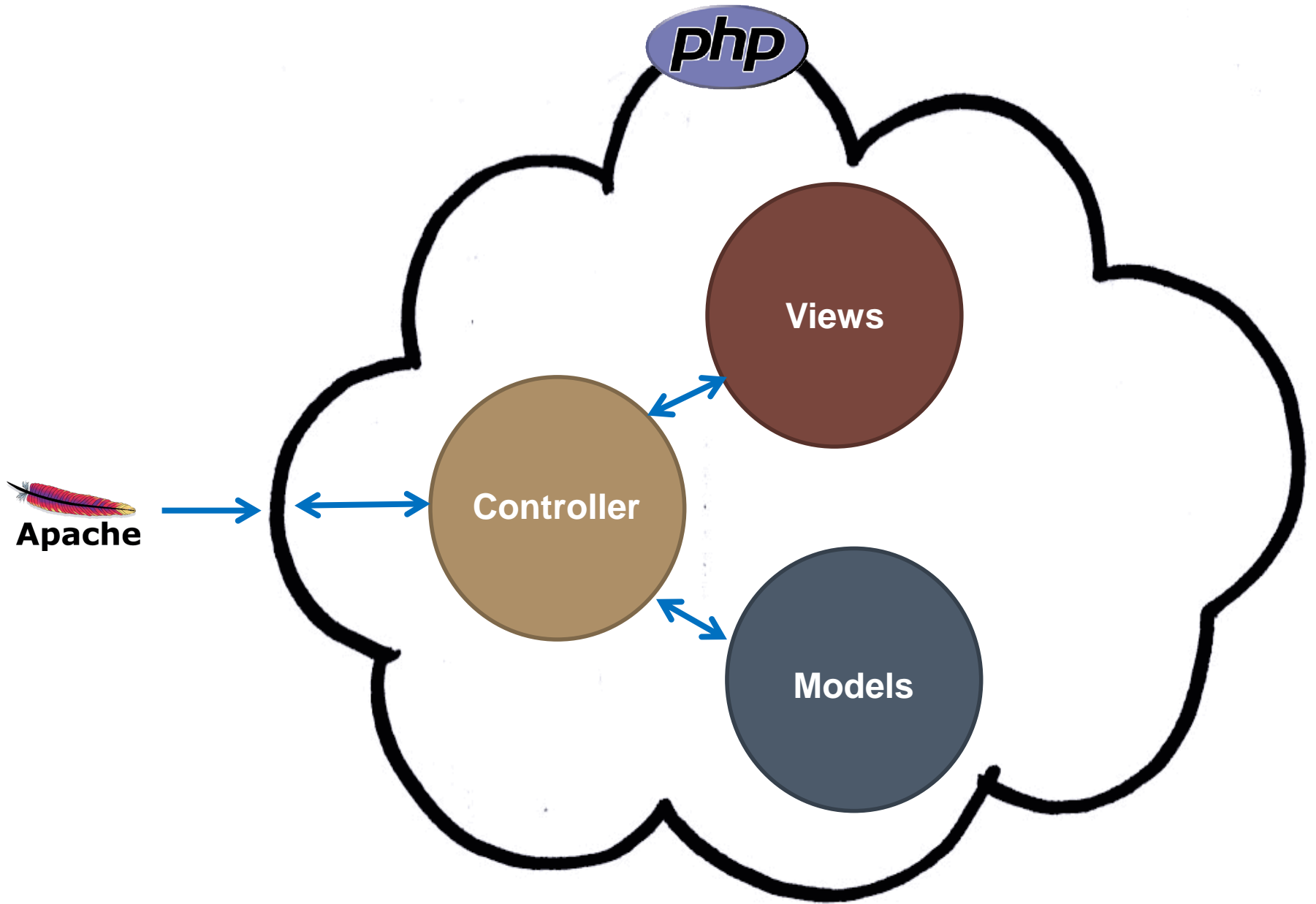
# Views

- Αποκλειστικά υπεύθυνα για την παραγωγή διεπαφής
- Κυρίως HTML κώδικας
- Μόνο η απαραίτητη PHP
  - π.χ. foreach για παραγωγή επαναλαμβανόμενων τμημάτων
- Πρόσβαση σε συγκεκριμένες ονομασμένες μεταβλητές
- Λέγονται και «**Templates**» ή πρότυπα

# Controllers

- Αποφασίζουν ποιο view και ποιο model θα κληθεί
  - Παίρνουν τα δεδομένα από τον χρήστη
  - Δίνουν τα **δεδομένα του χρήστη** στο σωστό model
  - Δίνουν τα δεδομένα του χρήστη **και του model** στο view
  - Στέλνουν πίσω τα δεδομένα στον χρήστη
- 
- Το «λεπτότερο» τμήμα
  - Το πρώτο και τελευταίο πράγματα που τρέχει
  - Έχει τον **έλεγχο** (controller = ελεγκτής)





```
if ( isset( $_SESSION[ 'username' ] )
    && isset( $_POST[ 'shout' ] ) ) {
    include 'models/models/shouts.php' ;
```

```
    SaveShout(
        $_SESSION[ 'userid' ],
        $_POST[ 'shout' ]
    );

    header( 'Location: index.php' );
}
else {
    ?>Πρέπει να έχεις κάνει είσοδο.<?php
}
```

