

Linguaggi e Traduttori

a.a. 2005/2006

Tesina n° 6

Titolo: HTML programming

Descrizione

Si vuole realizzare un linguaggio di programmazione che permetta di costruire pagine html. Dato in ingresso un file contenente una serie di istruzioni si vuole ottenere un file in formato html.

Il programma dovrà automaticamente inserire nel file di uscita uno scheletro di pagina html come il seguente:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
</head>

<body>
</body>
</html>

```

Esistono istruzioni che agiscono internamente ai tag `<head>` e `</head>`, altre che agiscono internamente ai tag `<body>` e `</body>` e altre su entrambi i tag.

Un'istruzione è composta dal carattere "+", seguito da un identificatore, da una lista opzionale di parametri racchiusi tra parentesi quadre "[" e "]", da un corpo delimitato dai caratteri di "(" e di ")" e da altri due delimitatori opzionali "{" e "}" che hanno lo scopo di indicare i parametri di un tag html:

```
+<id> ("[/ * lista si parametri * /"])? (/ * corpo *) ("{/ * parametri tag html * /}")?
```

Verranno descritte con esempi le funzioni che compongono il linguaggio.

Si ipotizza che tutte le funzioni che agiscono sui tag `<head>` e `</head>` siano utilizzate sempre prima delle funzioni che agiscono sui tag `<body>` e `</body>`.

Funzioni che agiscono solo sui tag `<head>` e `</head>`:

- **+meta(descrizione,la descrizione)** : La funzione +meta serve per la scrittura dei metadati nel header del file html. Il primo parametro identifica il tipo di metadato, il secondo il contenuto.
output: `<meta name="description" content="la descrizione">`
- **+titolo(titolo della pagina)** : Titolo della pagina html
`<title>titolo della pagina</title>`
- **+css(css/stile.css)** : Indica quale foglio di stile deve essere utilizzato in questa pagina html
`<link rel="stylesheet" type="text/css" href="css/stile.css">`

Funzioni che agiscono solo sui tag `<body>` e `</body>`:

- **+h1(titolo h1)**
- **+h2(titolo h2)**
- **+h3(titolo h3)**
- **+h4(titolo h4)**
- **+h5(titolo h5)**
- **+p(paragrafo)**
- **+pre(tag pre)**

Consentono di rappresentarli i titoli di vario livello, i paragrafi e i tag pre.

```
<h1>titolo h1</h1>
```

```
<h2>titolo h2</h2>
```

```
<h3>titolo h1</h3>
<h4>titolo h1</h4>
<h5>titolo h1</h5>
<p>paragrafo</p>
<pre>tag pre</pre>
```

- **+b(bold)**
+i(italic)
+u(underline)
+center(centrato)
 Tag che servono per formattare il testo: grassetto, corsivo, sottolineato e centrato

```
<b>bold</b>
<i>italic</i>
<u>underline</u>
<center>centrato</center>
```
- **+a** : La funzione per la creazione di ancore (link) può avere uno, due o tre parametri.
+a(http://www.skenz.it)
+a(http://www.skenz.it,SKENZ)
+a(http://www.skenz.it,)
+a(http://www.skenz.it,SKENZ,link a Skenz)
 verranno riscritte come:

```
<a href="http://www.skenz.it">http://www.skenz.it</a>
<a href="http://www.skenz.it">SKENZ</a>
<a href="http://www.skenz.it"></a>
<a href="http://www.skenz.it" title="link a Skenz">SKENZ</a>
```
- **+img**: Tag per la gestione di immagini che possono essere considerate o meno come link, anche in questo caso il numero di parametri non è costante e possono essere o 2 o 3.
+img(http://www.skenz.it/immagine.jpg,testo alternativo)
+img(http://www.skenz.it/immagine.jpg,,http://www.skenz.it)
 riscritte come:

```

<a href="http://www.skenz.it"></a>
```
- **+linea()** : Stampa una linea orizzontale

```
<hr>
```
- **+listaNumeri(**
 +l(a)
 +l(b)
 +l(c)
)
 Serve per creare una lista numerata. Di tag **+l()** ce ne possono essere un numero indefinito. Nota che le liste possono essere anche annidate, quindi all'interno di un tag **+l()** può essere definita un'ulteriore lista (o qualsiasi altro tag).

```
<ol>
  <li>a</li>
  <li>b</li>
  <li>c</li>
</ol>
```
- **+listaPunti(**
 +l(a)
 +l(b)
 +l(c)
)

Serve per definire una lista composta da punti

```
<ul>
  <li>a</li>
  <li>b</li>
  <li>c</li>
</ul>
```

- **+tabella:** Quest'istruzione serve per la definizione di tabelle. Al suo interno esistono una serie di funzioni obbligatorie e alcune facoltative che servono a configurare alcuni parametri della tabella. Un esempio che usa tutte le funzioni obbligatorie, quindi indispensabili, è il seguente:

```
+tabella [2,3](
  1,1(Testo 1\,1)
  1,2(Testo 1\,2)
  1,3(Testo 1\,3)
  2,1(Testo 2\,1)
  2,2(Testo 2\,2)
  2,3(Testo 2\,3)
```

)

In cui come parametro della funzione si passa la dimensione della tabella come numero di righe e di colonne. All'interno del corpo della funzione, si può indicare ogni singola cella con il comando `<numero_riga>,<numero_colonna>(/* contenuto */)`. Questi comandi forniranno il seguente output:

