

**Giorno 11**

# **XML Pointer Language (XPointer)**

XML Pointer Language (XPointer) offre gli strumenti per indirizzare e localizzare informazioni che si trovano all'interno di un altro documento XML. XPointer usa (ed estende) XPath, il linguaggio che abbiamo studiato nel Giorno 9. Combina espressioni XPath e URI: abbiamo già esplorato entrambe queste tecnologie, perciò qui ci concentreremo su come possono incorporate in un'espressione XPointer. Oggi vedremo:

- Come XPointer permetta di migliorare il metodo delle àncore con nome per l'identificazione di frammenti, tipico di HTML
- La sintassi delle espressioni XPointer
- Vari operandi speciali di XPointer
- Come usare le espressioni delle relazioni XPath in XPointer
- Varie caratteristiche speciali di XPointer

## **I “puntatori” di HTML**

Le specifiche di XPointer, working draft del W3C datato 9 luglio 1999, descrivono un'espressione basata su stringhe che permette l'indirizzamento diretto delle strutture interne di documenti XML. Le stringhe di XPointer permettono di fare riferimento diretto a elementi, stringhe di caratteri e altre parti di documenti XML perché sono basate sull'espressività e la selettività del linguaggio XPath. Questa è una caratteristica di XPointer che lo distingue nettamente dal meccanismo di localizzazione di HTML.



*Se non l'avete ancora fatto, è bene che leggiate le lezioni su XPath (Giorno 9) e XLink (Giorno 10) prima di studiare i materiali di questa lezione, che in parte riprendono ed estendono alcune fra le caratteristiche di quelle tecnologie e ne presuppongono quindi una conoscenza di base.*

## 218 Giorno 11

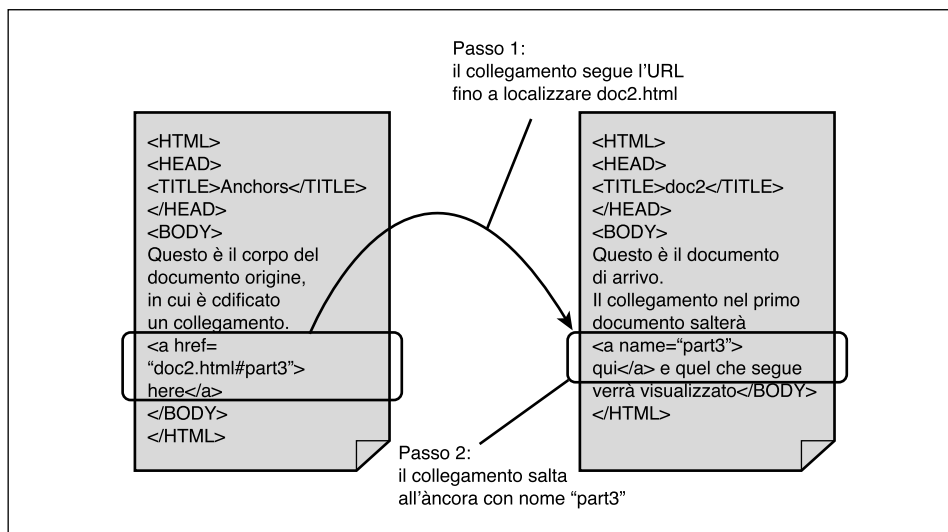
Vediamo, per avere un punto riferimento, come si “punta”, fondamentalmente, in HTML. Per localizzare un sottoinsieme, un frammento di una pagina HTML bisogna che in una posizione specifica sia stato fissato un nome, prima dell’interrogazione, e che il documento chiamante indirizzi specificamente quel collegamento con nome. In HTML questo procedimento serve per localizzare frammenti con nome nello stesso documento o in documenti diversi. La sintassi dell’etichetta che attribuisce un nome al punto di arrivo è questa:

```
<a name="nome_destinazione">Testo o oggetto di destinazione</a>
```

Così si definisce un indicatore (un *flag*) nel punto in cui sta l’ancora sulla pagina HTML. L’indicatore può essere in cima o in un punto qualsiasi del markup della pagina. Il valore dell’attributo `name` è associato con questo frammento selezionato della pagina HTML e fornisce un indirizzo. Un marcatore ancora separato che si riferisca esplicitamente al valore dell’attributo può risolvere l’indirizzo. La sintassi del marcatore ancora che effettua la chiamata è la seguente:

```
<a href="URI#nome_destinazione">Testo o oggetto di collegamento</a>
```

Il collegamento ipertestuale creato da questo marcatore punta al frammento con nome sulla pagina all’URI indicato. Quando l’utente attiva il collegamento con un clic del mouse sul testo o l’oggetto di collegamento visualizzati nella finestra del browser, quest’ultimo scarica il frammento di contenuto che si trova nella posizione di destinazione con quel nome e lo presenta al posto della pagina corrente. La Figura 11.1 rappresenta questa funzione in HTML.



