MODULO 2
PARTE 5.a

Programmare sul Web 2.0
Tagging systems and tag clouds

Cos'è il Web 2.0 - I

- **Web 2.0** = termine introdotto per la prima volta nel 2004 come titolo di una conferenza promossa dalla casa editrice O'Reilly
- L'idea è che ci si stia avviando verso una **nuova concezione** del web ("versione" 2.0), in contapposizione con la "vecchia" concezione ("versione" 1.0)
- Concetto **confuso e sfaccettato**… è difficile darne una definizione, generalmente si indicano semplicemente una serie di concetti emergenti
- La maggior parte dei **siti attualmente online** (es. Google, Yahoo, eBay, Amazon, Napster, del.icio.us, Wikipedia, ecc.) possono essere considerati a pieno titolo **Web 2.0**

Cos'è il Web 2.0 - II

- I siti diventano **servizi interattivi** (non più semplici collezioni di pagine) → la **navigazione** di un sito diventa un'**esperienza interattiva** estremamente più ricca
- Sito web = **aggregazione**, ri-combinazione di contenuti eterogenei e sempre aggiornati
- Cambia il modo di definire i **programmi**: non sono più "prodotti", ma "**servizi**"
- È possibile usufruire di **applicazioni web** che si sostituiscono a quelle installate sul proprio computer
- ***Mashup*** ("rimescolamento"): applicazioni Web che riutilizzano funzionalità e contenuti offerti in rete per creare servizi e contenuti nuovi
- ***Open API*** (servizi "aperti"): servizi e funzionalità sono messe a disposizione sulla rete (per es. Google Maps)

Cos'è il Web 2.0 - III

- **Web come "partecipazione"**
  gli utenti non sono più semplici lettori, ma diventano **autori di contenuti**: sono i navigatori a generare i contenuti e a destare l'attenzione su ciò che ritengono più interessante
  → es: **blog** e **communities** (es. *Facebook*)
- **Markup (semantico)**
  utilizzo di linguaggi basati su XML, in cui i tag "descrivono" il contenuto → es: **RSS** (formato, basato su **XML**, per la **distribuzione di contenuti** sul web)
- Partecipazione + markup = ***social tagging***
  gli utenti possono contribuire al markup…
  → es: *del.icio.us*, *Flickr*, ecc.

Web 2.0: social tagging - I

Ma che cos'è il *social tagging*?

- ***Tag*** = "etichetta" (formata da una o più parole) collegata ad un contenuto
- ***Metadato*** = "dato sui dati" = informazione che descrive il contenuto (es. autore di un documento, data dell'ultima modifica, …)

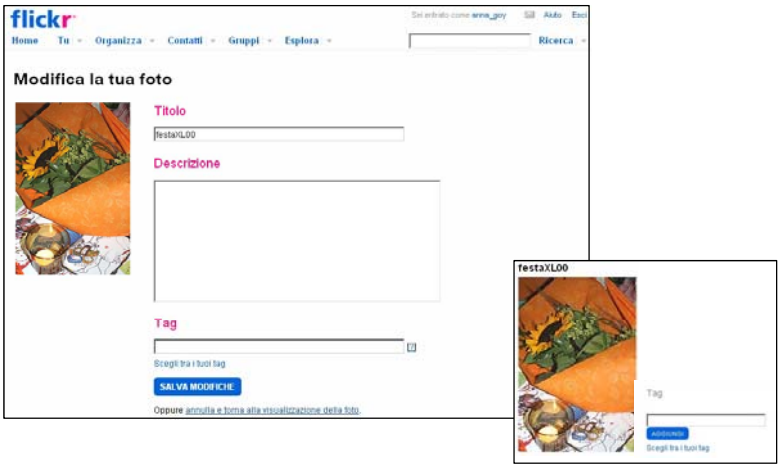
⇒ i **tag** possono essere considerati **metadati**

Sul Web 2.0 gli utenti (*social*) possono assegnare dei tag a dei contenuti (*tagging*); per **esempio**:

- *del.icio.us*: gli utenti possono assegnare dei tag a dei segnalibri (preferiti)
- *Flickr*: gli utenti possono assegnare dei tag a delle immagini

