

1. WebIbrix wi_

WebIbrix (di seguito **wi_**) è un "framework", un ambiente che genera dinamicamente [su di un server web] pagine html. Basato su script **php**, è rilasciato per l'utilizzo sia su server **Apache** che su server **IIS**.

Con wi_ si realizzano "moduli" che, scritti in html, vengono inseriti in una "cornice" di dimensioni discrete definite dall'utente e fisse, come posizionamento, nella pagina web da generare.

Tali "moduli" sono riutilizzabili -come tanti "mattoncini"- in altre pagine dello stesso progetto.

In pratica lo sviluppatore segmenta il codice html in parti (i "moduli" wi_) che rappresentano zone ben definite dell'area video visualizzabile e le utilizza come componendo un "puzzle".

Per utilizzare wi_ è richiesta una elementare conoscenza dei tag html (in appendice un breve riepilogo dei più comuni) ma wi_ supporta anche fogli di stile e script javascript (più in generale qualsiasi script *client-side*) permettendo, così, la realizzazione di pagine web ben curate ed adattabili a più browser.

Il framework wi_ è attivato dallo script php **main.php**.

Dalla versione 2.0 è possibile inserire script PHP in pagine wi_ e variabili PHP in moduli wi_.

2. *Gli elementi base di wi_*

wi_ utilizza tre entità:

- il modulo;
- la pagina;
- lo script PHP.

I moduli sono l'elemento primo del disegno della pagina web; le "pagine" i contenitori dei moduli e la base per la navigazione nel progetto stesso. In pratica una pagina non richiama ("linka") con un click un "modulo wi_" ma richiama una "pagina wi_" che contiene il modulo interessato (*assieme ad altri eventualmente definiti per quella pagina*). Le pagine possono anche richiamare script PHP rendendo possibili pagine dinamiche.

2.1 *Convenzioni di utilizzo.*

Moduli e Pagine

-I moduli sono documenti di testo con un prefisso costante "wi_mod" ed un'estensione ".txt"

-Le pagine sono documenti di testo con prefisso preferibilmente "wi_pag" ed estensione ".txt".

-Non sono ammessi caratteri di controllo e/o font all'interno del modulo o della pagina e questo dovrà essere prodotto con applicazioni del tipo "notepad".

-In una pagina sono ammessi "n" moduli ma deve essercene soltanto uno per riga.

-In una pagina sono ammessi "n" script PHP, ma deve essercene soltanto uno per riga.

-I moduli vanno indicati nelle pagine senza l'estensione .txt, mentre gli script devono avere l'estensione .php.

Esempi di nomi moduli e pagine :

wi_mod01.txt
wi_mod84a.txt

wi_pag04.txt

2.2 I moduli wi_

I moduli wi_ contengono i tag wi_, le variabili di sistema ed il codice html.

I tag wi_ normalmente presenti sono l' `*x=` e l' `*y=` che posizionano il modulo nella pagina web generata. Va ricordato che il framework wi_ centra orizzontalmente ed automaticamente (*per risoluzioni maggiori di 1024 pixel orizzontali*) la pagina visualizzata, tenendo così conto della risoluzione del video; ciò significa che se un modulo ha il tag `*x=1` e la risoluzione video è, per esempio, di 1280 pixel sarà posizionato orizzontalmente al pixel 129, ovvero

$$*x+(1280-1024/2)=129$$

Il tag wi_ `*mod=` precede il codice html e viene normalmente posizionato (per agevolare la lettura dell'html) alcune righe prima del codice stesso.

Esempio di modulo :

```
*x=10
*y=10
*mod=
    <font face= "verdana" size=1>
Ciao!
    </font>
```

Se non si indicano i tag `*w=` e `*h=` le dimensioni saranno determinate dinamicamente secondo il contenuto ma, se si desidera visualizzare un colore di sfondo in un'area maggiore del contenuto, i tag di `*w=` e `*h=` dovranno essere assegnati.

Attenzione : se non si indicano valori per `*x=` o `*y=`, saranno assunti i valori di default ovvero 1 per la coordinata orizzontale ed 1 per quella verticale.

2.2.1 I tag wi_

I tag wi_ delimitano fisicamente la cornice video che andrà a contenere il modulo wi_.

