Σχεδίαση Εφαρμογών και Υπηρεσιών Διαδικτύου
12η Διάλεξη: Επανάληψη / Ανακεφαλαίωση

Δρ. Απόστολος Γκάμας

Λέκτορας (407/80)

gkamas@uop.gr

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

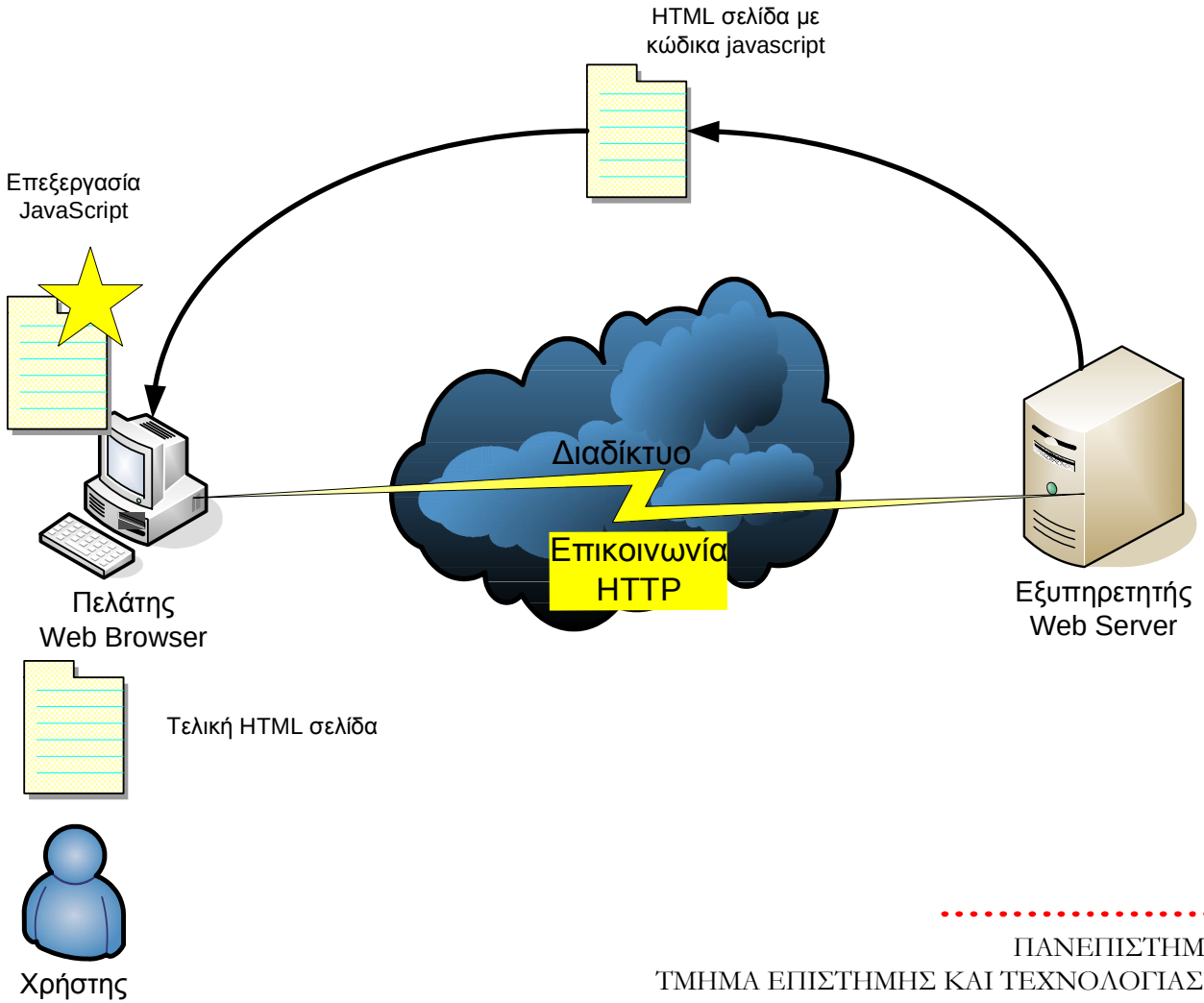
Η αρχιτεκτονική του WWW

- Βασική αρχιτεκτονική του WWW
  - HTTP
  - HTML
  - URL
- Επέκταση της βασικής αρχιτεκτονικής του WWW
  - Προγραμματισμός στην πλευρά του πελάτη (JavaScript,..)
  - Προγραμματισμός στην πλευρά του εξυπηρετητή (CGI, php, jsp,..)
  - Plug-ins
  - Εφαρμογές πολλών στρωμάτων (n-tier – application servers)
  - Web Services (SOAP,…)
  - XML
  - …

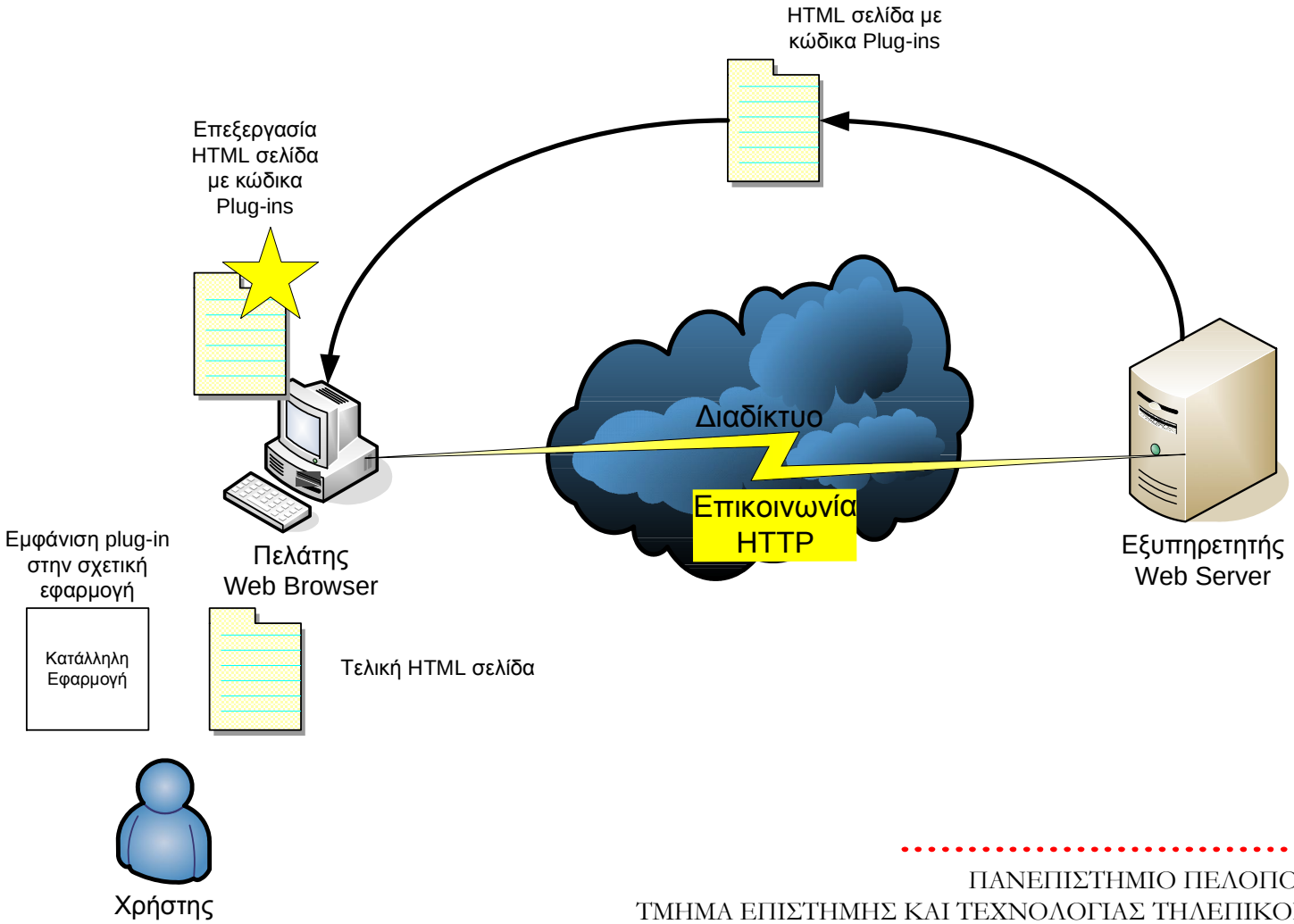
Δυναμικές Ιστοσελίδες

- Μία ιστοσελίδα είναι δυναμική όταν:
  - αλληλεπιδρά με το χρήστη (π.χ. αλλάζει η εμφάνιση ενός μενού επιλογών όταν ο δείκτης του ποντικιού τοποθετείται πάνω σε αυτό),
  - αλλάζει η μορφή της (π.χ. μετακινούνται λέξεις και αντικείμενα, αλλάζουν δυναμικά εικόνες, γράμματα και χρώματα),
  - μεταβάλλεται το περιεχόμενό της (π.χ. αλλάζουν τα περιεχόμενα ενός πίνακα).
- Για την ενημέρωση και την αλλαγή των περιεχόμενων μιας ιστοσελίδας πολλές φορές απαιτείται:
  - απομακρυσμένη αναζήτηση δεδομένων και
  - αλληλεπίδραση της ιστοσελίδας με αρχεία ή βάσεις δεδομένων.