**Figura 11.1**

Un collegamento ipertestuale HTML da un documento a una destinazione dotata di nome in un altro documento.



## Limiti delle àncore HTML

Questo metodo ha dei limiti. Innanzitutto, bisogna sapere il nome di tutte le destinazioni che servono. Se siete voi gli autori della pagina di destinazione, può essere una cosa abbastanza semplice, ma se dovete accedere a una pagina remota che non è una vostra creazione, il problema di sapere i nomi delle destinazioni può non essere banale. Ovviamente, tanto per cominciare, perché questo metodo funzioni bisogna che le destinazioni esistano; altrimenti, non si può fare riferimento a un'àncora con nome per identificare un frammento di pagina. Questo vuol dire che uno sviluppatore HTML deve essere stato abbastanza previdente da mettere àncore con nome nei punti utili sulla sua pagina.

Un altro problema è quello della sicurezza. Supponiamo che vogliate accedere a frammenti di una pagina HTML che non hanno ancora avuto un nome. Se volete creare le opportune àncore con nome, dovete avere tutti i diritti e i privilegi necessari per accedere a quella pagina e modificarla, sul server che ne è l'host. Le pagine Web statiche sono documenti di sola lettura; senza diritti di accesso all'host, non è possibile modificarle. Normalmente è possibile vedere il codice sorgente di una pagina Web di sola lettura e stabilire se già vi esistono punti di ancoraggio con nome, ma senza diritti di accesso adeguati non è possibile modificare la pagina HTML di sola lettura. XPointer dà agli autori XML la possibilità di accedere a frammenti di document XML di sola lettura.

## Espressioni XPointer

XPointer usa il linguaggio di espressioni XPath, che abbiamo studiato il Giorno 9, e lo estende notevolmente. XPointer in effetti è costituito da espressioni XPath in combinazione con URI qualificati, e permette di

- Indirizzare intervalli e nodi in un documento di destinazione
- Localizzare informazioni in una risorsa di destinazione mediante ricerca di stringhe
- Utilizzare espressioni di indirizzamento negli URI come identificatori di frammenti

Con la sua ricchezza espressiva, XPointer permette di localizzare informazioni navigando nella struttura di un documento. In questo modo, si possono selezionare frammenti sulla base di proprietà come tipi di elementi, valori di attributo, contenuto a caratteri, posizione relativa, ordine: si ha a disposizione una capacità di localizzazione molto potente, e una granularità molto più fine di quella offerta dal solo XPath. Quest'ultimo può localizzare nodi in un documento, mentre XPointer può localizzare nodi e pattern nei contenuti degli elementi, e può addirittura fornire l'indirizzo di un singolo carattere in un documento.



## Sintassi di XPointer

XPointer usa uno schema di indirizzamento URI ampliato, non molto diverso da quello usato dai marcatori àncora di HTML. La sintassi di una espressione XPointer è

*URI#schema(espressione)*

Quando si creano puntatori XPointer, si mette un URI, seguito da un segno die-sis o cancelletto (#) e da una parte frammento XPointer, costituita da uno schema e un'espressione. L'URI in questa stringa è separato dalla parte frammento (schema più espressione) dal cancelletto. L'URI ha la responsabilità di localizzare una risorsa (per lo più un documento) e l'espressione XPointer fornisce l'indirizzo di un frammento specifico entro il documento identificato.

In una espressione XPointer possono esserci anche più parti frammento, nel qual caso la sintassi è

*URI#schema(espressione) schema(espressione) schema(espressione)...*

L'unico schema definito per ora è `xpointer`, il quale dice al processore che si sta usando il linguaggio XPath come linguaggio di indirizzamento. Forse in futuro ci saranno altri linguaggi in grado di estendere questa sintassi oltre XPointer, e allora lo schema potrà essere codificato in qualche altro modo.