Norme sintattiche:

- i tag wi_ possono essere scritti in minuscolo, maiuscolo o misto minuscolo/maiuscolo;
- è ammesso un solo tag wi_ per riga;
- un tag wi_ deve sempre iniziare con il carattere asterisco (*), terminare con il segno uguale (=) e non deve contenere spazi bianchi tra gli stessi;
- un tag wi_ può avere un valore nullo/blank dopo il segno uguale (=) : il tag non sarà considerato e, dove presente, sarà inserito il valore di default.

2.2.1.1 Riepilogo tag wi_

Tag wi_	Funzione
*x=	Posizione orizzontale in pixel (se blank/vuoto = 1)
*y=	Posizione verticale in pixel (se blank/vuoto = 1)
*h=	Altezza in pixel del modulo (se blank/vuoto = altezza dinamica)
*w=	Larghezza in pixel del modulo (se blank/vuoto = larghezza dinamica)
*col=	Colore testo (vedi colori html)
*bcol=	Colore sfondo (vedi colori html)
*bt=	Tipo bordo 0 no 1 linea continua 2 punti 3 linea tratteggiata 4 doppia linea 5 3D
*bc=	Colore bordo
*bs=	Bordo spessore, valori compresi da 1 a 10 (se blank/vuoto = 1)
*mod=	Codice html del modulo
*aln=	Allineamento testo nel modulo 0/blank default pagina 1 sinistra 2 destra 3 centrato 4 giustificato
*ovf=	Overflow 0/blank (default) se il modulo supera le dimensioni definite da *h e *w , viene visualizzato secondo i default del browser 1 vengono aggiunte automaticamente le barre di scroll (verticali ed orizzontali) SOLO quando il modulo (altezza, larghezza) supera una delle dimensioni 2 vengono sempre aggiunte le barre di scroll anche se il modulo non supera le dimensioni definite da *h e *w
*fnt=	Font (verdana, arial, etc..)
*fsz=	Dimensioni font in pixel
*lht=	Altezza riga testo in pixel (default 14 pixel)

N.B. : Il tag **mod=* deve essere l'ultimo tag wi_ del modulo.

2.2.2 Variabili di sistema wi_

Ambito di utilizzo : **moduli , pagine** (*limitatamente alle espressioni condizionali*)

Nel codice html possono essere inserite le variabili di sistema wi_: nella pagina generata la variabile inserita sarà sostituita dal relativo valore corrispondente.

Norme sintattiche:

- le variabili di sistema wi_ devono essere scritte SOLO in minuscolo;
- non vi è nessun limite nel numero delle variabili di sistema wi_ utilizzate;
- una variabile di sistema wi_ deve sempre iniziare con ? e terminare con ?;
- non devono esserci spazi bianchi tra ? e ?.

Esempio :

```
*x=10
*y=10
*mod=

<font face= "verdana" size=1>

Ciao, oggi è ?dw? ?dd? ?my? ?yyyy?

</font>
```

Sarà visualizzata questa pagina :

Ciao, oggi è Venerdì 15 Ottobre 2010

N.B. : Nell'installazione di base è presente il modulo wi_modsys che riporta tutte le variabili di sistema.

2.2.2.1 Riepilogo variabili wi_

<i>Variabile</i>	<i>Valore restituito</i>
?dd?	Giorno
?mm?	Mese
?yy?	Anno a 2 cifre
?yyyy?	Anno a 4 cifre
?hh?	Ora
?hm?	Minuti
?hs?	Secondi
?dw?	Giorno della settimana
?dwe?	Giorno della settimana in inglese
?my?	Mese descrittivo
?mye?	Mese descrittivo in inglese
?nw?	Numero della settimana
?ip?	Ip utente
?dom?	Dominio del client
?ver?	Versione wi_
?ipgeo?	Nazione del client

2.2.2.2 **Costanti figurative**

Ambito di utilizzo : **moduli**

Le costanti figurative permettono di velocizzare l'inserimento di caratteri grafici o speciali senza dover ricorrere alla loro codifica esadecimale utilizzabile con l'html.

E' da notare la presenza della costante blank (*spazio bianco*) rappresentata in html dalla stringa ** ** : è bene ricordare che si ricorre a tale costante per poter rappresentare correttamente un numero di blank superiore ad uno, cosa altrimenti non possibile se gli spazi sono immessi da tastiera poiché il browser compatta stringhe di tali spazi.