Προγραμματισμός στην πλευρά του πελάτη (JavaScript,..)



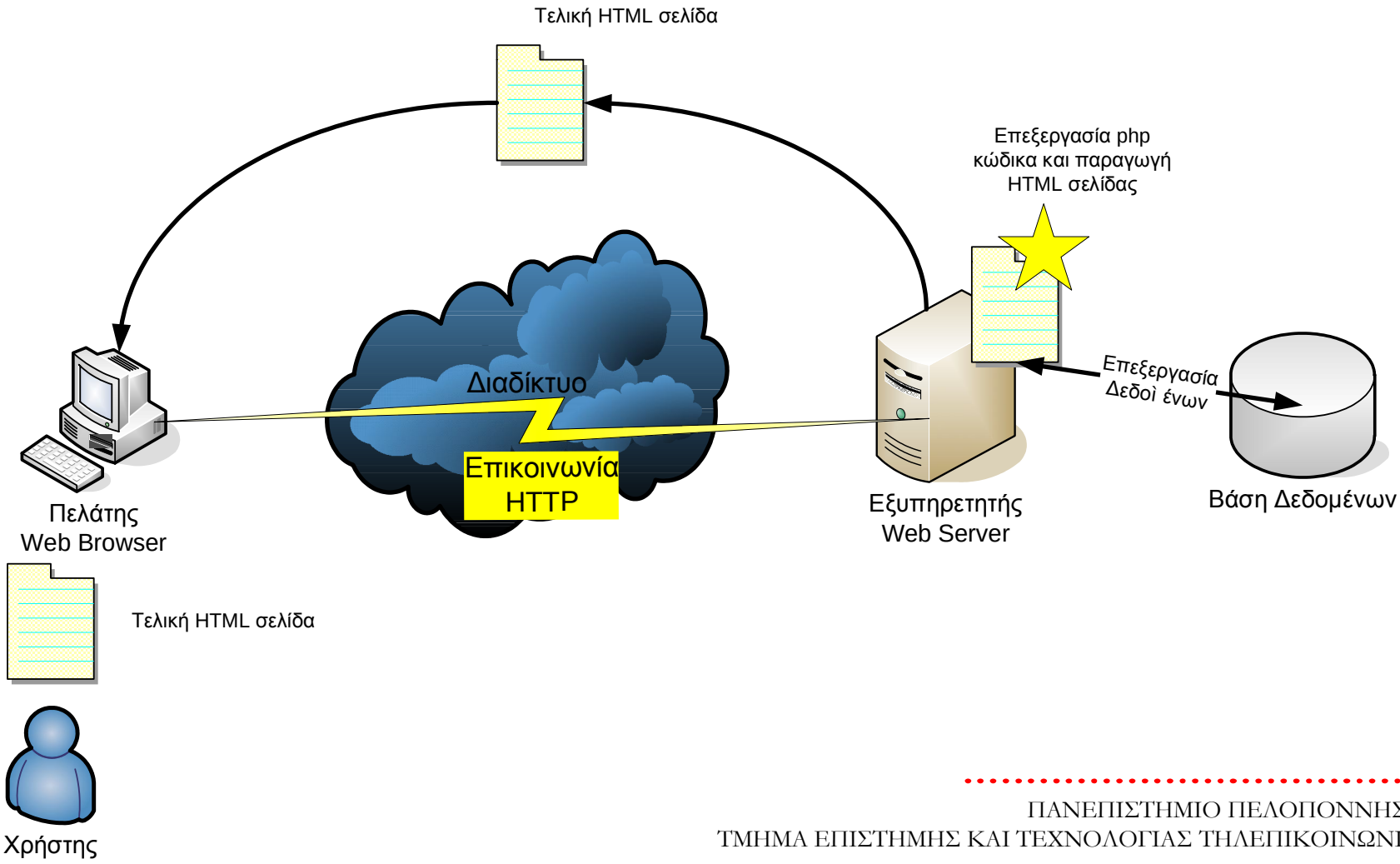
Plug-ins


HTML σελίδα με
κώδικα Plug-ins
Επεξεργασία
HTML σελίδα
με κώδικα
Plug-ins
Διαδίκτυο
Επικοινωνία
HTTP
Εμφάνιση plug-in
στην σχετική
εφαρμογή
Πελάτης
Web Browser
Εξυπηρετητής
Web Server
Κατάλληλη
Εφαρμογή
Τελική HTML σελίδα
Χρήστης

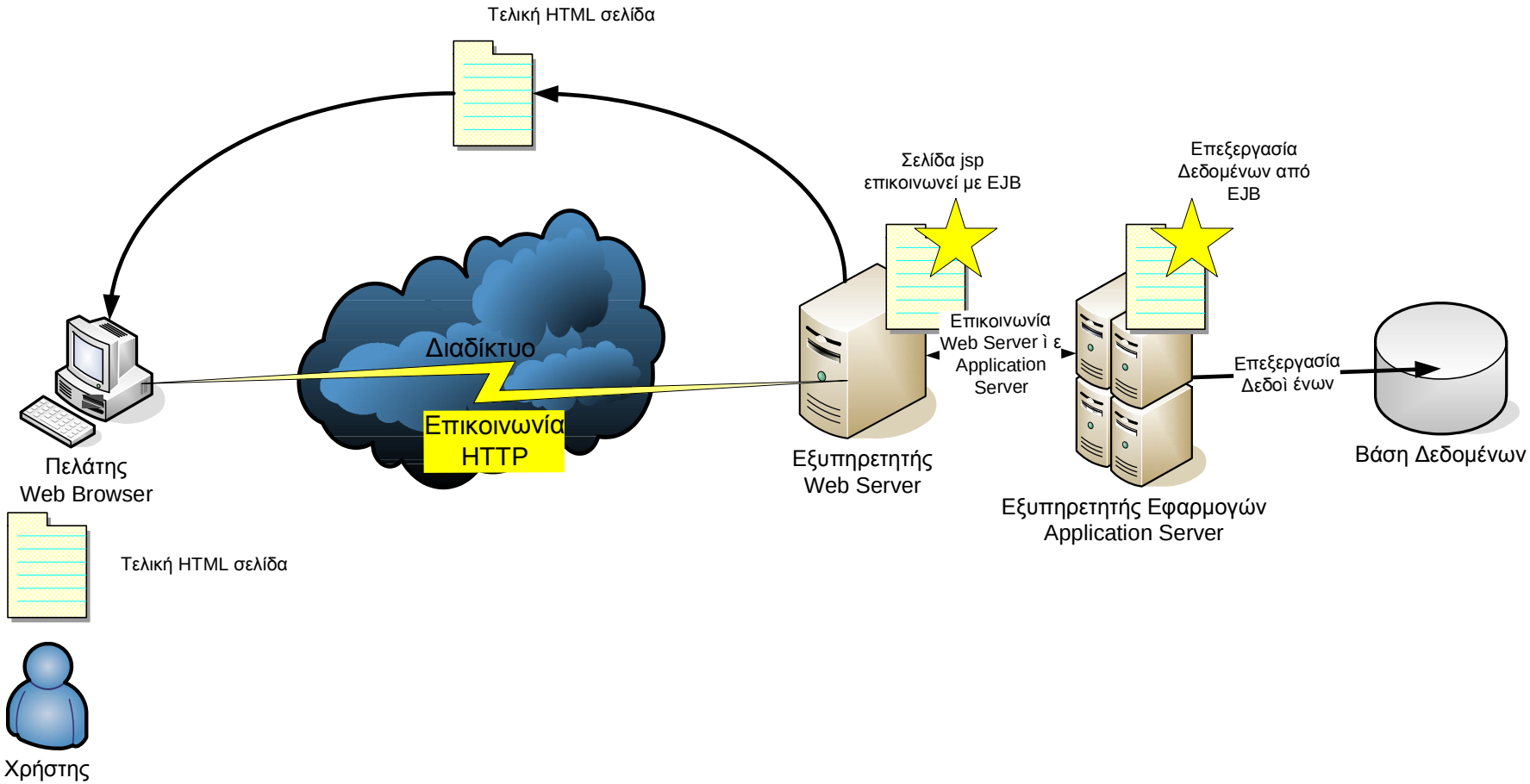
Καταλληλότητα, Πλεονεκτήματα, Μειονεκτήματα