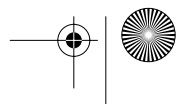
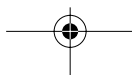
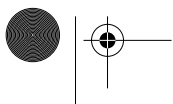
Normalmente si concatenano più parti frammento solo quando la struttura, la validità o la natura del documento di destinazione sono in dubbio, perché XPointer viene risolto leggendo la stringa dell'espressione da sinistra verso destra e il processore smetterà di leggere la stringa non appena un'espressione di percorso di frammento viene soddisfatta. In XPath le cose vanno diversamente, perché viene sempre interpretata un'intera espressione. Con XPointer, si può inserire una parte frammento e poi un'espressione alternativa, per l'eventualità che la prima fallisca.

Dato che XPointer è l'unico schema descritto al momento dalle specifiche, è accettabile una versione accorciata della sintassi:

*URI#espressione*

## Riferimenti ID XPointer

Un modo di usare XPointer consiste nel dare un attributo `id` all'elemento a cui si vuol puntare. Così si duplica il metodo della destinazione dotata di nome tipico di HTML: bisogna però essere in grado di predisporre anticipatamente i puntatori. Se il processore incontra una stringa semplice dopo il segno di cancelletto, senza ulteriori specificazioni, assume che si riferisca a un elemento con quell'ID. Per esempio, riprendendo il sistema di elaborazione di messaggi usato varie volte nelle giornate precedenti, supponiamo di voler trovare un messaggio





## XML Pointer Language (XPointer) 221

specifico, marcato con un particolare numero ID. Supponiamo che al momento nel documento XML ci siano tre messaggi, e che si voglia usare un'espressione XPointer per trovare il primo, identificato da un attributo ID dell'elemento `messaggio` con valore `m1`. Il Listato 11.1 è il documento istanza XML.

**Listato 11.1** Istanza XML con attributi ID per gli elementi `messaggio`: `message01.xml`.

```
1: <?xml version = "1.0"?>
2: <!-- Listato 11.1 - message01.xml -->
3:
4: <nota>
5:   <messaggio id="m1" da="Kathy Shepherd">
6:     Ricordati di comprare il latte tornando a casa
7:   </messaggio>
8:   <messaggio id="m2" da="Greg Shepherd">
9:     Ho bisogno di aiuto per i compiti a casa
10:  </messaggio>
11:  <messaggio id="m3" da="Kristen Shepherd">
12:    Gioca a scacchi con me stasera
13:  </messaggio>
14: </nota>
```

Nelle righe 5, 8 e 11 sono stati aggiunti attributi `id` agli elementi `messaggio`. Per localizzare l'elemento `messaggio` con attributo `id="m1"`, si può usare la forma abbreviata dell'espressione XPointer:

```
message01.xml#m1
```

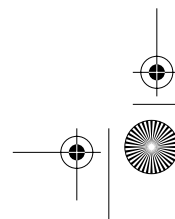
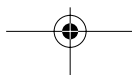
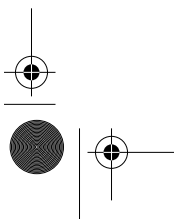
Questa espressione localizzerà l'elemento `messaggio` nella riga 5 del Listato 11.1. La forma più prolissa di questa espressione è

```
message01.xml#ID(m1)
```

Le due forme dell'espressione XPointer funzionano nello stesso modo. La versione più prolissa è coerente con la sintassi di altre espressioni XPointer, che in genere sono formate da una parola chiave di qualche tipo, seguita da parentesi che racchiudono un valore o un argomento.

### Indirizzamento assoluto

Nell'esempio precedente, il termine di localizzazione `id()` è in grado di localizzare un elemento specifico senza ulteriori aiuti. Questo è un esempio di termine di localizzazione assoluta. XPointer ammette quattro di questi termini speciali, che seguono il segno di cancelletto in una stringa di espressione, ma non possono essere combinati. In altre parole, in ogni espressione è ammesso un solo termine speciale. La Tabella 11.1 riassume questi termini.