Norme sintattiche:

- le costanti figurative devono essere scritte SOLO in minuscolo;
- non vi è nessun limite nel numero delle variabili di sistema wi_ utilizzate;
- una variabile di sistema wi_ deve sempre iniziare con ? e terminare con ?;
- non devono esserci spazi bianchi tra ? e ?.

Attenzione : Alcuni di questi caratteri possono non essere correttamente visualizzati ; il problema è dovuto al browser od al font utilizzato.

2.2.2.2.1 **Riepilogo costanti figurative wi_**

spazio bianco/blank	?&b? :
triangolo dx	?&t1? : ►
triangolo sx	?&t2? : ◄
triangolo su	?&t3? : ▲
triangolo giù	?&t4? : ▼
quadrato pieno	?&q? : ■
quadrato vuoto	?&q0? : □
punto	?&p? : ●
smile	?&s? : ☺
nota musicale	?&n? : ♪
micro	?&m? : μ
marchio registrato	?&r? : ®
copyright	?&c? : ©
diametro/zero barrato	?&d? : Ø
omega/ohm	?&o? : Ω
radice quadrata/spunta	?&k? : √
infinito	?&i? : ∞
sigma/sommatoria	?&g? : Σ
simbolo fi	?&f? : φ

N.B. : Nell'installazione di base è presente il modulo wi_modsys che riporta tutte le costanti figurative.

2.2.3 Variabili PHP in wi_

É possibile inserire il contenuto di variabili PHP in un modulo wi_ ;
ciò, essenzialmente, serve per interagire con i contenuti attivi
dell'applicazione (*vedi script PHP 2.3.2*).

Una variabile PHP in un modulo wi_ sarà utilizzabile con la
seguente sintassi :

?\$nome_variabibile_php?

Esempio :

```
<input type=text name="utente" value="?$utente?">
```

Array php

```
?$nod04[xl]?
```

Indica l'elemento con indice "xl" dell' array php \$nod04

N.B. : Per inserire il carattere ASCII (34) – doppie virgolette
(") – occorrerà inserire nella variabile PHP il doppio carattere
ASCII (94) – accento circonflesso (^^) -

2.3 Le pagine wi_

La pagina wi_ corrisponde alla classica pagina web visualizzata dal browser.

Caratteristiche:

- una pagina wi_ contiene SOLO moduli wi_ e script PHP e non codice html;
- l'ordine dei moduli inseriti nella pagina non è determinante: è il valore del tag *x= od *y= in essa contenuti che ne determina il posizionamento. È preferibile, però, inserirli nell'ordine posizionale crescente;
- l'ordine degli script PHP inseriti è determinante;
- la pagina visualizzata è indicata al framework wi_ come parametro ?p=(minuscolo) dello script main.php ovvero volendo visualizzare la pagina wi_pag08, l'indirizzo indicato nel browser sarà:

http://nomehost/main.php?p=wi_pag08

- la pagina di default del framework wi_ è la wi_pag00 ovvero se non si indica alcun parametro ?p=, l'indirizzo equivalente sarà:

main.php?p=wi_pag00

Le pagine wi_ , oltre ai nomi di moduli indicati esplicitamente, possono contenere espressioni di controllo che determinano quale modulo attivare a seconda della condizione verificatasi. Le espressioni di controllo hanno come parametri costanti e variabili di sistema wi_ .

Un esempio classico è la possibilità di attivare la pagina principale in lingua italiana se l'ip del cliente proviene dall'italia od in lingua inglese se proviene da una nazione diversa dall'italia.

Esempio :

La pagina di default index.html effettua un redirect in javascript alla pagina wi_pagini

```
<html>

  <head>
    <title>WebIbrix</title>
  </head>

  <body>

    <SCRIPT LANGUAGE="JavaScript">window.location="main.php?p=wi_pagini"</SCRIPT>

  </body>
</html>
```

La pagina wi_pagini (contenuta nel file moduli/wi_pagini.txt) conterrà il seguente codice :

```
&ifeq(?ipgeo?,ITALY) &then(wi_modini) &else(wi_modinie)
```

*che significa che se la nazione dell'ip che richiede la pagina (variabile **?ipgeo?**) è uguale alla costante **ITALY** sarà caricato il modulo wi_modini altrimenti sarà caricato il modulo wi_modinie .*

Ovviamente il modulo wi_modini (contenuto nel file moduli/wi_modini.txt) effettuerà un redirect javascript alla pagina principale in italiano :

```
*mod=
<SCRIPT LANGUAGE="JavaScript">window.location="./main.php?p=wi_pag00"</SCRIPT>
```

Mentre il modulo wi_modinie (contenuto nel file moduli/wi_modinie.txt) effettuerà un redirect javascript alla pagina principale in inglese :

```
*mod=
<SCRIPT LANGUAGE="JavaScript">window.location="./main.php?p=wi_pag00e"</SCRIPT>
```

N.B. : Nell'installazione di base è presente la pagina index.html, la pagina wi_pagini ed i moduli wi_modini e wi_modinie .