- Κατάλληλη τεχνολογία όπου:
  - Ελέγχεται η είσοδος του χρήστη (form validation)
  - Απαιτούνται πολύπλοκες λειτουργίες στον πελάτη (πχ. παιχνίδια)
- Πλεονεκτήματα:
  - Ταχύτερες λόγο τοπικής εκτέλεσης
  - Δυνατότητα εκτέλεσης χωρίς συνεχή σύνδεση server/client
  - Δυνατότητες, ευελιξία, ευχρηστία και καλαισθησία ιστοσελίδων
  - Οικομικότερες – Καμία εμπλοκή με server
- Μειονεκτήματα:
  - Ο κώδικας είναι ορατός
  - Δεν μπορεί να έχει πρόσβαση στο τοπικό file system
  - Δεν τρέχουν σε όλους τους clients (browsers)

Προγραμματισμός στην πλευρά του εξυπηρετητή (CGI, php, jsp,..)



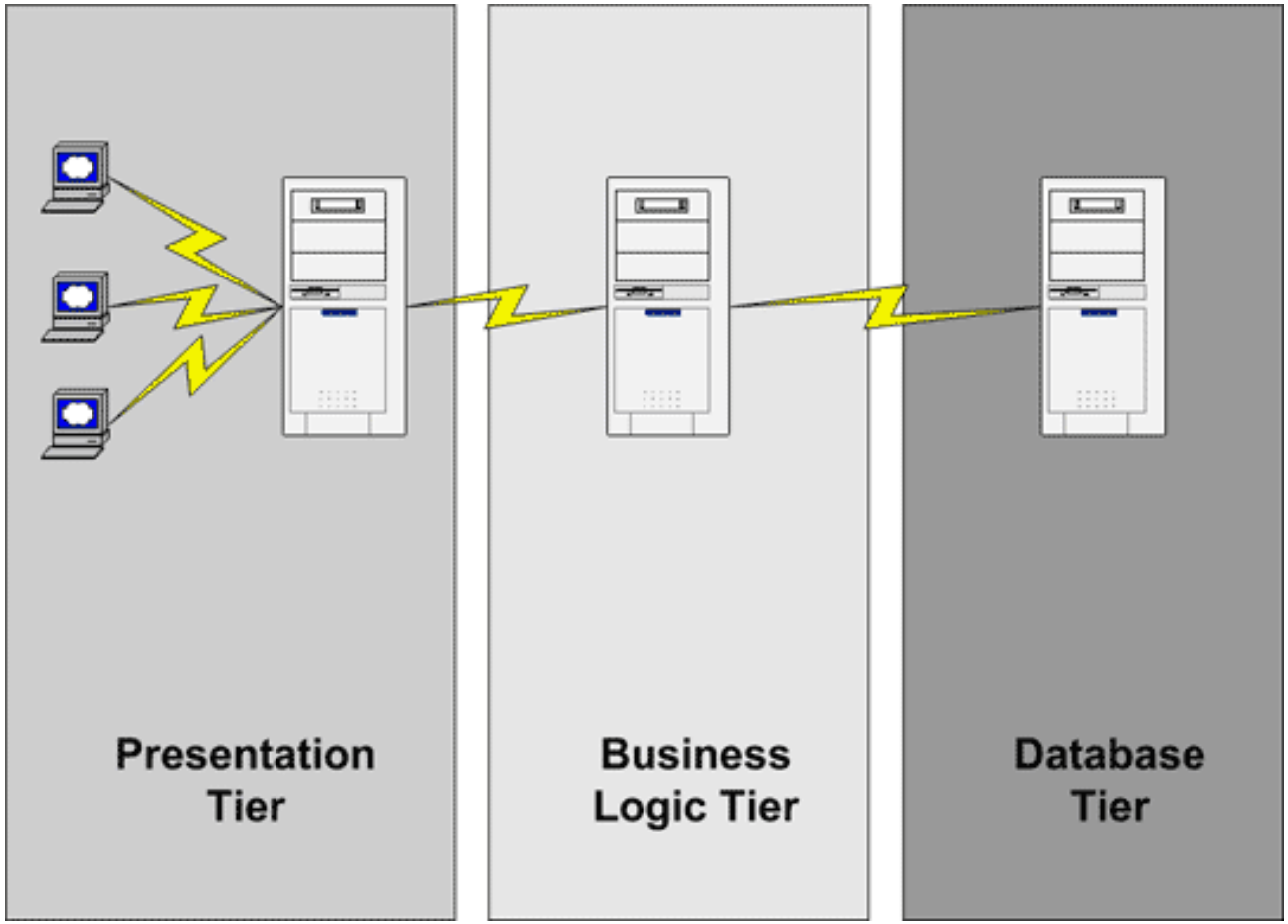
Εφαρμογές πολλών στρωμάτων (n-tier – application servers)



Server Side: Καταλληλότητα, Πλεονεκτήματα, Μειονεκτήματα

- Καταλληλότητα:
  - Δυναμική / Παραμετρική εμφάνιση περιεχομένου
  - Απαραίτητο όταν απαιτείται επικοινωνία (αλληλεπίδραση) με τον Server
  - Δυνατότητα ελέγχου των πελατών, π.χ. μετρητές επισκέψεων (hit counters), ελεγχόμενη πρόσβαση σε κάποιες σελίδες
- Πλεονεκτήματα:
  - Η επεξεργασία μεταφέρεται στο server, χρησιμοποιείται η ισχύς του server
  - Ο κώδικας είναι κρυφός
  - Η εκτέλεση του κώδικα είναι ανεξάρτητη του browser: στέλνεται «καθαρό»
  - HTML που εμφανίζεται πανομοιότυπο σε κάθε browser
  - Η μοναδική λύση για πρόσβαση στο file system του server
- Μειονεκτήματα:
  - Χρησιμοποιεί πολύτιμη επεξεργαστική ισχύ του server.
  - Κλιμάκωση (scalability);

