



Università degli Studi di Firenze



Corso di Laboratorio di Telematica - AA. 2002-2003

Franco Pirri, Maria Chiara Pettenati, Claudio Bizzarri, Maurizio Masseti

Le applicazioni web: HTML, PHP4 e MySQL

PHP: introduzione

- PHP è un linguaggio di programmazione
- PHP convive normalmente dentro l'HTML
- PHP viene eseguito *dal server* durante l'erogazione della pagina
- L'uscita di PHP è normalmente codice HTML per il browser, ma ci sono molte eccezioni

PHP e HTML

- PHP è in genere *incluso* in HTML
- Tutto ciò che compatibile con HTML sul client è compatibile con PHP
- PHP aggiunge funzionalità lato server mantenendo tutte le possibilità di sviluppo lato client

PHP e HTML: escape

- Per inserire il codice PHP all'interno di HTML si utilizzano speciali tag:
- `<?php ?>` (*universale, il più usato*)
- `<% %>` (*da abilitare in php.ini*)
- `<SCRIPT LANGUAGE="PHP">`
`</SCRIPT>` (*attenzione con javascript*)

PHP e HTML: primo esempio

```
<HTML>
```

```
<HEAD>Esempio 01</HEAD>
```

```
<BODY>
```

```
<?php
```

```
    print "Esempio 01 in php";
```

```
?>
```

```
</BODY>
```

```
</HTML>
```

PHP e HTML: regole di escape

Il server ha solo due possibilità:

- sta lavorando in modalità HTML
- sta lavorando in modalità PHP

Si può uscire ed entrare dal PHP un numero qualunque di volte all'interno di una sola pagina PHP, utilizzando i codici di escape visti prima.

Sintassi, variabili e output

- PHP è un linguaggio “New Age”
 - È indulgente
 - Non necessita di dichiarare le variabili
 - Non necessita di dichiarare le funzioni
 - Opera automaticamente funzioni di “cast”
- PHP è un mezzo pratico per realizzare velocemente pagine HTML dinamiche

Sintassi

- HTML non è PHP, la sintassi è diversa!
- PHP è sintatticamente simile al C
 - Indifferente agli spazi
 - Case-sensitive (funzioni e nomi delle variabili)
 - I comandi terminano per punto e virgola
 - Commenti C, C++ (Java) e Perl (script)
 - Espressioni standard

Sintassi: esempio

```
<?PHP
```

```
$a = 3 + 2 * 5;
```

```
$b = (3 + 2) * 5;
```

```
$c = $a + $b;
```

```
print ("Il valore è $c");
```

```
?>
```

Variabili: definizioni

- Tutte le variabili iniziano con dollaro (\$)
- Il valore è l'assegnazione (=) più recente
- Le variabili non devono essere dichiarate
- Le variabili non hanno un tipo oltre a quello del loro valore corrente
- Le variabili utilizzate prima di essere assegnate hanno valori predefiniti

Valori pre-assegnazione

Le variabili PHP non ancora assegnate:

- Se utilizzate in ambito numerico assumono il valore 0
- o se utilizzate come stringhe sono la stringa vuota
- o se utilizzate come valore booleano assumono il valore falso (FALSE)

Variabili: istruzione IsSet

```
$a = 0;  
print("a vale $a, b vale $b");  
if ($a == $b) print ("a e b sono  
    uguali");  
if (IsSet($a))  
    print ("a è stata impostata");  
if (IsSet($b))  
    print ("b è stata impostata");
```

Variabili: ambito

- Le variabili hanno valore nella stessa pagina HTML a prescindere dal numero di ESCAPE
- Nelle funzioni (trattate più avanti) le variabili hanno tutte ambito locale a meno di specifiche dichiarazioni

Variabili: ambito (esempio)

```
<HTML>
<HEAD>Ambito delle variabili</HEAD>
<BODY>
<?php
    $nome = "Claudio Bizzarri";
?>
<H1>Pagina iniziale</H1>
<?php
    print ("Autore: $nome");
?>
</BODY>
</HTML>
```

Output

- Utilizzeremo PRINT, ma esistono altre istruzioni (ECHO)
- PRINT inserisce una stringa nel flusso di informazioni di HTTP dal server verso il client
- Esiste anche PRINTF, ma si usa raramente a causa della potenza di PRINT