2.3.1 Espressioni di controllo

Ambito di utilizzo : **pagine**

Le espressioni di controllo determinano, in una pagina *wi_*, quale modulo debba essere caricato, effettuando confronti tra variabili di sistema *wi_* e costanti .

Norme sintattiche :

- una espressione di controllo *wi_* è scritta in questa forma :

&ifxx(?variabile di sistema?,costante da verificare)
&then(modulo *wi_* da eseguire se la condizione è verificata)
&else(modulo *wi_* da eseguire se la condizione NON è verificata)

- le espressioni di controllo possono essere scritte in minuscolo, maiuscolo o misto minisclo/maiuscolo;
- le espressioni di controllo **&ifxx** possono essere più di una per confronto e si comportano, in tal caso, in modalità AND ; non è ammesso l'OR tra **&ifxx** nello stesso confronto; ovviamente l'OR si potrà ottenere con confronti successivi;
- un'espressione di controllo *wi_* deve sempre iniziare con una **&** (*e commerciale*) e terminare con una **)** (*parentesi chiusa*) ;
- l'espressione di controllo **&else()** è facoltativa mentre la **&then()** è obbligatoria;
- i termini di confronto sono sempre solo due e separati da una virgola; l'ordine di indicazione dei termini è indifferente (si può indicare prima la variabile di sistema come la costante).

2.3.1.1 Riepilogo espressioni di controllo

Espressione di controllo	Funzione
&ifeq(,)	Attiva &then se primo termine =al secondo termine
&ifne(,)	Attiva &then se primo termine != (diverso) dal secondo termine
&ifgt(,)	Attiva &then se primo termine > del secondo termine
&ifge(,)	Attiva &then se primo termine >= del secondo termine
&iflt(,)	Attiva &then se primo termine < del secondo termine
&ifle(,)	Attiva &then se primo termine <= del secondo termine
&then()	Indica il modulo <i>wi_</i> attivato dalla condizione verificata
&else()	Indica il modulo <i>wi_</i> attivato dalla condizione NON verificata

2.3.2 **Inclusione di script PHP in pagine wi_**

É possibile inserire script PHP nelle pagine wi_ utilizzando la seguente sintassi :

wi_php=nome_script_php comprensivo di estensione
(normalmente **.php**)

- si possono mettere più script in una pagina wi_ ma ne può essere indicato solo uno per riga;
- l'estensione deve essere compresa nello script mentre non deve essere indicato il percorso che è normalmente.

../script

- l'utilizzo degli script PHP presuppone una buona conoscenza di questo linguaggio.

3. **Link ipertestuali**

I link ipertestuali sono inseriti SOLO nei moduli come codice html, utilizzandone la sintassi.

Quando un modulo wi_ contiene un link che porta ad una pagina wi_, la sintassi sarà quella del framework wi_, ovvero:

main.php?p=nomepaginawi_

Esempio :

```
*x=5
*y=5
*mod=

<A HREF= "main.php ?p=wi_pag00" title="pagina principale" style="text_decoration: none">

  Clicca qui per tornare alla pagina principale

</A>
```

Esempio per l'invio di e-mail :

```
<A HREF="mailto:centrodata@centrodata.com">

  invia e-mail a : centrodata@centrodata.com

</A>
```

4. Debug

wi_ dispone di un **debug** estremamente funzionale.

Il debug è possibile in due modalità:

- *direttamente sul framework nella visualizzazione della pagina completa;*
- *con una apposita applicazione che permette di visualizzare un singolo modulo o tutta una pagina.*

4.1. Debug da framework wi_

Questa funzione permette di visualizzare i moduli attivi nella pagina e la loro delimitazione.

Per attivare questa funzionalità è sufficiente inserire la variabile "&debug" nell'indirizzo della pagina.

La variabile "&debug" può assumere due valori :

- **1** per visualizzare solo i nomi dei moduli nella pagina;
- **2** per visualizzare i nomi ed i bordi di moduli nella pagina.