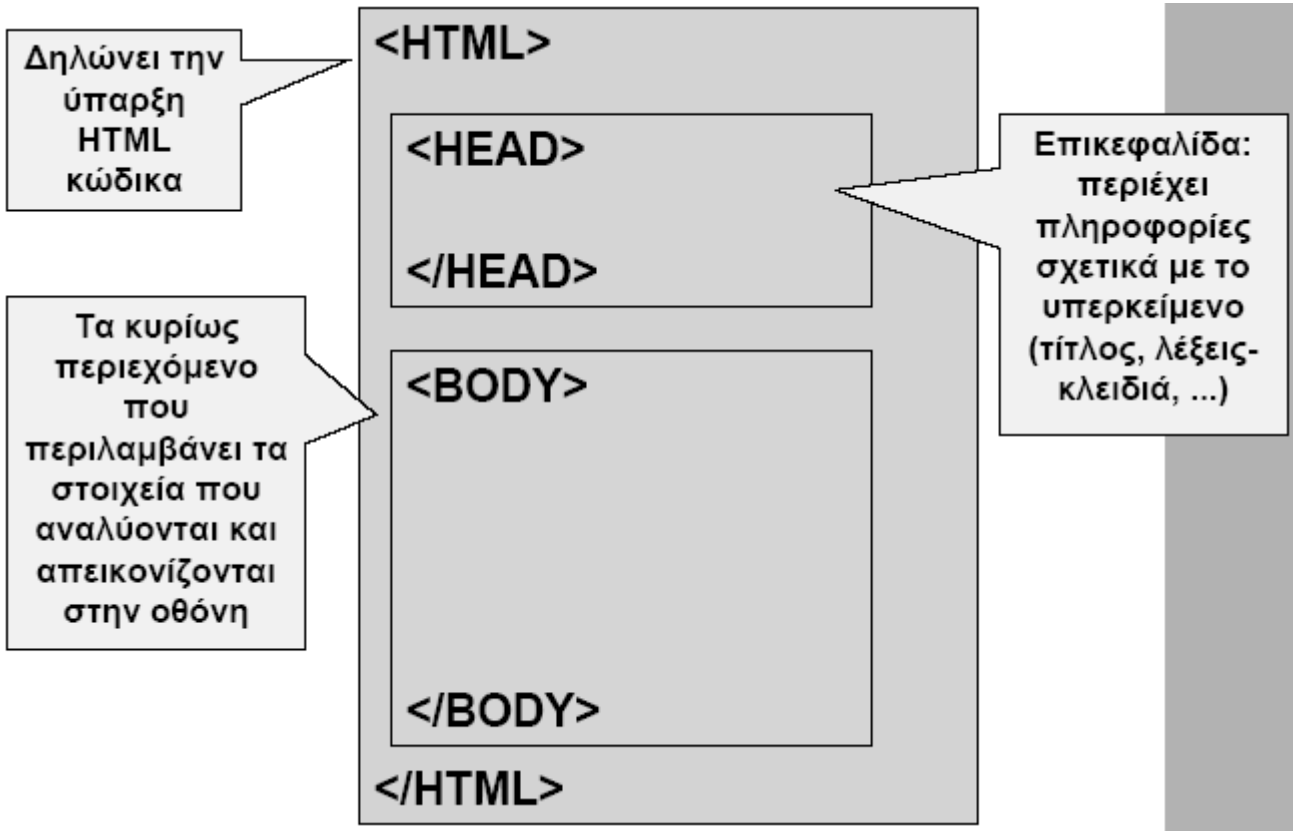
Εφαρμογές πολλών στρωμάτων (n-tier – application servers)



HyperText Markup Language: HTML

- Οι οδηγίες της HTML δίνονται με χρήση των ετικετών (tags). Οι ετικέτες είναι το τμήμα εκείνο του κειμένου που περικλείεται από τα σύμβολα μικρότερο (<) και μεγαλύτερο (>) και την εντολή μέσα στα σύμβολα αυτά που αποτελεί την οδηγία. Οι ετικέτες «ανοίγουν» και «κλείνουν». Π.χ.:
  - This text will be displayed as bold!
- Ένα HTML αρχείο είναι ένα απλό αρχείο κειμένου (text file). Έτσι, δεν χρειάζεται ένα ειδικό επεξεργαστή κειμένου. Αρκεί ένας απλός επεξεργαστής κειμένου όπως είναι το NotePad ή το MS Word
- Υπάρχουν ωστόσο και ειδικά προγράμματα για τη γρήγορη και εύκολη συγγραφή HTML κώδικα (web authoring tools), όπως το Micromedia Dreamweaver (απαιτούν άδεια!) και άλλα τα οποία διατίθενται δωρεάν

Βασική δομή ενός HTML εγγράφου



Cascading Style Sheets

- Καθορίζουν το style, τη δομή και τη θέση των στοιχείων μέσα στο αρχείο
- Κάνουν εφικτό το διαχωρισμό της εμφάνισης των δεδομένων και των ίδιων των δεδομένων
- Σύνταξη
  - selector {property: value; property2: value2}
    - selector : html tag – {body, p, h1,…}
    - property : το χαρακτηριστικό που θέλουμε να αλλάξουμε
      - color, font-family, text-align

Τρόποι εισαγωγής style sheet

- Inline styles
  - Καθορίζει το style για συγκεκριμένα elements
  - Χρήση του attribute “style” μέσα σε tag
  - Μπορεί να καθορίσει πολλά properties
- Internal styles sheets
  - Ορίζεται μέσα στο <head> με το tag <style>
  - Επηρεάζει τα elements στα οποία αναφέρεται
- External style sheets
  - Δημιουργία ξεχωριστού αρχείου css με το οποίο συνδέεται το html αρχείο
  - Ιδανικό όταν το ίδιο style sheet εφαρμόζεται σε περισσότερα από ένα αντικείμενα

Javascript

- Γλώσσα σεναρίου (script language)
- Αναπτύχθηκε από τη Netscape
- Τρέχει σε όλους τους browsers (cross-platform)
- Γράφεται απευθείας μέσα σε HTML έγγραφα με ή χωρίς χρήση ειδικού λογισμικού (HTML authoring tools, script editors)
- Πλεονεκτήματα/Μειονεκτήματα client-side programming
  - Αντίδραση/επεξεργασία στις κινήσεις του χρήστη (user actions)
  - Επεξεργασία/έλεγχος δεδομένων εισόδου του χρήστη (user input)
  - Δυναμική εμφάνιση (συγγραφή) HTML
  - Βελτίωση ευχρηστίας, ευελιξίας πλοήγησης
  - Εύκολη στην εκμάθηση/χρήση – Όχι όμως πλήρης γλώσσα

Χρήση της JavaScript

- Μικρά κομμάτια κώδικα σε ένα HTML αρχείο
  - π.χ. εμφάνιση σειράς αριθμών (1, 2, …, 100)
- Εμφάνιση «δυναμικού» περιεχομένου
- Αντιλαμβάνεται και αντιδρά σε γεγονότα
  - π.χ. ο χρήστης επιλέγει ένα link
- Αλλάζει τα περιεχόμενα σε html elements
- Ελέγχει δεδομένα που δίνονται σε μία φόρμα πριν αυτή γίνει submit

DOM -Document Object Model

- Το DOM είναι μια πλατφόρμα και μια ανεξάρτητη από γλώσσα προγραμματισμού διεπαφή που επιτρέπει στα προγράμματα και τα scripts να έχουν πρόσβαση δυναμικά και να ενημερώνουν το περιεχόμενο, τη δομή, και το ύφος ενός εγγράφου.
- Το HTML DOC αποτελείται από μία δεντρική δομή όπου στοιχεία ενσωματώνονται σε άλλα στοιχεία. Τα στοιχεία αυτά μπορούν να προσπελαστούν από το DOM δέντρο.

PHP

- Υψηλή απόδοση: με ένα φτηνό server μπορούν να εξυπηρετηθούν εκατομμύρια επισκέψεων καθημερινά.
- Συνεργάζεται εύκολα με τους περισσότερους database servers (βάσεις δεδομένων)
- Σημαντικό για χτίσιμο πληροφοριακών συστημάτων (π.χ. Εφαρμογές ηλεκτρονικού εμπορίου)
- Ενσωματωμένες βιβλιοθήκες για συνήθεις web διαδικασίες: δυναμική δημιουργία εικόνων, αποστολή email, χειρισμός cookies
- Χαμηλό κόστος: δωρεάν
- Ευκολία μάθησης και χρήσης: η σύνταξή της βασίζεται σε άλλες γλώσσες (μοιάζει με Java, C)
- Υποστηρίζεται από τους περισσότερους web servers σαν module (επιπρόσθετο δομικό στοιχείο)
- Μεταφερσιμότητα (portability): ο ίδιος κώδικας δουλεύει χωρίς αλλαγές και σε άλλο λειτουργικό σύστημα
- Διαθεσιμότητα του κώδικα προέλευσης (open source): μπορούν να πραγματοποιηθούν αλλαγές στη γλώσσα

Επαναχρησιμοποίηση Κώδικα

- Δυνατότητα για επαναχρησιμοποίηση κώδικα από άλλα αρχεία (php, html, οποιοδήποτε άλλο)
- Συνήθως αρχεία .inc αλλά προσοχή! Ο πηγαίος κώδικας ενός .inc μπορεί να φανεί αν φορτωθεί απευθείας από browser -> καλύτερα να χρησιμοποιούμε .php ή «κρύψιμο» των .inc (αποθήκευση σε κατάλογο που δεν είναι 'δημοσιευμένος'
- require() & include()
- include()-> δεν υπολογίζεται αν η εντολή δεν εκτελεστεί
- Πλεονεκτήματα επαναχρησιμοποίησης κώδικα:
- Μικρότερο «κόστος» (όχι περιττή επανεγγραφή κώδικα)
- Αυξημένη αξιοπιστία (αν ο κώδικας δουλεύει κάπου, δουλεύει με τον ίδιο τρόπο παντού)