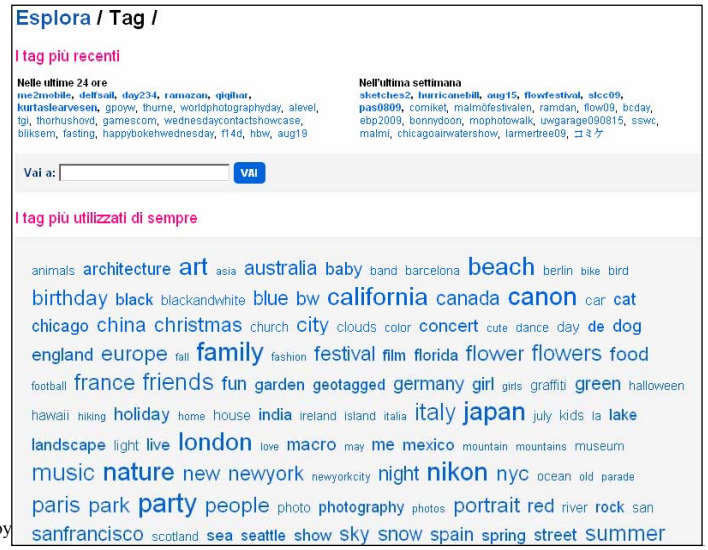
Web 2.0: social tagging - II

Esempio: **www.flickr.com – inserimento tag**



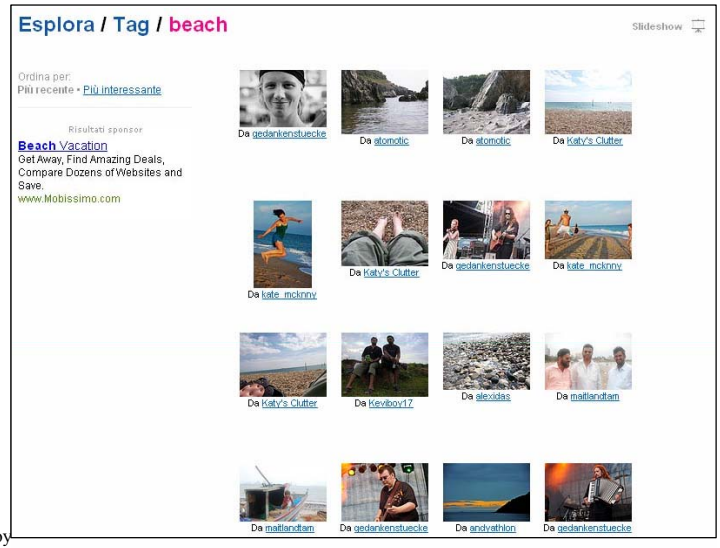
Web 2.0: social tagging - III

Esempio: **www.flickr.com – ricerca per tag**



Web 2.0: social tagging - IV

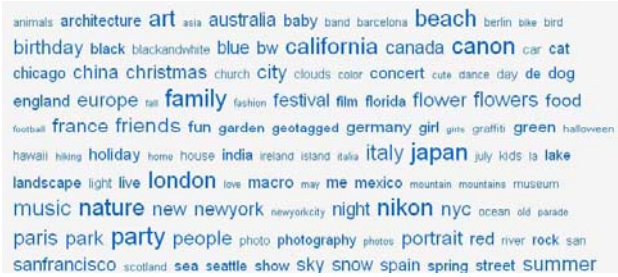
Esempio: **www.flickr.com – click on *beach* tag**



Web 2.0: social tagging - V

Generalmente i tag inseriti dagli utenti vengono mostrati in una ***tag cloud***, utilizzata per **navigare** e trovare i contenuti desiderati all'interno del sito: cliccando su un tag (link), verranno mostrati tutti i contenuti a cui è associato quel tag

NB generalmente, in una *tag cloud* i tag più **popolari** sono visualizzati con **caratteri più grandi**…



Web 2.0: social tagging - VI

L'insieme dai tag assegnati dagli utenti ai contenuti di un sito rappresenta una **categorizzazione "sociale"**, che esprime **il punto di vista degli utenti**, il modo in cui gli utenti classificano quei contenuti

Per esempio, in un archivio di immagini, il **webmaster** può suddividerle in un certo numero di **categorie** e sottocategorie:

animali | mare | montagna | arte | …

ma queste non necessariamente corrispondono al modo in cui le classificherebbe l'**utente**… i **tag** rappresentano una **classificazione alternativa**, più flessibile ed intuitiva:

bianco_e_nero compleanno felicità
mare Mario_e_Maria pittura
tramonto vacanze Venezia