Esempi :

`http://nomehost/main.php?p=wi_pag08&debug=1`

visualizzerà i nomi dei moduli presenti nella pagina wi_pag08

`http://nomehost/main.php?p=wi_pag04&debug=2`

visualizzerà i nomi dei moduli, ed i relativi bordi, presenti nella pagina wi_pag04

Attenzione : se il debug viene effettuato sulla pagina principale del framework wi_, la variabile sarà indicata con "?debug" e non con "&debug".

Esempio :

`http://nomehost/main.php?debug=2`

visualizzerà i nomi dei moduli e dei relativi bordi della pagina principale di default wi_ ovvero la pagina wi_pag00

4.2. Applicazione debug.php

L'applicazione si attiva immettendo nel browser l'indirizzo :

http://nomehost/debug.php

sarà sufficiente indicare il nome del modulo o della pagina per poterlo visualizzare.

Attenzione : l'applicazione di debug wi_ visualizza i moduli e le pagine SENZA la funzione di autocentratura per risoluzioni video maggiori di 1024 pixel e SENZA il colore / o l'immagine di sfondo.

N.B. : Nell'installazione di base è presente il modulo wi_modsys che riporta tutte le variabili di sistema e le costanti figurative.

Appendice A

Html, Css, Javascript

Riepilogo tag html più utilizzati

I documenti HTML consistono di testo cosparso di direttive markup, chiamate **tag**. Un tag non è altro che l'istruzione al browser di come deve visualizzare il "testo".

Un tag è una direttiva racchiusa tra i segni di minore e maggiore ed indifferentemente in maiuscolo o minuscolo. Ogni tag è costituito da un tag di apertura, `<...>`, ed uno di chiusura, `<...>`. Ne esistono comunque taluni che non richiedono quello di chiusura essendo evidente la parte di testo a cui si riferiscono.

Un tag può contenere, oppure no, un numero imprecisato di attributi, purché consistenti tra loro, che ne specificano meglio le caratteristiche.

Il tag `<b>...</b>` racchiude il testo in grassetto (bold).

Sintassi

```
<b>
Questo testo è in grassetto
</b>
```

**
**

Il tag `<br>` inserisce un ritorno a capo nel testo (un'interruzione di riga). È utile notare il fatto che normalmente il browser è in grado di inserire automaticamente i ritorni a capo basandosi sulla dimensione della finestra; per questo motivo, il tag `<br>` deve essere utilizzato per esplicitare un ritorno a capo forzato.

Sintassi

```
<br>
non richiede chiusura
```

<div>

Il tag `<div>` ha la funzione di impostare nel documento una determinata area. L'attributo **align** indica l'allineamento orizzontale; può assumere i valori **left** (a sinistra, valore predefinito), **center** (al centro), **right** (a destra).

Sintassi

```
<div align="justify">
Contenuto del testo
</div>
```

Il tag ** consente di specificare il tipo di carattere da utilizzare per visualizzare il testo.

Sintassi

```
<font color="colore"  
face="nome del font"  
size="dimensione del carattere"  
>/font>
```

L'attributo **font color** indica il colore del testo. L'attributo **face** indica l'elenco in ordine di preferenza dei tipi di carattere utilizzabili per visualizzare il testo. L'attributo **size** indica le dimensioni del carattere.

<i>

Il tag *<i>* consente di inserire testo in corsivo nella pagina.

Sintassi

```
<i>  
Contentuto  
</i>
```

Il tag ** ha la funzione di includere immagini all'interno della pagina.

Sintassi

```
  
non richiede chiusura
```

Attenzione : l'attributo **src** è OBBLIGATORIO in quanto indica l'immagine da visualizzare. Gli attributi **title** e **alt** sono equivalenti con le seguenti differenze:

- **title** visualizza un testo al passaggio del mouse;
- **alt** visualizza un testo al passaggio del mouse ed in caso di impossibilità di visualizzazione dell'immagine.

N.B. I tipi di immagini visualizzabili nelle pagine html sono:

- .jpg (.jpeg)
- .png
- .gif

<a>

Il tag <a> definisce un collegamento all'interno del documento ed ha una sintassi molto articolata.

Sintassi

```
<a href="URL">
collegamento
</a>
```

L'attributo href indica l'URL della risorsa collegata. Per collegamento si intende una qualunque risorsa utilizzabile allo scopo (URL remota, collegamento all'interno del documento e così via).