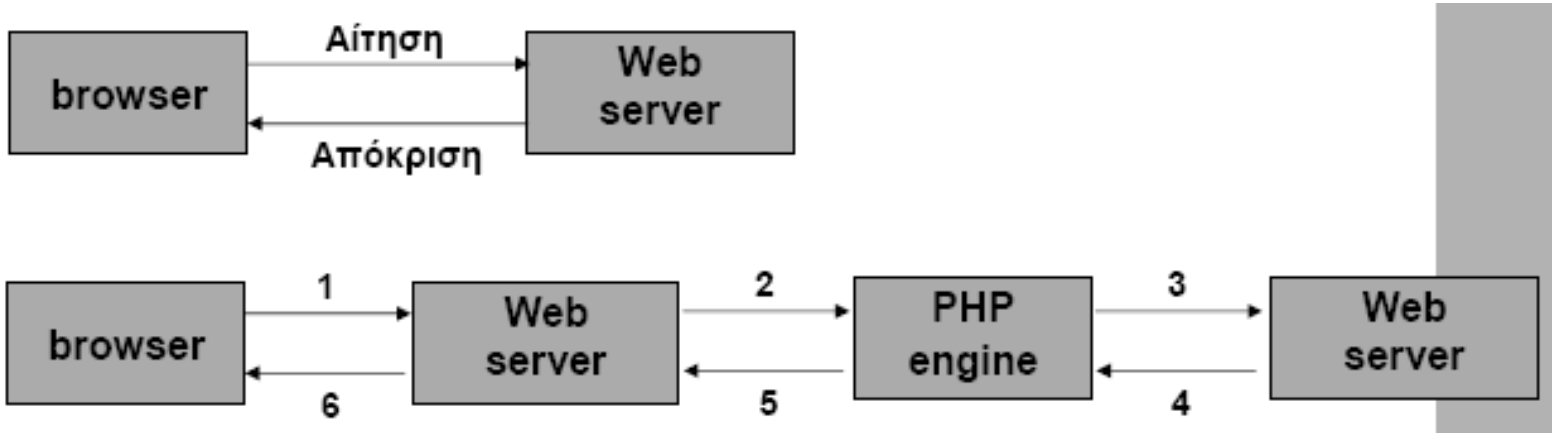
Διαφορά Get – Post στις HTML φόρμες

- Get, Post: Μέθοδοι μεταφοράς δεδομένων μιας φόρμας από τον web client στον web server
  - Post: Αποστολή δεδομένων φόρμας «διαφανώς» από το χρήστη
  - Get:Τα στοιχεία της φόρμας μεταφέρονται ως τμήμα της URL, π.χ. http://www.uop.gr/~gkamas/grade.php?grade=4
- Ερώτηση: πως μπορώ να στείλω δεδομένα μιας φόρμας προς επεξεργασία από ένα PHP script χωρίς να χρειαστεί να ανοίξω τη φόρμα;
- Καμία διαφορά ως προς την επεξεργασία των δεδομένων από το server-side script

Φόρμα

- **$_GET[ ]**
  - Συλλέγει τις τιμές από τη φόρμα ως κείμενο
  - Η πληροφορία που αποστέλλεται από τη φόρμα με τη μέθοδο GET είναι ορατή σε όλους (στο πεδίο διευθύνσεων). Θυμίζουμε ότι η GET περιορίζει την ποσότητα πληροφορίας που μπορεί να σταλεί
- **$_POST[ ]**
  - Χρησιμοποιείται για να συλλεχθούν οι τιμές που έχουν σταλεί από μια φόρμα με την POST
  - Η πληροφορία που στέλνεται από τη φόρμα με αυτή τη μέθοδο είναι αόρατη στον πελάτη. Η μέθοδος POST δεν έχει περιορισμούς ως προς το μέγεθος της πληροφορίας.

Αρχιτεκτονική Web Βάσεων Δεδομένων



1. Αίτηση
2. Ο web server λαμβάνει αίτηση
3. Η PHP engine αρχίζει την ανάλυση του script, συνδέεται στον database server και στέλνει το ερώτημα
4. Ο database server επεξεργάζεται το ερώτημα και στέλνει πίσω τη πληροφορία
5. Μορφοποίηση των αποτελεσμάτων σε HTML και επιστροφή
6. Ο web server επιστρέφει την HTML σελίδα στον browser

Τι είναι Servlet (SERVer appLET)

- Τα Servlets είναι μικρά προγράμματα γραμμένα στη γλώσσα Java που λειτουργούν στον server και επεκτείνουν τις λειτουργίες ενός Web Server.
- Όπως και οι άλλες αντίστοιχες τεχνολογίες (CGIs, ASP, PHP, ..), χρησιμοποιείται για την δημιουργία Web σελίδων που το περιεχόμενό τους δεν είναι στατικό αλλά μπορεί να εξαρτάται από τα δεδομένα που εισαγάγει ο χρήστη και χρειάζεται να ανακτηθεί από βάσεις δεδομένων ή από άλλα συστήματα

Τι είναι Servlet (SERVer appLET)

- Εκτελούνται σε ένα Web Server (Servlets: server-side πρόσωπο της Java – Applets: client-side πρόσωπο της Java, δηλ. εκτελούνται σε Web Browsers)
- Τα Servlets είναι εγκατεστημένα σε Web Server, δέχονται δεδομένα μέσω του πρωτοκόλλου HTTP και απαντούν στέλνοντας στον Web Browser αρχεία τύπου HTML.
- Για να προγραμματίσουμε Servlets είναι απαραίτητο το JSDK (Java Servlet Development Kit) ή Servlets API (Application Programming Interface) που είναι ενσωματωμένο σε αρκετά εργαλεία προγρ/μού Java (π.χ. JDeveloper)
- Το Servlets API αποτελεί πλέον μέρος του JDK (v. 1.4)
- Τα Servlets υποστηρίζονται από (μπορούν να τρέξουν σε) όλους σχεδόν τους Web Servers, π.χ. Apache, Microsoft IIS, Java Web Server, κλπ.

Η δύναμη των Servlet

- Είναι γραμμένα σε γλώσσα Java κατά συνέπεια είναι platform independent: “Write once Serve Everywhere”
- Εκμεταλλεύονται πλήρως το Java API, RMI, CORBA, Database Connectivity.
- Αποδοτικότητα & Αντοχή - Μένουν στην μνήμη μεταξύ διαδοχικών καλεσμάτων
- Κομψότητα (Elegance), Object-Oriented, Clean Code, Modular, Simple
- Λειτουργούν με το πρωτόκολλο HTTP

PHP vs. Servlets

- Τα servlets και τα PHP scripts αποτελούν εναλλακτικές για server-side προγραμματισμό.
- Τα servlets «φορτώνονται» μία φορά και όχι κάθε φορά που καλούνται, αντίθετα με τα PHP scripts
- Τα servlets είναι τεχνολογία που βασίζεται σε μια πλήρη αντικειμενοστραφή γλώσσα (Java), η PHP είναι γλώσσα σεναρίου (script)
- Υπάρχει διαφορά στη λογική: η PHP μοιάζει περισσότερο με την τεχνολογία JSP, ο PHP κώδικας είναι ενσωματωμένος σε HTML κώδικα, το στατικό HTML διακρίνεται από το HTML που παράγει δυναμικά η PHP. Οι servlets αποτελούν κλάσεις Java που όταν εκτελούνται παράγουν HTML κώδικα (στατικό & δυναμικό)