Output: interpolazione

- Le variabili all'interno dei doppi apici vengono interpolate
- Le stringhe fra apici singoli vengono inviate inalterate

```
print ("Il valore di a è $a");  
print ('Con $a si indica una variabile');
```


Output: andare a capo

- Come in C, il carattere “\n” significa newline (a capo)
- Nel codice HTML l’interruzione di riga viene di solito ignorata
- Occorre utilizzare il tag

- Si può ricorrere al testo preformattato con <PRE> </PRE>

Output: andare a capo (PHP)

```
<?php
    $nome = "Claudio Bizzarri";
    $titolo = "Corso di PHP";
    /* metodo BR */
    print ("Autore: $nome<BR>\n");
    print ("Titolo: $titolo<BR>\n\n");
    /* metodo PREFORMATTED */
    print ("<PRE>\nAutore: $nome\n");
    print ("Titolo: $titolo\n</PRE>\n");
?>
```

Output: andare a capo (HTML)

...

Autore: Claudio Bizzarri

Titolo: Corso di PHP

<PRE>

Autore: Claudio Bizzarri
Titolo: Corso di PHP
</PRE>

...

Tipi in PHP

- Prima regola: non preoccuparsi
 - Niente dichiarazione dei tipi delle variabili
 - Conversione di tipo automatica
 - Tipi assegnati per contesto
- I tipi in PHP sono solo 6, quattro semplici e due composti

Tipi in PHP: esempi (1)

- Niente dichiarazione dei tipi delle variabili

```
$a = 2.34;
```

```
$b = "Ciao a tutti";
```

```
$c = 3;
```

Tipi in PHP: esempi (2)

- Conversione di tipo automatica

```
$pi = 3 + 0.1415;
```

Tipi in PHP: esempi (3)

- Tipi assegnati per contesto

```
$sub = substr(12345,2,2);  
print ("sub vale $sub\n");
```

sub vale 34

Tipi in PHP: elenco dei tipi

Tipi semplici

- Interi
- A virgola mobile
- Booleani
- Stringhe

Tipi composti

- Array
- Oggetti

Tipi in PHP: interi

- Gli interi in PHP sono a 32 bit
- Hanno sempre il segno
- Il massimo è $2^{31} - 1$ (2.147.483.647)
- Il minimo è $-2^{31} + 1$ (- 2.147.483.647)

Tipi in PHP: a virgola mobile

- Hanno i valori decimali
- Possono essere scritti in vari modi:

`$a = 2.3;`

`$b = 23e-5;`

`$c = 2e4;`

`$d = 224.23221e-24`

Tipi in PHP: booleani

- Assumono due valori
 - TRUE (vero)
 - FALSE (falso)
- Altri tipi utilizzati come booleani:
 - Se è un numero è falso se esattamente uguale a zero
 - Se è una stringa è falso se è la stringa è “0” oppure è vuota
 - Se è un tipo composto è falso se non contiene altri elementi

Tipi in PHP: stringhe

- Le stringhe sono sequenze di caratteri
- Sono racchiuse da apici singoli se occorre inserire una stringa esattamente come la scriviamo
- Sono racchiuse fra apici doppi se occorre attivare l'interpolazione

Stringhe: interpolazione

- Certe sequenze di caratteri che cominciano con il backslash (\) vengono sostituite con caratteri speciali
- I nomi delle variabili che cominciano per \$ vengono sostituite con il valore delle variabili dopo che questo è stato convertito in tipo stringa

Stringhe: caratteri di escape

- `\n` = newline (nuova riga)
- `\r` = carriage return (ritorno a capo)
- `\t` = carattere di tabulazione (TAB)
- `\$` = carattere dollaro (\$)
- `\"` = carattere doppi apici (")
- `\\` = carattere backslash (\)

Stringhe: interpolazione (es.)

```
$a = "questo";  
$b = "quello";  
$c = .22e+1;  
print ("$a,$pippo,$b+$c");
```

questo,,quello+2.2

Array

- Gli array in PHP sono sostanzialmente diversi dagli altri linguaggi (tranne le variabili hash del PERL)
- Gli array in PHP sono sempre e soltanto di tipo associativo, ovvero memorizzano i valori degli elementi *associandoli* a delle stringhe dette *chiavi*, anziché utilizzare il classico metodo di indice numerico
- Possono essere pensati come vettori con indice alfanumerico

Array: esempi 1

```
$a[3] = "Ciao";  
$a[5] = "mondo";  
$a[100000] = "Quanti siamo?";  
print "$a[3], $a[5]. $a";
```

Ciao, mondo.