Il collegamento può essere, ad esempio, ad un'altra pagina web :

```
<A HREF="http://www.paginaweb.it">          </A>
```

o ad un'altra pagina wi_ :

```
<A HREF="http://nomehost/main.php?p=wi_pag08">          </A>
```

L'attributo target="_blank" o target="blank" indica che si aprirà la pagina in una nuova finestra del browser.

Il tag ha la funzione di raggruppare logicamente una porzione di testo a cui preferibilmente assegnare attributi di stile CSS.

Sintassi

```
<span>
</span>
```

CSS

È inoltre possibile, e consigliato, inserire gli elementi CSS (o fogli di stile); questi saranno preferibilmente inseriti in un tag <div> o ma anche nelle ancore dei link ipertestuali.

Sintassi

`style=" parametro : valore; "`

Esempi :

```
<A HREF="/moduli/pdf/pdf_mod15_01.pdf" title="apri .pdf" target="_blank" style="text-decoration: none">

</A>
```

In questo esempio è stata tolta la sottolineatura del link.

```
<A HREF="/moduli/pdf/pdf_mod19_03.pdf" title="apri .pdf" target="_blank" style="text-decoration: none">

</A>
```

In questo esempio l'immagine btn_pdf.png è allineata con la baseline del testo.

JAVASCRIPT

Illustrare un linguaggio di scripting quale è javascript esula dagli scopi di questo manuale.

In questa sede verranno presentati alcuni esempi per la soluzione delle problematiche più comuni.

N.B. : Gli script javascript iniziano con il tag

```
<SCRIPT type="text/javascript">
```

e terminano con

```
</SCRIPT>
```

Ogni riga dello script deve terminare con un punto e virgola (;).

Esempio :

Reindirizzamento automatico ad altra pagina web

```
*x=250
*y=100
*h=
*w=600
*bcol=white
*col=black

*mod=

<script type="text/javascript">
window.location="http://www.centrodata.com";
</script>
```

Esempio :

Emissione finestra messaggio di avvertimento

```
*x=250
*y=100
*h=
*w=600
*bcol=white
*col=black

*mod=

<script type="text/javascript">
window.alert("Sono le ore ?hh?:?hm?:?hs? di ?dw? ?dd? ?my? ?yyyy?");
</script>
```

Appendice B

I colori html

I colori in HTML si possono esprimere in due modi:

- in formato **RGB** (***R**ed **G**reen **B**lue*);
- indicandone il nome.

Il primo è sicuramente il più flessibile ed è anche il metodo utilizzato da gran parte degli editor per le elaborazioni delle immagini.

Si basa sulla combinazione dei tre colori fondamentali Rosso, Verde e Blu per ottenere un qualsiasi colore nella gamma di quelli disponibili. A ciascuno di questi tre colori è possibile assegnare un valore esadecimale compreso tra "00" e "FF" (corrispondenti a 0 e 255) combinandoli in modo da ottenere la tonalità desiderata. Si ottengono in questo modo dei codici di sei cifre che vanno sempre preceduti dal simbolo # e compresi tra apici.

La seconda possibilità, non compatibile per tutti i browser, permette di indicare direttamente il colore desiderato (naturalmente in inglese).

Si riportano di seguito i codici esadecimali e i nomi dei 126 colori fondamentali (per poterli visualizzare la scheda grafica deve supportare almeno 256 colori).