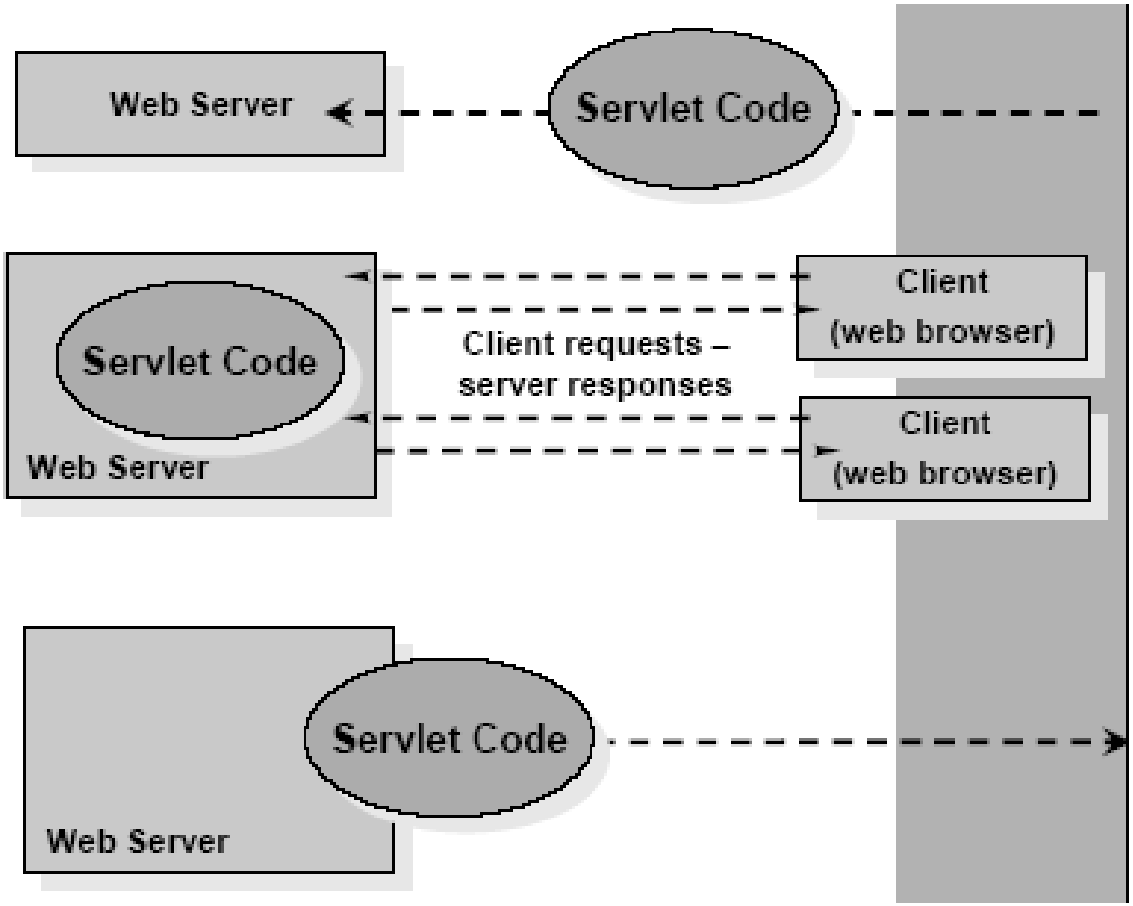
PHP vs. Servlets

- Οι κλήσεις σε «έτοιμες» (built-in) συναρτήσεις της PHP (που περιλαμβάνονται στις βιβλιοθήκες της PHP) είναι συνήθως γρηγορότερες από κλήσεις σε συναρτήσεις των Servlets. Το αντίστροφο όμως ισχύει για τον επιπλέον κώδικα που γράφει ο προγραμματιστής PHP.
- Η PHP προσφέρεται για γρήγορη ανάπτυξη κώδικα λόγω απλής σύνταξης
- Τα servlets προσφέρονται για μεγαλύτερης κλίμακας έργα λόγω της εκμετάλλευσης των πλούσιων βιβλιοθηκών αλλά και της αντικειμενοστραφούς (object-oriented) φύσης της Java.
- Με τους servlets είναι εύκολη η μετάβαση από μια ΒΔ σε άλλη (με αλλαγή λίγων γραμμών κώδικα)

Ο κύκλος ζωής του Servlet

- Κάθε servlet έχει τον ίδιο κύκλο ζωής:
  - Ο server το κάνει load και το αρχικοποιεί: «τρέχει» η μέθοδος init()
  - Το servlet δέχεται μηδέν ή και περισσότερα client requests: «τρέχουν» οι μέθοδοι service() ή doGet()/doPost()
  - Ο server το κάνει remove (ορισμένοι servers εκτελούν αυτό το βήμα μόνο όταν κάνουν shut down): «τρέχει» η μέθοδος destroy()

Ο κύκλος ζωής του Servlet



Τι είναι Servlet Java Server Pages (JSP)

- Η τεχνολογία JavaServer Pages (JSP) είναι server-side τεχνολογία
- Βασίζεται στην τεχνολογία των Java Servlets
- Η κύρια διαφορά τους είναι πως ενώ ο κώδικας των Servlets είναι δομημένος σε κλάσεις που γράφονται, μεταγλωττίζονται και εκτελούνται όπως κάθε κλάση Java, οι JSP σελίδες αποτελούνται από στατικό HTML και δυναμικά παραγόμενο HTML περιεχόμενο τα οποία είναι διαχωρισμένα με ειδικά tags
- Με απλά λόγια, η JSP τοποθετεί Java κώδικα μέσα σε HTML έγγραφα

Τι είναι Servlet Java Server Pages (JSP)

- Μία JSP σελίδα έχει extension *.jsp (αυτό πληροφορεί τον web server ότι η σελίδα πρέπει να φορτωθεί από έναν JSP container για να διερμηνεύσει και να εκτελέσει τον JSP κώδικα
- Ο JSP container διερμηνεύει τα JSP tags και τα scriptlets, παράγει το δυναμικό περιεχόμενο και το στέλνει πίσω στον client ως HTML ή XML σελίδα.

```
<HTML> <BODY>
Hello! The time is now <%= new java.util.Date() %>
</BODY> </HTML>
```

JSP: Πως δουλεύει

- Βήματα που λαμβάνουν χώρα όποτε ζητείται μια JSP σελίδα:

1. Ο web browser κάνει ένα request μέσω Internet για μια *.jsp σελίδα
2. Το request φτάνει στον Web server
3. Ο Web server αναγνωρίζει ότι ζητήθηκε ένα ειδικό αρχείο (*.jsp), και περνάει το JSP αρχείο στο JSP Servlet Engine (application server)
4. Αν το JSP αρχείο ζητείται για πρώτη φορά, τότε διερμηνεύεται (parsing), διαφορετικά η διαδικασία προχωράει στο βήμα 7

JSP: Πως δουλεύει

5. Στη συνέχεια παράγεται ένας Servlet από το JSP αρχείο. Όλο το στατικό HTML που περιέχεται στη JSP σελίδα μετατρέπεται σε 'out.println' εντολές
6. Ο πηγαίος κώδικας του Servlet που προκύπτει μεταγλωττίζεται σε ένα class αρχείο
7. Ο Servlet εκτελείται, καλούνται οι κατάλληλες μέθοδοί
8. Ο HTML κώδικας που προκύπτει από το Servlet output στέλνεται στον web browser
9. Ο HTML κώδικας εμφανίζεται στον web browser του χρήστη

Συμπεράσματα: JSP vs Servlets

- Η JSP δεν προσφέρει κάτι που δεν μπορούμε να επιτύχουμε με Java Servlets, άλλωστε βασίζεται στην τεχνολογία των Servlets
- Η διαφορά έγκειται κυρίως στη διαφορετική προσέγγιση, καθώς στους Servlets ή παραγωγή στατικού ή δυναμικού HTML είναι ενσωματωμένη σε κώδικα Java
- Αντίθετα, στη JSP το στατικό HTML περιεχόμενο είναι ξεκάθαρα διαχωρισμένο από το δυναμικό HTML που παράγεται από JSP (Java) κώδικα. Αυτό συνιστά έναν πιο «καθαρό» σχεδιασμό καθώς διαχωρίζει τον απλό HTML κώδικα από την προγραμματιστική λογική και αφήνει τον web designer να επικεντρωθεί στην παρουσίαση και μορφοποίηση του περιεχομένου (HTML)

Πλεονεκτήματα των JSPs

- Είναι ευκολότερη και ταχύτερη η ανάπτυξή τους, κυρίως για μικρά projects
- Με τους servlets είναι δυσκολότερη η παραγωγή στατικού HTML κώδικα, χρειάζονται πολλές out.println εντολές, π.χ.

```
out.println("<body>");
out.println("<h2>Hello " + user + "</h2>");
```

- Δίνουν έμφαση στην μορφή και παρουσίαση της σελίδας (HTML) και αφήνουν την προγραμματιστική λογική σε Java κώδικα
- Αν έχουμε έτοιμο JSP κώδικα, μπορεί εύκολα να ενσωματωθεί σε HTML σελίδες από web authors που δε γνωρίζουν προγραμματισμό

Μειονεκτήματα των JSPs

- Εύρεση και διόρθωση λαθών (debugging) είναι πολύπλοκη
- Μεγάλες ποσότητες ενσωματωμένου κώδικα σε scriplets μπορεί να οδηγήσουν σε σελίδες που είναι δύσκολο να συντηρηθούν
- Η απόδοση JSPs είναι κατώτερη των servlets (το compilation 200 JSPs θα δημιουργήσει 200 servlets). Αν χρησιμοποιούσαμε τεχνολογία Servlets θα μπορούσαμε να ενσωματώσουμε περισσότερη λειτουργικότητα σε λιγότερους servlets.