## 222 Giorno 11

**Tabella 11.1** Termini di localizzazione assoluta di XPointer.

Termine	Descrizione
id()	Localizza qualsiasi elemento con attributo <code>id</code> che risolve il valore specificato dall'espressione
root()	Localizza l'elemento radice del documento specificato da un URI
html()	Localizza l'attributo <code>&lt;A&gt;</code> in una pagina HTML che è anche XML ben formato (cioè XHTML), se il valore di quell'attributo risolve le specifiche dell'espressione
origin()	Localizza un elemento radice di documento che è simile a <code>root()</code> , ma solo da un riferimento relativo

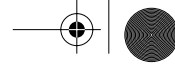
## Espressioni relazionali

Le espressioni relazionali, o termini di localizzazione relativa, indirizzano la posizione di un elemento sulla base di un nodo contesto. In altre parole, l'espressione dà un risultato diverso a seconda del punto da cui inizia il percorso. Abbiamo parlato di relazioni e assi XPath nel Giorno 9: usandoli insieme con un URI, separato da un segno di cancelletto, si ottiene un'espressione XPointer relativa. Le espressioni relazionali funzionano in XPointer come in XPath. Come ricorderete, esistono parecchi assi di relazione, ma non tutti hanno un valore in XPointer. Quelli rilevanti sono:

- Child
- Ancestor
- Descendent
- Following
- Preceding
- Psibling (corrispondente a preceding-sibling)
- Fsibling (corrispondente a following-sibling)

Perché alcuni assi XPath, come `parent` e `descendent-or-self`, non sono presenti nell'elenco precedente? Perché XPointer permette di aggiungere argomenti ai termini di localizzazione relativa dell'elenco, il che permette di selezionare effettivamente qualsiasi asse standard di XPath. Se per esempio, si sceglie il primo antenato relativamente al punto contesto, si sceglie effettivamente il genitore di quel contesto.

Ciascuno dei termini di localizzazione relativa accetta fino a quattro argomenti per una migliore definizione dell'espressione. Questi argomenti consentono di aumentare la sensibilità dell'espressione XPointer. Hanno lo stesso effetto sulla specificità di ciascuna relazione e sono caratterizzati dalla loro posizione assoluta fra le parentesi accodate all'espressione relazionale. Gli argomenti sono descritti brevemente nei prossimi paragrafi: sono espressioni XPath tipiche dell'uso di XPointer, perciò dovrete vedere qualche somiglianza fra questi



esempi e i tipi di espressioni che abbiamo creato nel Giorno 9. Comunque daremo esempi di stili diversi, per precisare la logica condivisa da XPath e XPointer e per farvi esercitare un po' di più nella lettura e nell'interpretazione delle espressioni.

## Selezione numerica

Il primo argomento dà il numero della selezione desiderata, rispetto a un insieme di relazioni papabili. L'argomento può essere un intero positivo, un intero negativo o la parola `all`. L'argomento `all` seleziona tutti i membri qualificati di un insieme. Un numero seleziona il membro indicato, nell'ordine del documento se l'intero è positivo, nell'ordine inverso del documento se il numero è negativo. Per esempio, supponiamo di voler selezionare tutti gli elementi `messaggio` del Listato 11.1 e di trovarci al momento nel contesto dell'elemento `nota`. In termini relazionali, vogliamo scegliere tutti gli elementi figli relativamente al nodo contesto. Lo possiamo fare con un'espressione come questa:

```
child(all,messaggio)
```

Quest'espressione seleziona tutti i nodi elemento figlio `messaggio` contenuti nell'elemento corrente. Nel Listato 11.1, il risultato sarebbe la selezione di tutti gli elementi `messaggio`, purché l'espressione venga risolta relativamente all'elemento `nota`. Se il contesto fosse in un altro punto del documento, l'espressione non darebbe il risultato desiderato. In effetti, nel caso di `message01.xml` (Listato 11.1), questa espressione fallirebbe in tutti i contesti diversi dall'elemento `nota`. Presumendo che il contesto sia effettivamente l'elemento `nota`, come si dovrebbe selezionare il terzo elemento figlio `messaggio` in un'espressione XPointer? L'espressione corretta sarebbe

```
child(3,messaggio)
```

Il valore numero (3) seleziona un particolare elemento, quando questa espressione viene risolta. In questo caso viene scelto il terzo elemento figlio `messaggio` del contesto `nota`.