Controller

Controller


View

```
if ( isset( $_GET[ 'username' ] ) ) {
    $res = mysql_query(
        "SELECT userid
        FROM users
        WHERE
            username = '" . $_GET[ 'username' ] . "'
        LIMIT 1;" )
    );
    if ( mysql_num_rows( $res ) == 1 ) {
        ?>Username already taken :-(
        <input type="text" value="" /><?php
    }
    else {
        ?><input type="text" value="<?php
        echo $_GET[ 'username' ];
        ?>" /><?php
    }
}
else {
    ?><input type="text" value="" /><?php
}
```

**model**

**controller**

**view**



# 1. Controller

```
$exists = false;
$username = '';
if ( isset( $_GET[ 'username' ] ) ) {
    include 'models/user.php';
    $exists = UsernameExists( $_GET[ 'username' ] );
    $username = $_GET[ 'username' ];
}
include 'views/register.php';
```

## 2. Model

```
function UsernameExists( $username ) {  
    $res = mysql_query(  
        "SELECT  
            userid  
        FROM  
            users  
        WHERE  
            username = '$username'  
        LIMIT 1;"  
    );  
    return mysql_num_rows( $res ) == 1;  
}
```

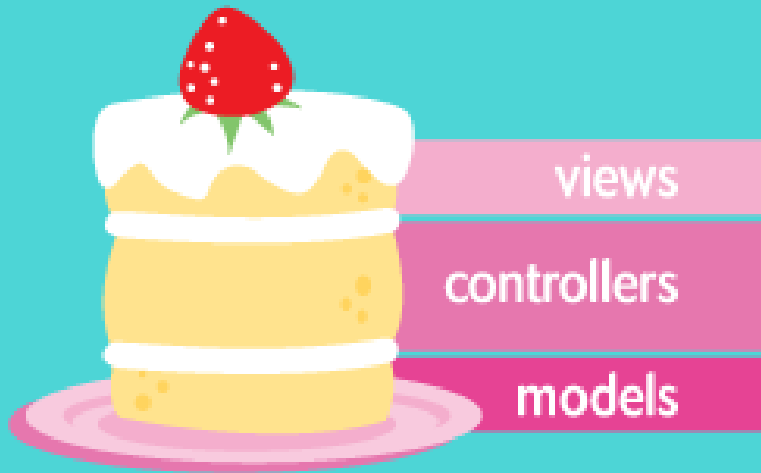
## 2. View

```
<?php
if ( $exists ) {
    ?>Το όνομα χρήστη υπάρχει ήδη :-(<?php
}
?>
<input type="text" value="<?php
    echo $username;
?>" />
```

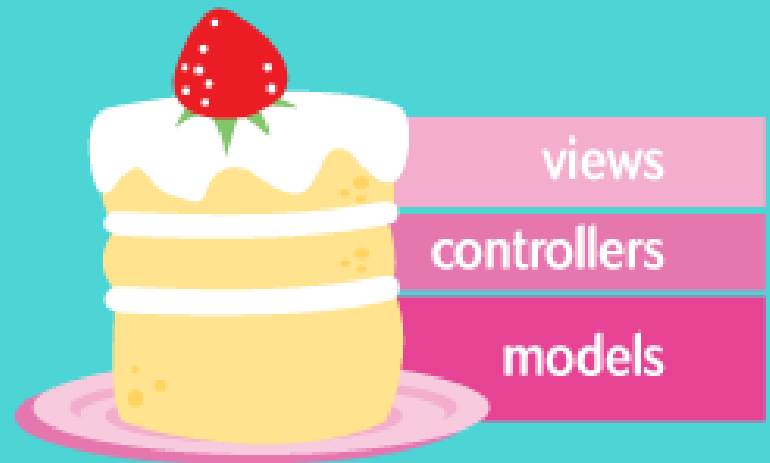
# Σε ποιο τμήμα του MVC ανήκουν;

- Εμφάνιση μηνύματος σφάλματος (Δεν έχεις κάνει login)
- Ανάγνωση δεδομένων GET
- Ανάγνωση δεδομένων POST
- Προσθήκη εγγραφής στη βάση δεδομένων
- Εμφάνιση κουμπιού που αποθηκεύει
- Χρήση `mysql_fetch_array` για ανάγνωση δεδομένων
- Εύρεση μέσου όρου βαθμολογιών μαθητών
- Χρωματισμός γραμμάτων διεπαφής με κόκκινο χρώμα
- Include του view
- Include του model

# Bad cake!



# Good cake!





# Συγχαρητήρια!

- Μπορείτε να δομείτε τον κώδικά σας σωστά!



# Την επόμενη φορά...

- Εισαγωγή στην Javascript
  - **Πόσες** γλώσσες προγραμματισμού θα χρειαστούμε πια;