	"#A0CE00"	"ALICEBLUE"		"#FAEBD7"	"ANTIQUWHITE"
	"#00FFFF"	"AQUA"		"#7FFFD4"	"AQUAMARINE"
	"#F0FFFF"	"AZURE"		"#F5F5DC"	"BEIGE"
	"#FFE4C4"	"BISQUE"		"#000000"	"BLACK"
	"#FFEB3D"	"BLANCHEDALMOND"		"#0000FF"	"BLUE"
	"#8A2BE2"	"BLUEVIOLET"		"#A52A2A"	"BROWN"
	"#DEB887"	"BURLYWOOD"		"#5F9EA0"	"CADETBLUE"
	"#7FFF00"	"CHARTREUSE"		"#D2691E"	"CHOCOLATE"
	"#FF7F50"	"CORAL"		"#6495ED"	"CORNFLOWERBLUE"
	"#FFF8DC"	"CORN Silk"		"#DC143C"	"CRIMSON"
	"#00FFFF"	"CYAN"		"#00008B"	"DARKBLUE"
	"#483D8B"	"DARKSLATEBLUE"		"#008B8B"	"DARKCYAN"
	"#B8860B"	"DARKGOLDENROD"		"#A9A9A9"	"DARKGRAY"
	"#FF1493"	"DEEPPINK"		"#00BFFF"	"DEEPSKYBLUE"
	"#696969"	"DIMGRAY"		"#1E90FF"	"DODGERBLUE"
	"#822222"	"FIREBRICK"		"#FFFAF0"	"FLORALWHITE"
	"#228B22"	"FORESTGREEN"		"#FF00FF"	"FUCHSIA"
	"#DCDCDC"	"GAINSBORO"		"#F8F8FF"	"GHOSTWHITE"
	"#FFD700"	"GOLD"		"#DAA520"	"GOLDENROD"
	"#808080"	"GRAY"		"#008000"	"GREEN"
	"#ADFF2F"	"GREENYELLOW"		"#F0FFF0"	"HONEYDEW"
	"#FF69B4"	"HOTPINK"		"#CD5C5C"	"INDIANRED"
	"#4B0082"	"INDIGO"		"#FFFFFF0"	"IVORY"
	"#F0E68C"	"KHAKY"		"#E6E6FA"	"LAVENDER"
	"#FFF0F5"	"LAVENDERBLUSH"		"#FFFACD"	"LEMONCHIFFON"
	"#ADD8E6"	"LIGHTBLUE"		"#F08080"	"LIGHTCORAL"
	"#E0FFFF"	"LIGHTCYAN"		"#FAFAD2"	"LIGHTGOLDENRODYEL LOW"
	"#90EE90"	"LIGHTGREEN"		"#D3D3D3"	"LIGHTGRAY"
	"#FFB6C1"	"LIGHTPINK"		"#FFA07A"	"LIGHTSALMON"
	"#20B2AA"	"LIGHTSEAGREEN"		"#87CEFA"	"LIGHTSKYBLUE"
	"#778899"	"LIGHTSLATEGRAY"		"#B0C4DE"	"LIGHTSTEELBLUE"
	"#FFFFE0"	"LIGHTYELLOW"		"#00FF00"	"LIME"
	"#32CD32"	"LIMEGREEN"		"#FAF0E6"	"LINEN"
	"#FF00FF"	"MAGENTA"		"#800000"	"MAROON"
	"#66CDAA"	"MEDIUMAQUAMARINE"		"#0000CD"	"MEDIUMBLUE"
	"#BA55D3"	"MEDIUMMORCHID"		"#9370DB"	"MEDIUMPURPLE"
	"#3CB371"	"MEDIUMSEAGREEN"		"#7B68EE"	"MEDIUMSLATEBLUE"
	"#00FA9A"	"MEDIUMSPRINGGREEN"		"#48D1CC"	"MEDIUMTORQUOISE"
	"#C71585"	"MEDIUMVIOLETRED"		"#191970"	"MIDNIGHTBLUE"
	"#F5FFFA"	"MINTCREAM"		"#FFE4E1"	"MISTYROSE"
	"#FFDEAD"	"NAVAJOWHITE"		"#000080"	"NAVY"
	"#FDF5E6"	"OLDLACE"		"#808000"	"OLIVE"
	"#6B8E23"	"OLIVEDRAB"		"#FFA500"	"ORANGE"

	"#FF4500"	"ORANGERED"		"#DA70D6"	"ORCHID"
	"#EEE8AA"	"PALEGOLDENROD"		"#98FB98"	"PALEGREEN"
	"#AFEEEE"	"PALETURQUOISE"		"#DB7093"	"PALEVIOLETRED"
	"#FFEDD5"	"PAPAYAWHIP"		"#FFDAB9"	"PEACHPUFF"
	"#CD853F"	"PERU"		"#FFC0CB"	"PINK"
	"#DDA0DD"	"PLUM"		"#B0E0E6"	"POWDERBLUE"
	"#800080"	"PURPLE"		"#FF0000"	"RED"
	"#BC8F8F"	"ROSYBROWN"		"#4169E1"	"ROYALBLUE"
	"#8B4513"	"SADDLEBROWN"		"#FA8072"	"SALMON"
	"#F4A460"	"SANDYBROWN"		"#2E8B57"	"SEAGREEN"
	"#FFF5EE"	"SEASHELL"		"#A0522D"	"SIENNA"
	"#C0C0C0"	"SILVER"		"#87CEEB"	"SKYBLUE"
	"#6A5ACD"	"SLATEBLUE"		"#708090"	"SLATEGRAY"
	"#FFFAFA"	"SNOW"		"#00FF7F"	"SPRINGGREEN"
	"#4682B4"	"STEELBLUE"		"#D2B48C"	"TAN"
	"#008080"	"TEAL"		"#D8BFD8"	"THISTLE"
	"#FF6347"	"TOMATO"		"#40E0D0"	"TURQUOISE"
	"#EE82EE"	"VIOLET"		"#F5DEB3"	"WHEAT"
	"#FFFFFF"	"WHITE"		"#F5F5F5"	"WHITESMOKE"
	"#FFFF00"	"YELLOW"		"#9ACD32"	"YELLOWGREEN"