Tagging system - I

Immaginiamo di avere un **catalogo**, per es. di immagini (tabella ***pictures*** nel database *tagging_sys*):

Campo	Tipo	Collation	Attributi	Null	Predefinito	Extra
id_img	int(11)			No	*Nessuno*	auto_increment
title	varchar(50)	latin1_swedish_ci		No	*Nessuno*	
descr	varchar(200)	latin1_swedish_ci		Sì	*NULL*	
file	varchar(20)	latin1_swedish_ci		No	*Nessuno*	

id_img	title	descr	file
1	Cala Chia	*NULL*	calaChia-20.jpg
2	Colonna romana	*NULL*	colonna-20.jpg
3	Fuochi d'artificio	*NULL*	fuochi.jpg
4	Rifugio Sella, Val d'Aosta	*NULL*	sella.jpg
5	Vigneti valdostani	*NULL*	vigneti.jpg

Tagging system - II

Per salvare i tag inseriti dagli utenti abbiamo bisogno di **due tabelle**:

- ***tags***, che conterrà l'elenco dei tag, con la loro frequenza (*count*):

Server: localhost ▸ Database: tagging_sys ▸ Tabella: tags

Campo	Tipo	Collation	Attributi	Null	Predefinito	Extra
id_tag	int(11)			No	*Nessuno*	auto_increment
tag	varchar(100)	latin1_swedish_ci		No	*Nessuno*	
count	int(11)			No	0	

- ***item_tag***, per salvare l'**associazione tra tag e immagine**:

Campo	Tipo	Collation	Attributi	Null	Predefinito	Extra
id_img	int(11)			No	*Nessuno*	
id_tag	int(11)			No	*Nessuno*	

Tagging system - III

Poi abbiamo bisogno di un'**interfaccia utente per l'inserimento dei tag**; questa può essere un ***form*** con dei campi di testo; per esempio:

Your tag:

Your tag:

send tags

Tagging system - IV

Infine vogliamo utililzzare i tag per navigare, cioè vogliamo mostrare all'utente una **tag cloud**; per esempio:

bianco_e_nero compleanno felicità
mare Mario_e_Maria pittura
tramonto vacanze Venezia

Quando l'utente **clicca su un tag** (es: *vacanze*):

```
<a href="tagging.php?s=14">vacanze</a>
```

lo script (contenuto in *tagging.php*) leggerà il parametro (*s=14*) e selezionerà tutte e sole le immagini taggate con quel tag

Tagging system - V

VEDI home.html
VEDI tagging.php

home.html: ridirezionamento (fatto con **Javascript**)

```
<BODY onLoad =
      "window.location.href='tagging.php?s=all'">
```

così il parametro *s* è inizializzato anche alla prima chiamata!
(cioè quando l'utente NON ha cliccato su alcun tag)

tagging.php:

1. leggiamo il contenuto di ***$_POST*** (che contiene eventuali tag inseriti dall'utente nel form) e **inseriamo i nuovi tag nel database** (effettuando gli opportuni controlli…)
2. leggiamo il parametro *s* in ***$_GET***: se è *all*, **mostriamo** (sulla sinistra) tutte **le immagini** (con il form per taggarle), altrimenti mostriamo solo le immagini taggate con il tag corrispondente al valore di *s* (es: *vacanze*)
3. mostriamo la **tag cloud** (sulla destra)

Vediamo ora i **dettagli**…

Tagging system - VI

VEDI tagging.php

1. leggiamo il contenuto di ***$_POST*** (assumiamo che il form per il tagging abbia *METHOD=POST*) e **inseriamo i tag nel database**

```
foreach (array_keys($_POST) as $item) {
[*] $clean_tag = htmlspecialchars(stripslashes($_POST[$item]));
 if ($clean_tag != "") {
  $sql_tag = "SELECT * FROM tags WHERE tag='".$clean_tag."'";
  $ris_tag = mysql_query($sql_tag) or die ...
  // se il tag esiste gia'...
  if (mysql_num_rows($ris_tag) != 0) {
   //assumiamo che $ris_tag contenga un solo record
   $riga_tag = mysql_fetch_array($ris_tag);
   // leggo l'id del tag (mi serve dopo)
   $id_tag = $riga_tag["id_tag"];
```

ciclo su *$_POST* (array associativo): le **chiavi** sono gli *id* delle immagini; i **valori** sono i tag inseriti [vedi form nelle slide successive]: per ogni immagine (*$item*), se è stato inserito un tag (se *$_POST[$item]* ≠ "") cerco nella tabella *tags* se il tag inserito (*$_POST[$item]*) esiste già; se esiste…