Array: esempi 2

```
$cibo['es'] = "Paella";  
$cibo['it'] = "Pasta";  
$cibo['jp'] = "Sashimi";  
print("Preferisco la $cibo['it']");
```

Preferisco la pasta.

Array – la funzione array()

- La funzione array() serve per generare un array
- Se non si specificano gli indici, PHP assume che si voglia utilizzare gli array come normali vettori con indice numerico

Array – esempi

```
$spesa = array("mele", "pere", "pane", "latte");  
/* oppure */  
$spesa[0] = "mele";  
$spesa[1] = "pere";  
$spesa[2] = "pane";  
$spesa[3] = "latte";  
/* ma anche */  
$spesa[] = "mele";  
$spesa[] = "pere";  
$spesa[] = "pane";  
$spesa[] = "latte";
```

Array – esempi (continua)

```
/* associazione esplicita fra chiave e valore */  
$spesa = array(  
    0 => "mele",  
    1 => "pere",  
    2 => "pane",  
    3 => "latte");
```

Array multidimensionali

- Ogni elemento di un array può essere di tipo qualunque, anche array
- Un array che contiene altri array può essere visto come un array multidimensionale

Array multidimensionale – es.

```
/* comando for più avanti nel corso */  
for($i=1;$i<$n;$i++)  
{  
    for($j=1;$j<$n;$j++)  
    {  
        for($k=1;$k<$n;$k++)  
        {  
            $s[$i][$j] += $a[$i][$k] * $a[$k][$j];  
        }  
    }  
}
```

Variabili predefinite

- Esistono molte variabili già definite che contengono alcuni dati del sistema
- Vi sono differenze fra le varie versioni di php, noi faremo riferimento alla 4.1 (e successive)
- Quelle di nostro interesse sono `$_GET` `$_POST` `$_REQUEST` ma ce ne sono molte altre (vedere manuale)

`$_GET`

Si tratta di un array associativo composto dalle variabili passate allo script tramite il metodo GET di GTTP. E' una variabile 'superglobale', ovvero viene dichiarata globale in modo automatico e quindi può essere utilizzata in un qualunque punto dello script senza necessità di altre operazioni.

`$_POST`

Si tratta di un array associativo composto dalle variabili passate allo script tramite il metodo POST di HTTP. E' una variabile 'superglobale', ovvero viene dichiarata globale in modo automatico e quindi può essere utilizzata in un qualunque punto dello script senza necessità di altre operazioni.

`$_REQUEST`

Si tratta di un array associativo composto dalle variabili degli array `$_GET`, `$_POST` e `$_COOKIE`. E' una variabile 'superglobale', ovvero viene dichiarata globale in modo automatico e quindi può essere utilizzata in un qualunque punto dello script senza necessità di altre operazioni.

Oggetti:

- PHP fornisce un limitato numero di funzioni a supporto della programmazione a oggetti (OOP):
 - Definizione di classi
 - Funzioni di membri
 - Ereditarietà
- Non verranno date nozioni di OOP durante questo corso

Strutture di controllo

Il controllo è probabilmente l'aspetto in cui PHP assomiglia maggiormente al linguaggio C

- Espressioni booleane
- Diramazioni
- Cicli
- Strutture di controllo alternative

Espressioni booleane

```
$a = FALSE;
```

```
$b = TRUE;
```

```
if ($a and $b)
```

```
    print("a e b sono vere");
```

```
if ($a or $b)
```

```
    print("a o b è vera");
```

Confronto - 1

```
$a = 3;  
$b = 4;  
if ($a == $b)  
    print("a è uguale a b");  
else  
    print("a è diverso da b");
```

Confronto - 2

```
$a = 3;  
$b = 4;  
if ($a > $b)  
    print("a è maggiore di b");  
else  
    print("a è minore o uguale a b");
```


Confronto fra stringhe

- Gli operatori di confronto funzionano con tutti i tipi, anche con le stringhe
- Può essere ambiguo quando una stringa somiglia ad un numero: meglio usare `strcmp()` (più avanti nel corso)