Appendice C

Installazione del framework **WebIbrix wi_**

L'installazione consiste nel modificare alcuni parametri per adattarli al nuovo ambiente che si andrà ad utilizzare.

Sono necessari i seguenti passi operativi :

a) modificare lo script \$parm.php nella cartella

service

inserendo i vari parametri d'ambiente.

Dovranno essere inseriti i seguenti valori :

(N.B. : *Inserire i valori tra i doppi apici lasciando il punto e virgola (;) finale*)

\$usr_path="";	nome della cartella sul server che conterrà le pagine
\$usr_owner="";	nome dominio del proprietario delle pagine
\$usr_title="";	titolo delle pagine
\$usr_keywords="";	parole chiave per l'indicizzazione nei motori di ricerca
\$usr_contact="";	indirizzo e-mail per contattare il responsabile del sito
\$usr_datedef="";	mettere a 1 se la versione php richiede il fuso orario di default (valori ammessi 1 / blank ; con 1 viene messo quello di Roma)
\$usr_bgcolor="";	inserire un eventuale colore di sfondo per il framework
\$usr_background="";	inserire un eventuale nome di file grafico quale sfondo del framework

b) creare sul server la cartella da utilizzare ed indicata in precedenza nel parametro \$usr_path;

c) copiare l'ambiente *WebIbrix* *wi_* nella cartella sopra descritta (se in remoto usare l'FTP);

d) creare (se non presente) la sottocartella

moduli

nella cartella principale sopra indicata; **questa cartella conterrà i moduli e le pagine *wi_* dell'utente.**

e) creare (se non presente) la sottocartella

script

nella cartella principale indicata; **questa cartella conterrà gli script PHP dell'utente.**

C.1

Parametri di php.ini richiesti per il funzionamento di *WebIbrix wi_*

OBBLIGATORI

register_globals = On

CONSIGLIATI

display_errors = On

error_reporting = E_ALL & ~E_NOTICE

max_execution_time = (blank)

Appendice D

I nomi delle variabili PHP usate da wi_, e pertanto riservate, iniziano per

`$wi_`

`$ip_geo_`

`$usr_`

Indice generale

1. WebIbrix wi_	1
2. Gli elementi base di wi_	2
2.1. Convenzioni di utilizzo	2
2.2. I moduli wi_	3
2.2.1. I tag wi_	4
2.2.1.1. Riepilogo tag wi_	4
2.2.2. Variabili di sistema wi_	5
2.2.2.1. Riepilogo variabili wi_	7
2.2.2.2. Costanti figurative	8
2.2.2.2.1. Riepilogo costanti figurative wi_	9
2.2.3. Variabili PHP in wi_	10
2.3. Le pagine wi_	11
2.3.1. Espressioni di controllo	13
2.3.1.1. Riepilogo espressioni di controllo	13
2.3.2. Inclusione di script PHP in pagine wi_	14
3. Link ipertestuali	15
4. Debug	16
4.1. Debug da framework wi_	17
4.2. Applicazione debug.php	18

Appendice A

Riepilogo tag HTML più utilizzati	18
CSS	22
JAVASCRIPT	23

Appendice B

I colori HTML	24
---------------	----

Appendice C

Installazione del framework <i>WebIbrix</i> w i _	27
---	----

C.1.

Parametri di php.ini richiesti per il funzionamento di <i>WebIbrix</i> w i _	29
--	----

Appendice D

Nomi variabili PHP riservate	30
------------------------------	----