Τι είναι η XML;
EXtensible Markup Language

- Υποσύνολο της SGML (Standard Generalized Markup Language)
  - Μετα-γλώσσα → κατάλληλη για τον ορισμό άλλων γλωσσών
- Σχεδιάστηκε για τον ορισμό δεδομένων
  - Οι δομές δεδομένων ανεξάρτητες από την πλατφόρμα
  - Εύκολη η αυτόματη επεξεργασία των δεδομένων
  - Ο χρήστης μπορεί να ορίσει τα δικά του tags
- Δεν περιγράφει τον τρόπο εμφάνισης δεδομένων!
  - Ένα XSL αρχείο ορίζει την εμφάνιση ενός XML αρχείου
- Ένα DTD (Document Type Definition) ή ένα XML Schema ορίζει τη σύνταξη ενός XML αρχείου

Γιατί άλλη μία γλώσσα?

- Η XML χρησιμοποιείται για την ανταλλαγή δεδομένων
  - Επιτρέπει σαφή ορισμό των δεδομένων
  - Όλοι οι συμμετέχοντες «μεταφράζουν» με τον ίδιο τρόπο τα δεδομένα
- Αντικαθιστά το EDI (Electronic Data Interchange)
  - Χρησιμοποιεί το διαδίκτυο για την ανταλλαγή δεδομένων
  - Είναι πιο ευέλικτη
- Επιτρέπει τον ορισμό άλλων γλωσσών
  - WSDL – Web Services Description Language

XML & HTML

- Η XML δεν έχει σκοπό να αντικαταστήσει την HTML, αλλά να την συμπληρώσει
  - Η HTML σχεδιάστηκε για να παρουσιάζει δεδομένα δίνοντας έμφαση στο πώς αυτά φαίνονται
  - Η XML σχεδιάστηκε για να περιγράφει δεδομένα δίνοντας έμφαση στο τι είδος δεδομένα είναι

Web Services (1/2)

- Επιτρέπουν ένα πρόγραμμα σε έναν υπολογιστή να καλέσει μια λειτουργία σε έναν άλλο υπολογιστή χωρίς να δίνουν σημασία στα ακόλουθα:
  - Λειτουργικό σύστημα
  - Γλώσσα προγραμματισμού
  - Κατασκευαστής
  - Τοποθεσία στο Διαδίκτυο

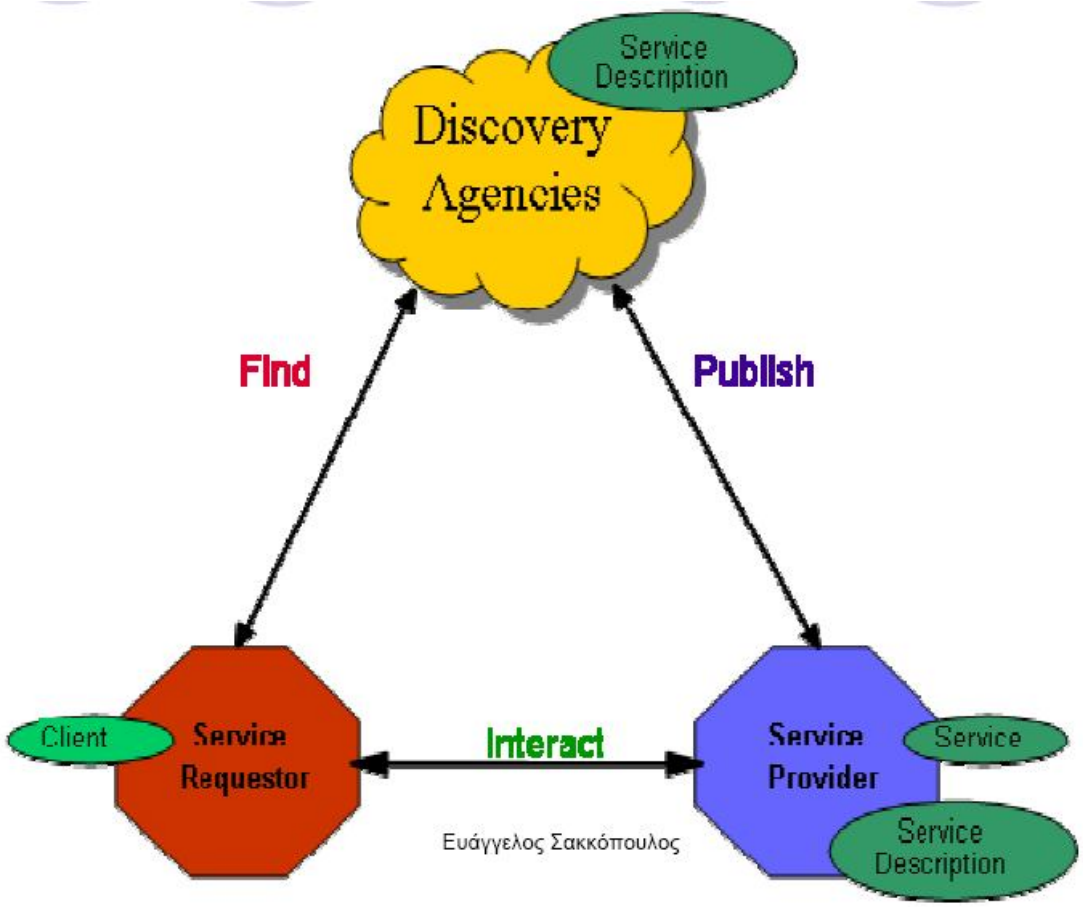
Web Services (2/2)

- Ανεξάρτητες από την αρχιτεκτονική
  - Schema descriptions
  - Discovery standards
- Απλές
  - Η δημιουργία Web Services είναι εύκολη, γρήγορη και απλή
  - Το data schema είναι εύκολα αναγνώσιμο από τον άνθρωπο
  - Χρησιμοποιείται οποιαδήποτε προγραμματιστική γλώσσα
- Interoperable
  - Όλες οι Web Services μιλούν με τα ίδια πρότυπα και κατά συνέπεια μπορούν να επικοινωνούν μεταξύ τους
  - Microsoft, IBM και Sun έχουν συμφωνήσει και χρησιμοποιούν τα ίδια πρότυπα

Αρχιτεκτονική Web Services (1/2)

- Architecture components
  - Service: η υλοποίηση μίας web service
  - Service description: περιέχει τις πληροφορίες της διασύνδεσης και της υλοποίησης της υπηρεσίας
- Roles
  - Service Provider: η πλατφόρμα που φιλοξενεί την υπηρεσία
  - Service Requestor: η εφαρμογή που καλεί μια υπηρεσία
  - Discovery Agency: ένα σύνολο από περιγραφές υπηρεσιών στο οποίο οι services providers δημοσιεύουν τις περιγραφές των υπηρεσιών τους
- Αλληλεπίδραση ανάμεσα στου παραπάνω ρόλους
  - Publish
  - Find
  - Interact

Αρχιτεκτονική Web Services (2/2)



Web Services Stack

- Transport layer
  - HTTP, FTP, SMTP
- Packing layer
  - SOAP (Simple Object Access Protocol)
  - Επιτρέπει στις εφαρμογές να ανταλλάσσουν πληροφορία μέσω του HTTP
- Description layer
  - WSDL (Web Services Description Language)
  - Χρησιμοποιείται για την περιγραφή των Web Services και για το πώς να αποκτηθεί η πρόσβαση σε αυτές
- Discovery layer
  - UDDI (Universal Description Discovery and Integration)

Συμμετρική και Ασύμμετρη κρυπτογραφία

- Συμμετρική (Κλασική) Κρυπτογραφία
  - Το ίδιο κλειδί χρησιμοποιείται για την κρυπτογράφηση και για την αποκρυπτογράφηση δεδομένων
  - Τα συναλλασσόμενα μέρη πρέπει να συμφωνήσουν εκ των προτέρων για το κλειδί που θα χρησιμοποιηθεί
  - Η προστασία του κλειδιού αποτελεί κρίσιμο πρόβλημα
- Ασύμμετρη (Δημόσιου Κλειδιού) Κρυπτογραφία
  - Χρησιμοποιούνται δύο διαφορετικά κλειδιά, ένα ιδιωτικό (μυστικό) και ένα δημόσιο, τα οποία σχετίζονται μεταξύ τους με μονόδρομες συναρτήσεις (one-way functions)
  - Τα δεδομένα που κρυπτογραφούνται με το ένα κλειδί, αποκρυπτογραφούνται αποκλειστικά με το άλλο
  - Μόνο μία φυσική οντότητα γνωρίζει το ιδιωτικό κλειδί, ενώ το δημόσιο κλειδί είναι διαθέσιμο στο κοινό.