Confronto fra stringhe – es.

```
$a = " 04";  
$b = "0003";  
if ($a>$b) print("a è maggiore di b");  
else print("a non è maggiore di b");  
print ("<BR>")  
if (strcmp($a,$b)) print("a è più grande di b");  
else print("a non è più grande di b");
```

a è maggiore di b
a non è più grande di b

Diramazioni

Si può fare interamente riferimento all'equivalente in linguaggio C, faremo solo un rapidissimo accenno

- if-else
- if-elseif-else
- switch-case

if-else (esempio)

```
if ($a>$b)  
    print("a è maggiore di b");  
else  
    print("a non è maggiore di b");
```

if-elseif-else (esempio)

```
if ($a==1)
    print("a è uguale a uno");
elseif ($a==2)
    print("a è uguale a due");
else
    print("a non vale né uno né due");
```

switch-case (esempio)

```
switch($a)
{
    case 1:
        print("a è uguale a uno");
        break;
    case 2:
        print("a è uguale a due");
        break;
    default:
        print("a non vale né uno né due");
}
```

Cicli

- `while(condizione)`
 dichiarazione
- `do` dichiarazione
 `while(condizione)`
- `for(espressione iniziale,`
 `condizione, espressione fine`
 `ciclo)`
 dichiarazione

Algoritmo di Newton

(Esempio di ciclo non limitato)

Sia t il numero di cui si cerca la radice,
sia ε l'approssimazione ed s il valore cercato:

- $s \leftarrow 1$
- ripeti finché $|s^2 - t| > \varepsilon$
 - $s \leftarrow (s^2 + t) / 2s$

while (esempio)

```
/* Radice quadrata (metodo di Newton) */
$t=121; $s=1.0; $p=0.0000001;
$sq = $s * $s;
while(($sq-$t > $p) or ($sq-$t < -$p))
{
    print("valore corrente: $s<BR>");
    $s = ($s + ($t/$s))/2;
    $sq = $s * $s;
}
print("<h2>La radice di $t è circa $s</h2>");
print("con approssimazione di $p");
```

while (output dell'esempio)

Valore corrente: 1

Valore corrente: 61

Valore corrente: 31.491803278689

Valore corrente: 17.667036464956

Valore corrente: 12.257974853712

Valore corrente: 11.064549844141

Valore corrente: 11.000188289738

La radice di 121 è circa 11.0000000001611

con approssimazione di 1E-007

Funzioni

- Ogni vero linguaggio di programmazione ha un metodo per *l'astrazione procedurale*
- Il PHP utilizza le funzioni
- Il PHP ha migliaia di funzioni incorporate
- Si possono definire anche funzioni proprie

Funzioni – esempio definizione

```
function maggiorenne($a)
{
    if ($a >= 18)
    {
        $res = TRUE;
    }
    else
    {
        $res = FALSE;
    }
    return($res);
}
```

Funzioni – esempio chiamata

```
<?PHP
    /* la variabile $eta è già definita */
    if (maggiorenne($eta))
    {
        print("Puoi accedere al sito<BR>");
    }
    else
    {
        print("Non puoi accedere<BR>");
    }
?>
```

Funzioni – cosa accade

Quando viene chiamata una funzione PHP accadono le seguenti cose:

- PHP cerca la funzione per nome (se non la trova genera un errore)
- PHP sostituisce le variabili nell'elenco di parametri della definizione con i valori degli argomenti (passaggio per valore)
- Vengono eseguite le dichiarazioni nel corpo della funzione, restituendo l'eventuale valore se viene eseguita la dichiarazione di “return”

Funzioni - variabili

- PHP è tollerante con il numero di parametri passati (sia in eccesso che in difetto)
- Le variabili all'interno di una funzione sono sempre private, tranne definizioni apposite (keyword *global*)
- Le funzioni non tengono memoria della loro esecuzione a meno dell'uso di *static*

Funzioni - ambito

- Una funzione definita in uno script è disponibile ovunque in quello script
- Per chiarezza è meglio definire le funzioni all'inizio, ma non è necessario
- Solitamente si usano file.inc che contengono tutte le funzioni necessarie e si utilizza *include* oppure *require*
 - include: include il file sempre
 - require: sostituisce l'istruzione con il file (nei cicli avviene quindi una sola volta)

Stringhe e funzioni associate

- Si utilizza gli apici singoli o doppi
 - apice doppio: interpolazione
 - apice singolo: nessuna interpolazione
- Non esiste il tipo “carattere”, si utilizza una stringa con indice
- Operatori classici: concatenazione (.), confronto, ecc.