VEDI tagging.php

Tagging system - VII

```
 // aggiorno il count
 $new_count = $riga_tag["count"]+1;
 $sql_up = "UPDATE tags SET count=".$new_count."
            WHERE tag='".$_POST[$item]."'";
 $ris_up = mysql_query($sql_up) or die ...
}
```

aumento il suo *count* di 1 (*update*)

```
else {
 $new_count = 1;
 $sql_ins1 = "INSERT INTO tags (tag, count) VALUES
             ('".$_POST[$item]."', ".$new_count.")";
 $ris_ins1 = mysql_query($sql_ins1) or die ...
 // leggo l'id del tag (mi serve dopo)
 $sql_id = "SELECT * FROM tags
            WHERE tag ='".$ POST[$item]."'";
 $ris_id = mysql_query($sql_id) or die ...
 $tag_rec = mysql_fetch_array($ris_id);
 $id_tag = $tag_rec["id_tag"];
}
```

se invece il tag appena inserito non esiste ancora, lo inserisco nel DB (*insert*) con *count* a 1

VEDI tagging.php

Tagging system - VIII

```
 $sql_sel = "SELECT * FROM item_tag
   WHERE id_tag=".$id_tag." AND id_img = ".$item;
 $ris_sel = mysql_query($sql_sel) or die ...
 if (mysql_num_rows($ris_sel) == 0) {
   $sql_ins2 = "INSERT INTO item_tag
                VALUES ('".$item."', ".$id_tag.")";
   $ris_ins2 = mysql_query($sql_ins2) or die ...
 }
}
}
```

infine inserisco l'associazione immagine-tag (se non esiste già)

```
[*]
$clean_tag = htmlspecialchars(stripslashes($_POST[$item]));
```

ripulisco l'input da eventuali caratteri "problematici":
- *stripslashes* → elimina le virgolette
- *htmlspecialchars* → converte i caratteri speciali (es. <) in "entità" (<)

Tagging system - IX

VEDI tagging.php

2. leggiamo il parametro *s* in ***$_GET***: se è *all*, **mostriamo** tutte **le immagini** in catalogo (con il form per taggarle), altrimenti mostriamo solo quelle taggate con il valore di *s* (es: 14/*vacanze*)

```
// leggo il tag da cercare
$tag = $_GET["s"];
$tag_name = "";
if ($tag == "all") {
 $mess = "Insert your tags";
 $sql = "SELECT * FROM pictures";
 }
 else {
 // cerco nel DB il tag selez. dall'utente (dato l'id)
   $sql0 = "SELECT * FROM tags WHERE id_tag='".$tag."'";
   $ris0 = mysql_query($sql0) or die ...;
 // assumiamo che $ris0 contenga un solo record
   $row = mysql_fetch_array($ris0);
    ...
```

imposto il messaggio per l'utente e la query al DB per il caso iniziale (tutte le immagini)

altrimenti (caso in cui l'utente ha selezionato un tag)...

Tagging system - IX

VEDI tagging.php

```
...
$tag_name = $row["tag"];
$mess = "Images for ".$tag_name;
// cerco (nel join di item_tag e pictures) le immagini
// taggate con quel tag
$sql = "SELECT * FROM item_tag INNER JOIN pictures ON
        item_tag.id_img = pictures.id_img WHERE
        item_tag.id_tag=".$tag;
}
```

leggo il nome del tag dal DB

imposto il messaggio per l'utente e la query al DB per il caso tag selezionato (cerco, nel join di *item_tag* e *pictures*, le immagini taggate con quel tag)