Υβριδική Κρυπτογραφία

- Η ασύμμετρη κρυπτογραφία είναι μη αποτελεσματική για την κρυπτογράφηση μεγάλου όγκου δεδομένων, αντίθετα από τη συμμετρική.
- Συνηθισμένη χρήση της ασύμμετρης κρυπτογραφίας είναι η αποστολή ενός συμμετρικού κρυπτογραφικού κλειδιού μέσω ενός ανασφαλούς καναλιού.
- Ένα 'Κέντρο Διανομής Κλειδιών' διανέμει με ασφάλεια στα συναλλασσόμενα μέρη ένα συμμετρικό κλειδί, κρυπτογραφημένο με τα δημόσια κλειδιά των εμπλεκομένων.
- Οι συναλλασσόμενοι αποκρυπτογραφούν το κλειδί και ξεκινούν εμπιστευτικές συνόδους μεταξύ τους, χρησιμοποιώντας συμμετρικούς αλγόριθμους
- Ο συνδυασμός των δύο τεχνολογιών ονομάζεται Υβριδική Κρυπτογραφία. Π.χ. πρωτόκολλο SSL.

Πλεονεκτήματα της Κρυπτογραφίας Δημόσιου Κλειδιού

- Τα δημόσια κλειδιά δεν χρήζουν προστασίας
- Τα ιδιωτικά κλειδιά δεν γνωρίζονται η διανέμονται σε τρίτους σε καμία περίπτωση
  - Για να σταλεί ένα εμπιστευτικό μήνυμα, χρησιμοποιείται το δημόσιο κλειδί του παραλήπτη. Μόνο το ιδιωτικό κλειδί που κατέχει ο παραλήπτης μπορεί να το αποκρυπτογραφήσει
  - Για να υπογραφεί ένα μήνυμα χρησιμοποιείται το ιδιωτικό κλειδί του αποστολέα. Οποιοσδήποτε τρίτος μπορεί να επαληθεύσει την υπογραφή με το δημόσιο κλειδί του αποστολέα
- Ελαχιστοποίηση της διαχείρισης κλειδιών – Δεν χρειάζεται κέντρο διανομής κλειδιών.
- Μεγάλος κύκλος ζωής των κλειδιών
- Δίνουν τη δυνατότητα επαλήθευσης της ακεραιότητας δεδομένων

Προβλήματα της Κρυπτογραφίας Δημόσιου Κλειδιού

- Πως επαληθεύεται η ταυτότητα του κατόχου ενός ζεύγους κλειδιών;
- Πως διασφαλίζεται η ιδιωτικότητα και η ακεραιότητα των κλειδιών κατά τη δημιουργία και τη χρήση τους;
- Πως διανέμονται στο κοινό τα δημόσια κλειδιά έτσι ώστε να διασφαλίζεται η σύνδεση τους με μία φυσική οντότητα;
- Πως τελειώνει ο κύκλος ζωής τους όταν αυτό κριθεί αναγκαίο;
- Διαφαίνεται η ανάγκη ύπαρξης μίας 'Εμπιστης Τρίτης Οντότητας' που διαχειρίζεται 'Ψηφιακά Πιστοποιητικά'.

Ψηφιακά Πιστοποιητικά

- Βεβαιώνουν την ακεραιότητα του Δημόσιου κλειδιού.
- Βεβαιώνουν τη σύνδεση ενός δημόσιου κλειδιού με ένα άτομο ή οργανισμό μέσω της Έμπιστης Τρίτης Οντότητας (Trusted Third Party).
- Ανάλογα με την Αρχή Πιστοποίησης το πιστοποιητικό έχει και διαφορετικό εύρος αναγνώρισης. Συνήθως υπάρχει ιεραρχία πιστοποίησης που ορίζεται από το X.509.
- Από τι αποτελείται ένα ψηφιακό πιστοποιητικό:
  - Κάποια πληροφοριακά στοιχεία για το χρήστη του
  - Το δημόσιο κλειδί του χρήστη
  - Το όνομα μιας Αρχής Πιστοποίησης
  - Την ψηφιακή υπογραφή της Αρχής Πιστοποίησης

Χαρακτήρας θεμάτων γραπτής εξέτασης

- Θεωρητικές ερωτήσεις που θα εξετάζουν την κατανόηση των θεμάτων τα οποία παρουσιάστηκαν
- Πιθανόν κάποιες ερωτήσεις πολλαπλής επιλογής
- 'Ετοιμος' κώδικας που θα πρέπει να εξηγήσετε τη λειτουργία του (πιθανόν να σας ζητηθεί να 'ζωγραφίσετε' το αποτέλεσμα της εκτέλεσής του)
- 'Ετοιμος' κώδικας που θα πρέπει να εξηγήσετε τυχών λανθασμένα σημεία
- Συγγραφή κώδικα με βάση κάποιες προδιαγραφές που θα σας δοθούν (Δεν χρειάζεται να αποστηθίσετε στοιχεία σχετικά με τη σύνταξη καμίας από τις τεχνολογίες αλλά θα σας δοθεί η σύνταξη της τεχνολογίας που θα ζητείται να γράψετε κώδικα)

Παράδειγμα XML: XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="cdcatalog.xsl"?>
<catalog>
        <cd>
                <title>Empire Burlesque</title>
                <artist>Bob Dylan</artist>
                <country>USA</country>
                <company>Columbia</company>
                <price>10.90</price> <year>1985</year>
        </cd>
</catalog>
```

Παράδειγμα XML: XSLT

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
        <xsl:template match="/">
        <html> <body>
        <h2>My CD Collection</h2>
        <table >
          <tr><th>Title</th> <th>Artist</th> </tr>
          <xsl:for-each select="catalog/cd"> <tr>
                  <td><xsl:value-of select="title"/></td>
                  <td><xsl:value-of select="artist"/></td> </tr>
          </xsl:for-each>
        </table>
        </body> </html>
        </xsl:template>
</xsl:stylesheet>
```

Παράδειγμα XML: XHTML

```
<html> <body>
    <h2>My CD Collection</h2>
    <table >
      <tr><th>Title</th> <th>Artist</th> </tr>
      <tr>  <td>Empire Burlesque</td>
            <td>Bob Dylan</td> </tr>
      </table>
    </body> </html>
```

Παράδειγμα php

```
<HTML><HEAD><TITLE>Sum calculator...</TITLE>
</HEAD>
<BODY>
<FORM name=form1
action="number_sum.php" method=get>
<P>Number 1: <INPUT name=number1> </P>
<P>Number2: <INPUT name=number2> </P>
<P><INPUT type=submit value=Submit name=Submit>
    </P></FORM></BODY></HTML>
```

Παράδειγμα php

```
<html>
<head>
<title>Calculated sum...</title>
</head>
<body>
<?
function AddNumbers($nu1,$nu2)
{
 $result=$nu1+$nu2;
}
AddNumbers($_POST[number1], $_POST[number2]);
echo "<H1>" . $_POST[number1] . " + " . $_POST[number2] . " = " . $result .  "</H1>";
?>
</body>
</html>
```

Παράδειγμα php

```
<html>
<head>
<title>Calculated sum...</title>
</head>
<body>
<?
function AddNumbers($nu1,$nu2)
{
 $result=$nu1+$nu2;
 return $result;
}
$result = AddNumbers($_GET[number1], $_GET[number2]);
echo "<H1>" . $_GET[number1] . " + " . $_GET[number2] . " = " . $result .  "</H1>";
?>
</body>
</html>
```