Stringhe - esempio

```
<?php
    $s = 'la variabile $a non esiste';
    print ("S vale: <B>$s</B><BR>\n");
    $len = 25; /* meglio usare strlen */
    for($i=$len;$i>=0;$i--)
    {
        $sinv = $sinv . $s[$i];
    }
    print ("sinv vale: <B>$sinv</B><BR>");
?>
```

Stringhe – esempio (out)

S vale: la variabile \$a non
esiste

sinv vale: etsise non a\$ elibairav
al

s vale: **la variabile \$a non esiste**

sinv vale: **etsise non a\$ elibairav al**

Stringhe - funzioni

- Esistono moltissime funzioni per la manipolazione delle stringhe: `strlen`, `strcpy`, `strrpos`, `strcmp`, ecc.
- Occorre il manuale a portata di mano
- Ci sono anche le espressioni regolari, solo per utilizzo avanzato (PERL)

Matematica

- 5 operatori base: + - * / %
- Due soli tipi: interi e floating point
- Operatori incremento e decremento: gli stessi del C
- Operatori di confronto: gli stessi del C, con l'aggiunta dell'operatore "identico a" (===)

Matematica: note sul confronto

- Gli operatori di confronto (tranne `===`) effettuano conversioni automatiche di tipo
- Le seguenti espressioni sono tutte vere:
 - `3 > 2.0`
 - `1 < "2"`
 - `0 == FALSE`

Matematica: operatore ===

- L'operatore === non effettua conversioni automatiche
- Utile per non confondere stringhe con valori numerici o booleani
- Le seguenti espressioni sono tutte false:
 - `2 === 2.0`
 - `2 === "2"`
 - `0 === FALSE`

Matematica: altre funzioni

- Sono disponibili moltissime funzioni: trigonometrica, calcoli logaritmici, conversioni di base e anche calcoli a precisione arbitraria.
- Prima di scrivere una funzione, è bene controllare che non sia già stata inclusa nelle librerie.

Passaggio di informazioni

- HTTP è **STATELESS**, ovvero ogni richiesta genera un processo che ha il compito di servire un file e poi viene eliminato
- Ogni richiesta è sostanzialmente separata dalle altre e non vengono passate informazioni su quante e quali pagine il client ha richiesto prima

Sessioni PHP

- PHP consente di salvare un certo numero di variabili e “propagarle” da una pagina all'altra
- Il meccanismo è molto sofisticato ed utilizza metodiche diverse per ottenere il risultato su qualunque tipo di browser (metodi GET, POST e Cookie)

Funzioni di sessione

- Le funzioni principali sono:
 - `session_start()`
 - `session_register()`
 - `session_unregister()`
 - `session_destroy()`
- Esistono altre funzioni che non vedremo in aula ma che sono comunque utili

session_start()

- Non ha argomenti e costringe PHP ad utilizzare l'ID di sessione che gli è stato passato o a creare un nuovo ID. Se viene trovato quello vecchio, PHP recupera tutte le variabili che sono state registrate per quella sessione

session_register()

- Prende come argomento il nome di una variabile (senza il segno \$ davanti) e la registra in quella sessione
- Può essere passato anche un array, per registrare un gruppo di variabili in una unica chiamata

session_unregister()

- Prende come argomento un nome di variabile (senza il segno \$ davanti) e la cancella dalla sessione
- Il valore della variabile non verrà più propagata nelle pagine successive

session_destroy()

- Distrugge la sessione in corso e tutte informazioni di sessione associate (variabili di sessione, id, ecc)
- Non cancella i valori delle variabili all'interno della pagina PHP

Considerazioni sulle sessioni

- I valori delle variabili sono memorizzate sul server e non sui client
- Le sessioni trasformano *di fatto* l'HTTP da **STATELESS** a **STATEFULL**, consentendo applicativi come “carrelli della spesa”, forum, quiz on line, ecc...
- L'applicativo che utilizzerete per il test finale è stato scritto in PHP da vostri colleghi del quinto anno

L'Universita' di Firenze impone il proprio copyright su tutti i documenti pubblicati sul sito:

<http://mmedia5.det.unifi.it>

E' pertanto vietata la riproduzione o la copia totale o parziale dei documenti per qualunque scopo e con qualunque mezzo o supporto, anche telematico.

