

# Università degli Studi di Torino

## Tecnologie Web (6 cfu) – MFN0634

### Consegna 03 – PHP

#### Testo del compito

Per questo esercizio dovrete scrivere il codice PHP delle pagine di recensioni simili a quelle realizzate in TMNT per la consegna precedente. Il codice ti dovrà permettere di creare le recensioni per una varietà di pagine, ma usando sempre lo stesso codice:

- `movie.php`, contiene il codice PHP per la creazione delle pagine di recensione dei film;
- `movie.css` – il foglio di stile per `movie.php`

Non ci si aspetta che riproduciate perfettamente la stessa pagina come quella mostrata in figura. Ci si aspetta che invece produciate una pagina seguendo le indicazioni per lo stile fornite in questo documento e che queste si presentino il più simile possibile per aspetto, layout e comportamento con quelle qui spiegate.

La pagina risulta uguale a quella che avete realizzato per l'esercizio precedente. La differenza è che la pagina può mostrare recensioni e informazioni per altri film oltre a TMNT. Per esempio il seguente screenshot mostra The Princess Bride:

Per rendere le cose più semplici, il titolo delle pagine al momento deve essere sempre 'Rancid Tomatoes'.

Dal sito moodle del nostro corso dovete scaricare un archivio zip come file di input per i film descritti nella prossima pagina. Decomprimete l'archivio nella stessa cartella **lab03** dove si trova il vostro file **movie.php**, così che potrete accedere ai file contenenti le informazioni dei film usando i cammini relativi (ad esempio: **tmnt2/info.txt** o **princessbride/overview.png**). Il vostro



codice deve assumere che questo è il path da usare. (Non inserire questi elementi in una cartella come **moviefiles/folder**. Inserite ogni cartella del film, come **tmnt2/or mortalkombat/** direttamente nella cartella **lab03**).

Le pagine usano le stesse immagini usate in lab02 con due piccole modifiche. La pagina **tmtn.html** mostra l'immagine **rottenbig.png** vicino a quella del rating 33%. Considerate che alcuni film hanno un rating migliore. Ogni film che ha rating di 60% o più deve mostrare l'immagine **freshbig.png**, mostrata qui a destra. L'immagine **overview.png** deve essere spostata, così che ogni film abbia la propria immagine di overview. Inserite un link per tutte le immagini 'fisse' usando un URL assoluto e non relativo.

Le risorse raggiungibili tramite URL assoluto sono raggiungibili, al solito, qui: <http://courses.cs.washington.edu/courses/cse190m/11sp/homework/2/>

Le immagini di overview costituiscono un'eccezione; provenendo dal file zip ed essendo mantenute nella cartella contenente le informazioni dei film, inserite un link a queste immagini usando un cammino relativo come **'tmtn2/overview.png'**.



Tutti gli elementi di stile non descritti qui sono stati specificati in lab02 o sono soggetti alle scelte di preferenza dal

browser. Gli screenshot presenti in questo documento sono stati realizzati su macchina Windows utilizzando l'ultima versione di FireFox: questo potrebbe rendere la presentazione differente da quella visibile sul vostro sistema.

### **Dettagli sull'aspetto:**

La pagina **movie.php** dovrà mostrare le recensioni dei film basate su una query parametrica chiamata film che è passata dal browser alla pagina mediante URL. Il valore di questa variabile dovrà essere una stringa che contiene il film da mostrare. Il browser richiederà la pagina con un URL come il seguente:

**`http://localhost:8080/tweb/lab03/movie.php?film=princessbride`**

Il codice PHP contenuto nel file movie.php può salvare il parametro in una variabile usando il codice come da esempio:

```
$movie=$_GET["film"];
```

Tutto il codice PHP deve usare questo valore parametrico, non dovrete mai modificare il nome del singolo film. Dovreste assumere che il film esiste e che corrisponde a un dato di input e un'immagine validi nella cartella corrispondente. Non dovrete al momento gestire i casi di film mancanti o di parametro vuoto, film non trovato etc.

Basandovi sul valore del nome del film assegnato alla variabile, dovrete mostrare la recensione di tale film. Ogni film è salvato in una directory chiamata allo stesso modo della query. Per esempio, se il film è **princessbride** questo sarà salvato in una cartella chiamata **princessbride/**.