Tagging system - X

VEDI tagging.php

```
$ris = mysql_query($sql) or die ...
while ($riga = mysql_fetch_array($ris)) {
 echo "<IMG SRC='img/".$riga["file"]."'>Your tag:
       <INPUT TYPE='text' NAME='".$riga["id_img"]."'/>";
}
```

per ogni immagine: la inserisco nella pagina e creo un campo di testo (*NAME* = *id* dell'immagine) per inserire un tag

Tagging system - XI

VEDI tagging.php

3. mostriamo la **tag cloud** [codice – un po' modificato – tratto da: www.v-nessa.net/2007/02/12/how-to-make-a-sexy-tag-cloud]:

```
function tag_info() {
 $result = mysql_query("SELECT * FROM tags") or die ...
 if (mysql_num_rows($result) != 0) {
  while ($row = mysql_fetch_array($result)) {
   $arr[$row['tag']] = $row['count'];
  }
  ksort($arr);
 }
 return $arr;
}
```

funzione che recupera dal DB le info sui tag e le mette in un array

- estraggo tutti i tag
- se c'è almeno un tag nel DB...
- costruisco un array associativo (*$arr*) in cui le chiavi sono i tag e i valori i corrispondenti *count*
- la funzione *ksort* prende come argomento un array associativo e lo ordina (ordine alfabetico sulle chiavi)

Tagging system - XII

VEDI tagging.php

```
function tag_cloud() {
 $min_size = 10;
 $max_size = 30;
 $tags = tag_info();
 if ($tags != null) {
  $minimum_count = min(array_values($tags));
  $maximum_count = max(array_values($tags));
  $spread = $maximum_count - $minimum_count;
  if ($spread == 0) {
   $spread = 1;
  }
  $cloud_html = "";
  $cloud_tags = array();
```

il *count* più piccolo

il *count* più grande

il delta tra il tag più popolare e il meno usato

variabile che conterrà il codice HTML della tag cloud

inizializzo l'array *$cloud_tags*

nella variabile *$tags* metto il risultato della funzione *tag_info()*, cioè un array associativo in cui le chiavi sono i tag e i valori i corrispondenti *count*

funzione che restituisce il codice HTML (stringa) della tag cloud

Tagging system - XIII

VEDI tagging.php

```
foreach ($tags as $tag => $count) {
 $size = $min_size + ($count - $minimum_count) *
         ($max_size - $min_size) / $spread;
 $cloud_tags[] = '<a style="font-size: '.floor($size).'px'.
        '"href="tagging.php?s='.$tag.'">'.$tag_name.'</a>';
}
```

per ogni coppia tag-count in *$tags*:
- calcolo la grandezza del carattere (in base alla popolarità del tag)
- aggiungo un elemento (stringa) all'array *$cloud_tags*

NB: `foreach` ha due forme:

1) **`foreach (array as $value)`** → scorro l'array e leggo solo i valori (utile soprattutto per array NON associativi)

2) **`foreach (array as $key => $value)`** → leggo sia le chiavi (che vengono assegnate a **`$key`**) sia i valori (che vengono assegnati a **`$val`**); utile soprattutto per array associativi

- la funzione *floor* prende come argomento un numero decimale (*float*) e lo arrotonda, restituendo l'intero inferiore (in formato *float* però)

Tagging system - XIV

VEDI tagging.php

```
    $cloud_html = join("\n", $cloud_tags)."\n";
  }
  else {
    $cloud_html = "<P>There are no tags!</P>";
  }
  return $cloud_html;
}
```

la funzione *join* è un alias della funzione *implode*: prendono come argomento un array e lo trasformano in stringa, concatenando gli elementi

Infine invoco la funzione *tag_cloud()* e stampo il risultato sulla pagina (*print*):

```
print tag_cloud();
```


Proposta di esercizio...

Modificate ***tagging.php*** e il **database** in modo che si gestiscano gli **utenti**... per esempio:

- costruite una o più tabelle nel DB per archiviare i dati degli utenti (potete anche popolarle a mano... sennò implementate un sistema di registrazione)
- implementare un semplice sistema di autenticazione (login e password)
- per ogni tag registrate l'autore (nel database: probabilmente dovrete aggiungere una tabella che memorizza le relazioni tra tags e utenti...)
- quando si clicca su un tag, fare comparire (per es. sotto la tag cloud) una nuova area che dice qualcosa del tipo:

  Il tag vacanze è stato inserito dai seguenti utenti:
  <lista utenti che hanno inserito quel tag>