In deroga a quanto sopra, e' permessa la consultazione a distanza dei documenti tramite una rete di comunicazione, per il solo uso personale. La memorizzazione, su qualunque supporto, e' ammessa solo per quanto necessario o implicito durante la consultazione remota. Ogni copia cosi' creata sara' distrutta immediatamente dopo la consultazione. Tuttavia l'Universita' di Firenze consente la circolazione dei documenti a scopo educativo o scientifico.

Questo puo' avvenire a titolo oneroso o gratuito a seconda dei casi. La disponibilita' di una licenza d'uso per un particolare documento e' sempre segnalata da un riferimento, contenuto nel documento stesso, ad un file specifico, che contiene la licenza d'uso. Il file e' di solito denominato "license.txt".

Ogni uso non espressamente autorizzato dai termini della licenza d'uso e' espressamente vietato.

Gli studenti dell'Universita' di Firenze sono autorizzati ad usare il contenuto del sito a titolo gratuito a fini didattici e personali durante tutto il percorso di studio.

The Universita' di Firenze mantains all rights on all documents published on the WEB site:

<http://mmedia5.det.unifi.it>

Therefore, partial or total copy or reproduction of any such document is forbidden. Remote consultation by electronic communication is allowed for personal use only. Memorization on any support is allowed only during remote consultation, and only if required or implied in the remote consultation. Any copy created during the remote consultation will be destroyed immediately after remote consultation ends.

Nevertheless, Universita' di Firenze allows free circulation of a document for educational or scientific purposes. License availability is always signalled by a reference in the document, usually to a file named "license.txt". Licence may be free or with cost, as specified in the license.

Any use, not specifically authorized by the license clauses is forbidden.

Students of the Universita' di Firenze are permitted to free use of the site content in connection with their studies at the university.

Questo documento puo' essere liberamente copiato e distribuito da chiunque, ma a nessuno e' permesso di cambiarlo in alcun modo.

Preambolo

Il copyright imposto sui documenti pubblicati sul sito MMEDIA5.DET.UNIFI.IT ha lo scopo di consentire la libera circolazione del lavoro a scopo educativo, mantenendo pero' il doveroso riconoscimento agli autori delle varie parti. Si vuole, inoltre, consentire l'ulteriore distribuzione del lavoro sotto qualunque forma, anche con modifiche, mettendo pero' il successivo ricevente in grado di conoscere da chi il materiale sia stato originariamente scritto e da chi rivisto o modificato. Per questo, si impone il Copyright su tutto il materiale, ma si concede gratuitamente licenza per l'uso e l'ulteriore distribuzione, con la possibilita' di modificare il materiale, purché vengano seguite le regole scritte piu' avanti.

Il diritto di cui sopra e' concesso con la restrizione che il materiale modificato e redistribuito sia soggetto alle stesse restrizioni del materiale originario, e che la distribuzione avvenga a titolo gratuito o con la sola copertura delle spese vive con un piccolo margine per le spese generali di distribuzione. Inoltre, la re-distribuzione del materiale o la distribuzione di materiale modificato dovranno essere fatte in modo da garantire che ulteriori distribuzioni vengano fatte mantenendo le condizioni originarie.

Ogni utente di MMEDIA5 potra' proporre all'Universita' di Firenze la pubblicazione di documenti in MMEDIA5. Scrivere a: fpirri@ing.unifi.it
Tali documenti saranno soggetti ai termini qui specificati. Le condizioni per la licenza di pubblicazione e modifica sono riportate di seguito.

LICENZA PER IL MATERIALE "MMEDIA5"

CONTENUTO NEL SITO WEB "MMEDIA5.DET.UNIFI.IT" CONDIZIONI PER L'USO, LA MODIFICA E LA DISTRIBUZIONE

1.- Questa licenza e' applicabile al materiale contenuto nel sito WEB del Laboratorio di Tecnologia della Telematica, Dipartimento di Elettronica e Telecomunicazioni dell'Universita' di Firenze, via di S. Marta 3, 50139 Firenze - Italia (attualmente con indirizzo internet: <http://mmedia5.det.unifi.it/>) che riporta al suo interno un avviso o legame con un file contenente la presente licenza. In questa licenza, con MMEDIA5 si intende un qualunque documento originariamente presente nel sito. Per "documento derivato" si intende ogni documento che contenga porzioni oppure un intero documento MMEDIA5, con o senza modifiche, con o senza traduzioni in altra lingua, con o senza variazioni di supporto di memorizzazione o stampa; in questa licenza ogni documento di questo tipo e' indicato come "documento derivato". Per licenziatario si intende qualunque persona o organizzazione che copia, consulta, legge, memorizza su un qualunque supporto, produce o distribuisce a terzi un MMEDIA5 o un documento derivato. Per UNIVERSITA' si intende l'Universita' di Firenze.