I file associati a ogni film sono i seguenti. Dovrete assumere che i file esistono e che siano validi:

- **info.txt**, un film contenete tre righe di informazioni sul film: il titolo, l'anno e la percentuale generale di rating. Le informazioni sono usate come intestazione della pagina e il numero di rating è rosso. Per esempio:

The Princess Bride

1987

95

- **overview.txt**, un file che contiene le informazioni da inserire nella sezione General Overview della pagina. Ogni coppia di righe vicine contiene un titolo e un valore per item di informazione da mostrare come una lista con definizione (dt) con la sua descrizione (dd). Il numero di righe del file varia da film a film. Esempio:

STARRING: Cary Elwes, Robin Wright, Andre the Giant, Mandy Patinkin

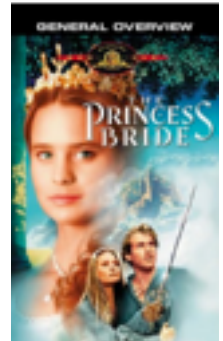
DIRECTOR: Rob Reiner

PRODUCER: Arnold Scheinman, Rob Reiner

SCREENWRITER: William Goldman

...

- **overview.png**, un'immagine da mostrare in alto nella sezione General Overview. Rappresenta la locandina del film, come si può notare dall'immagine sotto.



- **review1.txt**, **review2.txt**, ... i file che contengono le recensioni di ogni film. Ogni recensione contiene esattamente quattro righe: la recensione, un fresh/rotten rating, il nome del recensore e la pubblicazione. Per esempio, **review1.txt** potrebbe contenere:

One of Reiner's most entertaining films, effective as a...

FRESH

Emanuel Levy

[emanuellevy.com](http://emanuellevy.com)

- Film diversi hanno numero di recensioni diverse. Mostrate la prima metà delle recensioni nella colonna di sinistra, il resto a destra. Se un film ha un numero dispari di recensioni, le recensioni in più vanno a sinistra. Se un film ha solo una recensione, questa va a

sinistra. Non vi preoccupate se l'overview e se la sezione delle recensioni hanno un'altezza diversa. (Dovete assumere che il film abbia al minimo una recensione, ma al massimo 10; dovete mostrarle tutte). La barra verde in basso alla pagina deve mostrare anche "(1-N) of N" dove N è il numero di recensioni che il film ha ricevuto.

Se la recensione è composta di più di 10 recensioni (come in `tmtn2`), il nome sarà `review01.txt`... non manomettete il nome del file come `'review1.txt'`; considerate che ogni file deve essere nominato con "review" all'inizio ed estensione `txt`.

- L'immagine da mostrare per ogni recensione è definita dalla seconda riga presente nel file `txt`. Se i critici hanno dato al film un rating FRESH, verrà mostrata **fresh.gif**; se ROTTEN verrà mostrata **rotten.gif**.

Per esempio, se la variabile `film` riceve il valore "princessbride", aprite il file **`princessbride/info.txt`** per leggere titolo/anno/etc. e mostrate le informazioni sulla pagina. Aprite poi **`princessbride/overview.txt`** e mostrate il contenuto nella sezione di General Overview sul lato destro. Dopo accedete alle recensioni presenti in `princessbride/` e mostratele una per una nella sezione delle recensioni.

### **Parte creativa: il tuo film**

Come parte "creativa" del compito, vi chiediamo di creare un vostro insieme personale di dati di input per alimentare la pagina **`movie.php`**. Preparate un file **`info.txt`**, **`overview.txt`** e almeno 4 file di recensioni per il tuo film. Cercate un'immagine adatta da usare come **`overview.png`** delle dimensioni di 250x412px. Potete ridimensionare l'immagine usando GIMP o un altro editor per immagini.

### **Strategie di sviluppo e consigli**

È difficile fare il **debug** di PHP se ci sono degli errori. Scrivete il programma incrementalmente, aggiungendo piccole porzioni di codice alla pagina funzionante, e non avanzate finché non avete testato il nuovo codice aggiunto. Vi consigliamo di mostrare sulla vostra pagina un solo film (come in `princessbride` o `tmtn2`), poi provate a generalizzare la pagina in modo da visualizzare le informazioni di ogni altro film. Niente nel vostro codice deve

essere riferito ad uno specifico film come in `princessbride`. Le seguenti funzioni potrebbero esservi di aiuto:

- `count` – restituisce in numero degli elementi in un array
- `explode` – spezza una stringa in un array di stringhe più piccole delimitate.
- `file` – legge la riga di un file e la restituisce come un array
- `glob` – dato il path di un file o uan wildcard come `"foo/bar/*.jpg"` restituisce un array che si combina con il nome del file;
- `list` – scompone un array in un set di variabile; è utile per operazioni con un array di dati a dimensione fissa;
- `trim` – rimuove gli spazi bianchi all'inizio/fine della stringa.

