Appendice B
Tag HTML

In questa appendice è presentata una concisa ed utile illustrazione dei principali tag di formattazione HTML utilizzabili per la produzione di commenti doc. Per ogni tag viene riportata una breve descrizione e un semplice esempio di utilizzo.

&gt

Questa etichetta (*greater-than*) serve per generare il carattere maggiore (">"). Questo non può essere utilizzato direttamente poiché verrebbe interpretato, erroneamente, come delimitatore HTML di chiusura tag.

Esempio > Si consideri il seguente frammento di commento doc:

```
* The translation results in an offset of &lt;2.0, -2.0&gt;.
```

&lt

Questa etichetta (*less-then*) serve per generare il carattere minore ("<"). Questo non può essere utilizzato direttamente poiché verrebbe interpretato, erroneamente, come delimitatore HTML di apertura tag.

Esempio > Si consideri il seguente frammento di commento doc:

```
* The translation results in an offset of &lt;2.0, -2.0&gt;.
```


Spaces. Questa stringa serve per includere un certo numero di spazi bianchi. È necessario ricorrere a questa etichetta poiché la maggior parte dei parser HTML sostituiscono una ripetizione di spazi bianchi con un solo.
Per l'esempio, cfr. <pre>.

<a>

Il tag dell'ancora è utilizzato per inserire degli hyperlink all'interno della propria documentazione: link text .

Esempio > Si consideri il seguente frammento di commento doc:

```
* This interface is a member of the
* <a href="{@docRoot}/../guide/collections/index.html">
* Java Collections Framework</a>.
```


Questi due delimitatori sono utilizzati per riprodurre in carattere neretto (*bold*) il testo incluso.

Esempio > si consideri il seguente frammento di commento doc:

```
* <p><b>Note: It is rarely appropriate to use this constructor.
* Unless a <i>new</i> instance is required, the static factory
* {@link #valueOf(boolean)} is generally a better choice. It is
* likely to yield significantly better space and time performance.</b>
```


Line Break. Questo tag è utilizzato per forzare l'avanzamento di riga ed è necessario perché i parser HTML ignorano i caratteri return.

Esempio > Si consideri il seguente frammento di commento doc:

```
* A thread state.  A thread can be in one of the following states:
* <ul>
* <li>{@link #NEW}<br>
* A thread that has not yet started is in this state.
* </li>
```

<code> </code>

Questi tag sono utilizzati per inserire frammenti di codice utilizzando liberamente elementi, come le parentesi angolari, che altrimenti creano problemi con i tag HTML.

Esempio > Si consideri il seguente frammento di commento doc:

```
* <pre><code>
* // Base GMT offset: +1:00
*    // DST starts:      at 1:00am in UTC time
*    //                on the last Sunday in March
*    // DST ends:        at 1:00am in UTC time
*    //                on the last Sunday in October
*    // Save:            1 hour
*    SimpleTimeZone (3600000,
*                        "Europe/Paris",
*                        Calendar.MARCH, -1, Calendar.SUNDAY,
*                        3600000, SimpleTimeZone.UTC_TIME,
*                        Calendar.OCTOBER, -1, Calendar.SUNDAY,
*                        3600000, SimpleTimeZone.UTC_TIME,
*                        3600000)
* </code></pre>
```

<dl>

<dt>

</dt>

<dd>

</dd>

</dl>

I tag <dl> e </dl> permettono di definire particolari liste formate da item corredati dalla relativa descrizione. Gli item vanno riportati all'interno dei tag <dt> e </dt>, mentre le descrizioni vanno riportate all'interno dei tag <dd> e </dd>.

Esempio > Si consideri il seguente frammento di commento doc:

```
* <DL>
* <DT> read <DD> read permission
* <DT> write <DD> write permission
* <DT> execute
* <DD> execute permission. Allows <code>Runtime.exec</code> to
*         be called. Corresponds to <code>SecurityManager.checkExec</code>.
* <DT> delete
```

```
 * <DD> delete permission. Allows <code>File.delete</code> to
 *          be called. Corresponds to <code>SecurityManager.checkDelete</code>.
 * </DL>
```


Questi due delimitatori sono usati per riprodurre in carattere enfatizzato il testo incluso.
Esempio > Si consideri il seguente frammento di commento doc:

```
 * User interfaces and operating systems use system-dependent <em>pathname
 * strings</em> to name files and directories.  This class presents an
 * abstract, system-independent view of hierarchical pathnames.  An
 * <em>abstract pathname</em> has two components:
```

<h1> </h1>

<h2> </h2>

<h3> </h3>

Le tags <h1> </h1> sono utilizzate per specificare la sezione di header di più alto livello. Oltre a questi, è possibile utilizzare header di livello inferiore come <h2>, <h3>, etc.
Esempio > Si consideri il seguente frammento di commento doc:

```
 *
 * <h3>Memory Synchronization</h3>
 * <p>All <code>Lock</code> implementations <em>must</em> enforce the
 * same memory synchronization semantics as provided by the built-in
 * monitor lock:
```