```
<table>
  <tr>
    <td>Testo 1,1</td>
    <td>Testo 1,2</td>
    <td>Testo 1,3</td>
  </tr>
  <tr>
    <td>Testo 2,1</td>
    <td>Testo 2,2</td>
    <td>Testo 2,3</td>
  </tr>
</table>
```

Come anticipato esistono dei comandi opzionali. Prima dell'istruzione

`<numero_riga>,<numero_colonna>(/* contenuto */)` possono essere inseriti i comandi:

+titolo(Tabella): Che indica il titolo della tabella

+descrizione(Una tabella): Che fornisce una descrizione alla tabella

+cellAlign(left): Indica l'opzione di allineamento orizzontale del testo all'interno di ogni cella della tabella.

+cellValign(top): Indica l'opzione di allineamento verticale del testo all'interno di ogni cella della tabella.

Con queste nuove opzioni dando il file di ingresso:

```
+tabella [2,3](
  +titolo(Tabella)
  +descrizione(Una tabella)
  +cellAlign(left)
  +cellValign(top)
  1,1(Testo 1\,1)
  1,2(Testo 1\,2)
  1,3(Testo 1\,3)
  2,1(Testo 2\,1)
```

2,2(Testo 2\,2)

2,3(Testo 2\,3)

)

Si ottiene:

```
<table title="Tabella" summary="Una tabella">
  <tr>
    <td align="left" valige="top">Testo 1,1</td>
    <td align="left" valige="top">Testo 1,2</td>
    <td align="left" valige="top">Testo 1,3</td>
  </tr>
  <tr>
    <td align="left" valige="top">Testo 2,1</td>
    <td align="left" valige="top">Testo 2,2</td>
    <td align="left" valige="top">Testo 2,3</td>
  </tr>
</table>
```

Esiste poi la possibilità di fornire un allineamento diverso, sia verticale che orizzontale, a qualsiasi cella. Si usa l'opzione parametri tag html nel comando

<numero_riga>,<numero_colonna>(/* contenuto */) { /* parametri tag html */}. Dentro i delimitatori { } si possono utilizzare i comandi **+centro()** **+sinistra()** **+destra()** per fornire un allineamento orizzontale rispettivamente centrale, sinistro e destro. E la normale sintassi html **valign="top"** per indicare l'allineamento verticale.

Dato il file di ingresso:

```
+tabella [2,3](
  +titolo(Tabella)
  +descrizione(Una tabella)
  +cellAlign(left)
  +cellValign(top)
  1,1(Testo 1\,1) { +destra() valign="middle"}
  1,2(Testo 1\,2)
  1,3(Testo 1\,3) { +centro() }
  2,1(Testo 2\,1)
  2,2(Testo 2\,2)
  2,3(Testo 2\,3)
)
```

Si ottiene:

```
<table title="Tabella" summary="Una tabella">
  <tr>
    <td align="right" valige="middle">Testo 1,1</td>
    <td align="left" valige="top">Testo 1,2</td>
    <td align="center" valige="top">Testo 1,3</td>
  </tr>
  <tr>
    <td align="left" valige="top">Testo 2,1</td>
    <td align="left" valige="top">Testo 2,2</td>
    <td align="left" valige="top">Testo 2,3</td>
  </tr>
</table>
```

Funzioni che agiscono in ogni punto del file html:

- **+commento(commento1)** : Funzione di commento
<!-- commento 1-->

- **+include(file.txt)** : Esegue l'inclusione del file "file.txt" nel file html generato. Praticamente apre il file "file.txt", lo legge e lo include nel file.

All'interno del file può comunque essere scritto qualsiasi testo. Se si vuole ad esempio scrivere **+include()** nel file di ingresso basterà mettere il carattere "\" davanti al comando **\+include()**. Facendo in questo modo lo scanner anziché andare in uno stato esclusivo di riconoscimento dei comandi, stampa il carattere + come un normale carattere di testo e quindi non riconosce **include()** come un comando.

Bisogna prevedere la possibilità di utilizzare anche i caratteri riservati. Ad esempio **+a(http://www.skenz.it,a,b)** è interpretato come **a**, ma se scrivo **+a(http://www.skenz.it,a\b)** deve essere così interpretato: **a,b**

Si è parlato dell'opzione **{ /* parametri tag html */ }**. Tale opzione può essere presente in qualsiasi comando e non fa nient'altro che scrivere il testo presente tra { e } all'interno del tag di apertura rappresentato dal comando:

```
+h1(tag h1){ title="titolo" }  
<h1 title="titolo">tag h1</h1>
```

All'interno di **{ /* parametri tag html */ }** possono essere messi i comandi **+centro()** **+sinistra()** **+destra()** che hanno esattamente lo stesso significato che avevano nelle tabelle e cioè di scrivere **align="center"** o **align="left"** o **align="right"**.

All'interno del documento di testo possono essere inseriti un qualsiasi numero di commenti in stile C che non verranno riscritti nel file html generato.

Come ultima caratteristica, un istruzione può essere seguita da un punto più un nome. In tal caso all'interno del tag di apertura verrà riportata **class="<nome>"**.

```
+h1.write(tag h1){ title="titolo" }  
<h1 class="write" title="titolo">tag h1</h1>
```

Per questa tesina non è richiesta relazione. Dovranno essere allegati dei **file di esempio** per verificare il corretto funzionamento del programma.