Nonostante questi consigli, il vostro codice conterrà probabilmente degli errori durante lo sviluppo. Identificate quindi il prima possibile il log file degli errori php (ad esempio `/Applications/MAMP/logs/php_error.log` nella distribuzione MAMP), e controllate periodicamente se ci sono nuove linee in fondo al file. Ad esempio, da terminale potete usare il comando:

```
> tail php_error.log
```

per vedere la data delle ultime righe del file in questione.

Una configurazione ancora più professionale prevede l'installazione del modulo di XDebug congiuntamente al vostro IDE e quindi usare il debugging vero e proprio direttamente dall'ambiente di sviluppo. Questo vi salverà molte ore di sonno, anche se inizialmente dovrete perdere più tempo a configurare il vostro sistema, ma comunque ricordate che il monitoraggio del file di log vi permette di trovare buona parte degli errori: localizzatlo quindi il prima possibile nel vostro sistema!

## **Implementazione e valutazione**

Consegnate (caricandolo sulla vostra cartella gitlab2) i file **movie.css** e **movie.php**. Se realizzate anche la parte "creativa", caricate anche la cartella con i dati del "vostro" film. Non c'è

bisogno carichiate il resto, onde evitare di appesantire il nostro hard disk :)

Per fare tutto correttamente, le pagine per ogni film devono superare il test del validatore HTML W3C. (Non il codice PHP, ma solo il codice generato in HTML deve essere validato). Non usate tabelle HTML per questo esercizio.

Il vostro codice PHP non deve generare errori o warnings. Inoltre, sarà valutato lo stile del vostro codice PHP. Riducete al minimo l'uso della keyword global, usate la corretta indentazione e spaziatura e evitate righe più lunghe di 100 caratteri. Usate il materiale a disposizione dalle esercitazioni precedenti e dai primi 5 capitoli del libro di testo.

Fate attenzione alle modalità di passaggio dal codice PHP e dal tag HTML usando `<?php` e `?>`. Usa il più possibile HTML passando alla modalità PHP solo per i calcoli e per i contenuti dinamici. Per fare tutto perfettamente, dovete ridurre al minimo il numero di parti di codice in PHP inseriti all'interno del codice HTML. Quando possibile, sostituite alcune parti con funzioni chiamate in HTML e dichiarate altrove nel vostro file.

Inoltre, usate il meno possibile i print e gli echo. Inserite come possibili contenuti dinamici nella pagina usando espressioni blocco `<?=` e `?>` in PHP. In generale è considerato un errore l'uso di stampa di HTML realizzato nel seguente modo:

```
print "<p> I am adding content to the page! $x $y $z </p>";    #bad
```

Un altro parametro di auto-valutazione è la **ridondanza**. Non devete avere codice associato ad un particolare film o usare if/else per scegliere quale film mostrare. Usate i loop, le variabili e i commenti per ogni sezione PHP.

Un altro parametro di auto-valutazione sono i **commenti**. Ci dobbiamo sempre aspettare di trovare molti commenti nel codice. Inserite un header descrittivo (nome, corso, descrizione) all'inizio del vostro codice e commenti per ogni sezione di PHP.

Formattate il vostro codice HTML e PHP come negli esempi svolti in classe. Usate in modo appropriato gli spazi bianchi e l'indentazione. Non inserite più di un blocco di elementi per linea e iniziate un nuovo blocco dopo 100 caratteri. Il CSS non è la parte più rilevante dell'esercitazione, ma anche il CSS deve essere validato e non deve essere introdotto codice non valido.

Un altro parametro di auto-valutazione è il corretto **push** del vostro sito sul server gitlab2 nella sottocartella **lab03** del progetto tweb che avete creato precedentemente, in modo tale che sia accessibile e navigabile tramite il percorso:

### **tweb/lab03**

In fase di consegna, due cose molto importanti sono anche le seguenti:

1. Dovete consegnare su **moodle** soltanto la url del vostro clone (nel mio caso, per esempio: <https://gitlab2.educ.di.unito.it/ruffo/tweb.git>). Su moodle solo il testo della url, no html, no commenti. NIENTE. Solo la url che consenta l'accesso al vostro sito.
2. rendete **pubblico** il vostro progetto tweb, così noi possiamo scaricarlo e/o clonarlo nelle nostre macchine locali
3. Mentre ci siete, create anche le cartelle **lab01** e **lab02** così potete fare l'upload dei vostri vecchi esercizi (anche se non li avete consegnati prima, ma almeno nel repo resta traccia di tutto)