## Tipo di nodo

Il secondo argomento dà il tipo del nodo che viene attraversato da XPointer: nella maggior parte dei casi è il nome di un tipo di elemento. Abbiamo visto un esempio di questo secondo argomento nell'ultima sezione:

```
child(all,messaggio)
```

```
child(3,messaggio)
```



## 224   Giorno II

---

In entrambe queste espressioni XPointer, il tipo di nodo indicato è il tipo di elemento `messaggio`. Il primo esempio seleziona tutti gli elementi figli `messaggio`, il secondo solo il terzo figlio nell'insieme dei papabili.

L'argomento tipo di nodo offre una opzione "jolly" per ampliare la selezione al di là di un tipo specificato: è il valore di argomento `#element`. Per trovare tutti gli elementi figli relativamente all'elemento contesto, indipendentemente dai rispettivi tipi di elemento, si può usare l'espressione:

```
child(all,#element)
```

Questa espressione seleziona tutti gli elementi figli, indipendentemente dai loro nomi.

### Filtri degli attributi e dei valori d'attributo

Il terzo e il quarto argomento sono coppie di valori di attributo che si possono usare per filtrare la selezione degli elementi sulla base di loro specifici attributi. Supponiamo di dover interrogare il documento `message01.xml` (Listato 11.1) per localizzare l'elemento `messaggio` che ha l'attributo `da` con valore `Kathy Shepherd`. Si può scrivere quest'espressione in vari modi, ma uno dei più diretti e immediati è:

```
Child(all,#element,da,"Kathy Shepherd")
```

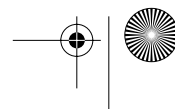
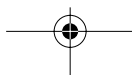
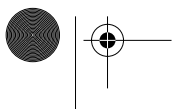
Questa espressione seleziona tutti gli elementi, indipendentemente dai loro nomi di tipo di elemento, purché abbiano un attributo `da` con valore `"Kathy Shepherd"`.

Al posto del nome dell'attributo, nella terza posizione della stringa degli argomenti, si può usare un asterisco (\*). Questo significa che qualsiasi attributo con il valore indicato soddisferà il criterio di selezione.

Per quanto riguarda la quarta posizione nella stringa di argomenti, si può usare o un asterisco (\*), come "jolly" per qualsiasi valore, oppure `#IMPLIED`. `#IMPLIED`, che è preso a prestito dalla logica delle DTD, indica che non è stato fornito alcun valore per l'attributo. Se è dato un argomento `#IMPLIED` e per un certo attributo è stato fornito un valore, quell'istanza non è selezionabile.

### Riepilogo

XPointer è una tecnologia che combina URI ed espressioni XPath per definire in modo particolareggiato la selezione di nodi o punti in un documento. XPointer estende XPath grazie ad alcune caratteristiche specifiche e consente un grado di granularità maggiore: il risultato è una tecnologia non solo potente ma anche selettiva.





## Domande e risposte

**D. Perché non si possono usare i marcatori àncora dell'HTML per accedere a frammenti di una struttura di documenti di sola lettura?**

**R.** In qualche caso è possibile, ma bisogna conoscere i nomi di àncore già esistenti nel documento HTML, all'inizio di frammenti predeterminati. Se si deve accedere a frammenti a cui non è associato un marcatore àncora con nome (o se non se ne conosce il nome), il metodo seguito da HTML non è di alcun aiuto. XPointer permette di selezionare sulla base di criteri che non sono limitati ai nomi ma contemplano anche pattern, numeri d'ordine di elementi e attributi e altri prodotti delle espressioni XPath.

**D. Qual è la relazione fra XPointer e XPath?**

**R.** XPointer è la combinazione di XPath e gli URI. XPath si usa per effettuare selezioni di frammenti di documento, una volta che il documento è stato localizzato sulla base dell'URI risolto.

**D. Perché la sintassi di XPointer permette più parti frammento?**

**R.** Una parte frammento è costituita da uno schema e un'espressione. Al momento l'unico schema è XPointer. Più parti frammento possono essere concatenate in modo che sulle istanze di documento XML si possano condurre test di selezione. Quando l'espressione XPointer viene percorsa, verifica effettivamente la prima parte frammento: se viene risolta, il procedimento si ferma. Se la parte frammento invece non è o non può essere risolta, il procedimento continua con la parte frammento successiva, se esiste. In questo modo il programmatore può selezionare pattern alternativi.