<HR>

Horizontal Rule. Questa etichetta genera una riga orizzontale nella pagina HTML.
Esempio > Si consideri il seguente frammento di commento doc:

```
 * <p><hr><blockquote><pre>
 * class PrimeThread extends Thread {
 *    long minPrime;
 *    PrimeThread(long minPrime) {
 *       this.minPrime = minPrime;
 *    }
 *
```

```
*    public void run() {
*      // compute primes larger than minPrime
*      &amp;nbsp;.&amp;nbsp;.&amp;nbsp;.
*    }
* }
* </pre></blockquote><hr>
* <p>
* The following code would then create a thread and start it running:
* <p><blockquote><pre>
* PrimeThread p = new PrimeThread(143);
* p.start();
* </pre></blockquote>
* <p>
```


Questi due delimitatori sono utilizzati per riprodurre in carattere italico il testo incluso.
Esempio > Si consideri il seguente frammento di commento doc:

```
* <p> For a given abstract pathname <i>f</i> it is guaranteed that
```


Questo tag singolo permette di includere delle immagini all'interno della documentazione.
<img src="URL immagine">
Esempio > Si consideri il seguente frammento di commento doc:

```
* <p>
* <img src="doc-files/List-1.gif"
* alt="Shows a list containing: Venus, Earth, JavaSoft, and Mars. Javasoft
* is selected." ALIGN=center HSPACE=10 VSPACE=7>
```


I tag <ol> e sono utilizzati per creare un elenco puntato ordinato i cui elementi (voci) sono specificati all'interno dei tag <li> e .
Esempio > Si consideri il seguente frammento di commento doc:

```
* <OL>
* <LI> If C declares a public field with the name specified, that is the
*      field to be reflected.</LI>
* <LI> If no field was found in step 1 above, this algorithm is applied
*      recursively to each direct superinterface of C. The direct
*      superinterfaces are searched in the order they were declared.</LI>
* <LI> If no field was found in steps 1 and 2 above, and C has a
*      superclass S, then this algorithm is invoked recursively upon S.
*   If C has no superclass, then a <code>NoSuchFieldException</code>
*   is thrown.</LI>
* </OL>
```


Questi tag sono i demarcatori di paragrafo (inizio e fine rispettivamente) e fanno sì che tutto il testo riportato sia visualizzato in una nuova riga. Per questo motivo, tipicamente, si utilizza la sola etichetta:

Esempio > Si consideri il seguente frammento di commento doc:

```
* The following code would then create a thread and start it running:
* <p><blockquote><pre>
* PrimeThread p = new PrimeThread(143);
* p.start();
* </pre></blockquote>
* <p>
```

<pre> </pre>

Questi due tag indicano porzioni di testo preformattate. Pertanto elementi quali indentazioni e ripetizioni di spazi bianchi, normalmente, ridotte a uno solo dai browser HTML, sono rispettati.

Esempio > Si consideri il seguente frammento di commento doc:

```
* <p><hr><blockquote><pre>
*  class PrimeThread extends Thread {
*    long minPrime;
*    PrimeThread(long minPrime) {
*      this.minPrime = minPrime;
*    }
*
*    public void run() {
*      // compute primes larger than minPrime
```

```
*         . . .
*    }
* }
* </pre></blockquote><hr>
```


Questi due delimitatori sono utilizzati per riprodurre in carattere enfatizzato e neretto il testo incluso.

Esempio > Si consideri il seguente frammento di commento doc:

```
* Returns a synchronized (thread-safe) collection backed by the specified
* collection.  In order to guarantee serial access, it is critical that
* <strong>all</strong> access to the backing collection is accomplished
* through the returned collection.<p>
```

<table>

<tr>

<th>

<th>

</tr>

<tr>

<td>

<td>

</tr>

</table>

I tag <table> **e** </table> permettono di definire una tabella. Questa tabella è organizzata in tante righe quante sono le coppie di tag <tr> e </tr>. Ogni riga poi è organizzata in celle: <td> e </td>. Inoltre è possibile specificare una linea per il titolo (header) con i tag <tr> </tr>.

Esempio > Si consideri il seguente frammento di commento doc:

```
* <blockquote><table summary="Element types and encodings">
```

```
* <tr><th> Element Type <th> Encoding
* <tr><td> boolean              <td align=center> Z
* <tr><td> byte                 <td align=center> B
* <tr><td> char                 <td align=center> C
* <tr><td> class or interface   <td align=center> L<i>classname;</i>
* <tr><td> double               <td align=center> D
* <tr><td> float                <td align=center> F
* <tr><td> int                  <td align=center> I
* <tr><td> long                 <td align=center> J
* <tr><td> short                <td align=center> S
* </table></blockquote>
```

Da notare che spesso in HTML i terminatori di tag sono omessi. Questa però non è una buona norma considerando evoluzioni del tipo XTML.

<tt> </tt>

Teletype font. Questi tag specificano che il testo inserito debba essere riportato in un font teletype.

Spesso questa etichetta viene incorrettamente utilizzata al posto di <code> (cfr.).

```
* Creates a new <tt>File</tt> instance by converting the given
* <tt>file:</tt> URI into an abstract pathname.
```


I tag <ul> e sono utilizzati per creare un elenco puntato (*unordered*, non ordinato) i cui elementi (voci) sono specificati all'interno dei tag <li> e .

Esempio > Si consideri il seguente frammento di commento doc:

```
* <p><ul>
* <li>An <code>ItemListener</code> object is registered
* via <code>addItemListener</code>.
* <li>Item events are enabled via <code>enableEvents</code>.
* </ul>
```