2.-Un licenziatario puo' copiare, consultare, leggere, memorizzare su un qualunque supporto, produrre e distribuire a terzi un MMEDIA5, purché su ogni copia, produzione, esibizione o distribuzione sia evidenziato il copyright originario, il ricevente sia adeguatamente informato della provenienza del materiale e dell'esistenza di questa licenza e questa licenza sia inserita indivisibilmente e senza modifica alcuna assieme all'MMEDIA5.

L'eventuale cessione o distribuzione devono essere a titolo gratuito. E' comunque ammessa l'imposizione di un rimborso delle spese legate al supporto fisico di memorizzazione dello MMEDIA5, con un piccolo margine per il recupero delle spese generali legate alla riproduzione fisica.

- 3.- Il licenziatario puo' produrre documenti derivati, ai sensi dell'articolo 1, e distribuirli a terzi purché siano rispettate tutte le seguenti condizioni:
- 3.1- ogni documento derivato deve riportare chiaramente la data e l'autore delle modifiche effettuate;
 - 3.2- il licenziatario deve assicurare che ogni documento derivato sia sottoposto alla stessa licenza del documento originario, così che la terza parte ricevente sia impegnata a sua volta negli stessi termini di questa licenza;
 - 3.3- ogni documento derivato deve riportare, all'inizio e in buona evidenza questa licenza o un legame ad un file che la contiene, e il file deve essere distribuito indivisibilmente dal documento derivato;
- 4.- Al licenziatario e' espressamente vietato copiare, consultare, leggere, memorizzare su un qualunque supporto, produrre e distribuire a terzi un MMEDIA5, se non nelle forme e nei modi previsti in questa licenza. Ogni forma di inosservanza di questa norma comporta l'immediata revoca di ogni diritto concesso con questa licenza.
- 5.- L'uso di materiale soggetto a Copyright senza un esplicito assenso del proprietario del Copyright e' proibito dalla legge. L'UNIVERSITA' pone come prerequisito per ogni uso di MMEDIA5 l'accettazione di questa licenza. Quindi, e' fatto espresso divieto, a chiunque non intenda accettare i termini di questa licenza, di usare MMEDIA5 in alcun modo. Peraltro, non e' richiesta alcuna forma di accettazione esplicita della licenza, l'uso di MMEDIA5 costituisce implicita accettazione e conferisce al licenziatario tutti i diritti qui espressi.
- 6.- Ogni volta che MMEDIA5 od un documento derivato vengono distribuiti, al ricevente e' automaticamente estesa questa licenza. Al licenziatario non e' permesso imporre altri obblighi sul ricevente oltre la presente licenza. In ogni caso il licenziatario non e' considerato responsabile dell'uso che il ricevente fa di MMEDIA5.
- 7.- Se per qualunque motivo, compresi regolamenti o leggi dello Stato, non e' possibile per il licenziatario imporre questa licenza, o parti di essa, al ricevente, allora il licenziatario non puo' distribuire MMEDIA5, ne' parti di esso, ne' documenti derivati, in alcun modo.
- 8.- L'UNIVERSITA' potra' pubblicare revisioni di questa licenza. Le nuove versioni avranno intendimenti simili, ma potranno differire nei dettagli per far fronte a nuove situazioni. Ogni nuova versione avra' un proprio numero distintivo e sara' applicata ai documenti MMEDIA5 da allora pubblicati.
- 9.- Usi di MMEDIA5 diversi da quanto sopra specificato potranno essere autorizzati dall'UNIVERSITA'. Queste autorizzazioni non comporteranno variazioni per i licenziatari preesistenti.
- 10.- Essendo la licenza gratuita, l'UNIVERSITA' fornisce MMEDIA5 così come si trova, e non assume alcuna garanzia di esattezza dei contenuti, o di adeguatezza a qualsiasi scopo. Inoltre, le singole parti di MMEDIA5 sono espressione dei vari autori o revisori menzionati e non espressione dell'UNIVERSITA', non comportano ne' implicano accettazione del contenuto da parte dell'UNIVERSITA'.