



Ce que vous devez savoir au sujet du bidi et du balisage au sein des blocs

Ce document est une traduction. En cas de divergences ou d'erreurs, la dernière version originale en anglais fait autorité. Comme indiqué ci-dessous, les droits d'auteur reviennent au W3C.

dans cette page: [définitions](#) - [ordres visuel et logique](#) - [algorithme bidi](#) - [coup de pouce](#) - [les chiffres, un cas à part](#) - [désactiver l'algorithme](#) - [pour approfondir](#)

Ce tutoriel décrit d'abord quelques principes de base de l'algorithme bidirectionnel. Il se penche ensuite sur quelques cas de figure fréquents où l'algorithme bidi a besoin d'information complémentaire qu'il faut fournir à l'aide de balises ou de caractères de commande.

Bien que ce tutoriel essaie d'être neutre en matière de balisage, la plupart des exemples utilisent XHTML car ce langage est sans doute le mieux connu. Pour des conseils sur un langage de balisage particulier, consulter les notes marginales.

définitions

Les **textes bidirectionnels** sont courants dans les écritures droite-à-gauche comme l'arabe, l'hébraïque, la syriaque et la thâna. De nombreuses langues s'écrivent à l'aide de ces écritures.

Toute incise de texte d'une écriture gauche-à-droite ainsi que tous les chiffres s'écrivent de gauche à droite au sein d'un texte écrit globalement de droite à gauche. (Le texte français de ce tutoriel comprend, bien sûr, également du texte bidirectionnel quand il contient des exemples arabes ou hébreux.)

Nous utiliserons le terme **bidi** dans le sens de « bidirectionnel ». De même, nous utiliserons **DàG** et **GàD** comme formes abrégées de « droite-à-gauche » et « gauche-à-droite » respectivement.

ordre visuel et ordre logique

L'**ordre visuel** des caractères s'utilisait couramment pour représenter l'hébreu en HTML dans les anciens navigateurs qui ne mettaient pas en œuvre l'algorithme bidi d'Unicode. Dans une certaine mesure, il persiste encore aujourd'hui, par inertie. En ordre visuel, le premier caractère stocké en mémoire est le caractère qui apparaît à l'extrême gauche de l'écran.

Prenons un texte hébreu à directionnalité mixte.

פעילות הבינום, W3C

Le tableau ci-dessous illustre les représentations en mémoire des caractères de ce même texte codé, d'une part, en ordre logique et, d'autre part, en ordre visuel. L'ordre de stockage représente également l'ordre de saisie ; l'hébreu visuel devait donc être tapé en commençant par la fin logique du texte !

Ordre logique	Ordre visuel
05E4 פ LETTRE HÉBRAÏQUE PÉ	0057 W LETTRE MAJUSCULE LATINE W
05E2 ץ LETTRE HÉBRAÏQUE AÏN	0033 3 CHIFFRE TROIS
05D9 ך LETTRE HÉBRAÏQUE YOUD	0043 C LETTRE MAJUSCULE LATINE C
05DC ם LETTRE HÉBRAÏQUE LAMÈD	0020 ESPACE
05D5 ך LETTRE HÉBRAÏQUE WAW	002C , VIRGULE
05EA ת LETTRE HÉBRAÏQUE TAV	05DD ם LETTRE HÉBRAÏQUE MÉM FINAL
0020 ESPACE	05D5 ך LETTRE HÉBRAÏQUE WAW
05D4 ה LETTRE HÉBRAÏQUE HÈ	05D0 א LETTRE HÉBRAÏQUE ALEF
05D1 ב LETTRE HÉBRAÏQUE BÈT	05E0 נ LETTRE HÉBRAÏQUE NOUN
05D9 ך LETTRE HÉBRAÏQUE YOUD	05D9 ך LETTRE HÉBRAÏQUE YOUD
05E0 נ LETTRE HÉBRAÏQUE NOUN	05D1 ב LETTRE HÉBRAÏQUE BÈT
05D0 א LETTRE HÉBRAÏQUE ALEF	05D4 ה LETTRE HÉBRAÏQUE HÈ
05D5 ך LETTRE HÉBRAÏQUE WAW	0020 ESPACE
05DD ם LETTRE HÉBRAÏQUE MÉM FINAL	05EA ת LETTRE HÉBRAÏQUE TAV
002C , VIRGULE	05D5 ך LETTRE HÉBRAÏQUE WAW
0020 ESPACE	05DC ם LETTRE HÉBRAÏQUE LAMÈD
0057 W LETTRE MAJUSCULE LATINE W	05D9 ך LETTRE HÉBRAÏQUE YOUD
0033 3 CHIFFRE TROIS	05E2 ץ LETTRE HÉBRAÏQUE AÏN
0043 C LETTRE MAJUSCULE LATINE C	05E4 פ LETTRE HÉBRAÏQUE PÉ

L'ordre visuel n'a jamais vraiment été utilisé en arabe. Sans doute la cursivité des lettres arabes qui sont toutes reliées a-t-elle poussé les développeurs arabes à privilégier la méthode de stockage en ordre logique.

L'ordre visuel d'un texte courant oblige l'auteur à désactiver le repli automatique de lignes, à insérer manuellement les passages à la ligne et à aligner à droite le texte situé à l'intérieur des éléments de type bloc et des cellules de tableau. Il faut ensuite coder le texte dans un codage qui évitera l'exécution de l'algorithme bidi d'Unicode dans les navigateurs récents. Voici un exemple écrit en HTML :

```
<table width="50%"><tr><td align="right" nowrap>
,INRIA מ-אירופה באירופה הארחה שירותי את מחליפה את שירותי הארחה באירופה מ-INRIA
W3C<br>
ממקמת בצרפת, ל-ERCIM. השינוי מאפשר ל-W3C
<br>
להעמיק את קשרי המחקר ברחבי אירופה, תוך שמירה
<br>
על הקשר ההיסטורי החזק עם INRIA, אחד ממייסדי
.ERCIM. השינוי יתבצע ב 1 לינואר 2003.
</td></tr></table>
```

(Cet exemple est en fait assez propre. On trouve parfois des choses moins présentables comme des paragraphes alignés à droite dont chaque ligne est entourée de balises `<nobr>..</nobr>`. Si la fenêtre d'affichage est trop étroite, le début de chaque ligne, à droite de l'écran, disparaît.)

L'ordre visuel produit un code très fragile et difficile à entretenir. Ainsi, outre la difficulté de taper de l'hébreu à reculons, si l'on veut ajouter quelques mots à la deuxième ligne de ce paragraphe, il faudra réajuster tous les passages à la ligne subséquents. Il faut également ajouter (et ensuite entretenir) du balisage supplémentaire pour les hyperliens ou les mises en exergue dont le texte s'étend sur plus d'une ligne.

Note: L'ordre visuel peut également poser des problèmes à un niveau supérieur car il force, par exemple, l'inversion manuelle de l'ordre des colonnes d'une table quand on traduit son contenu dans une autre langue. Il faudra également réajuster les passages à la ligne si la géométrie de la page devait changer. Etc.

L'ordre logique est une bien meilleure méthode car le texte est stocké en mémoire dans l'ordre où il est normalement saisi (et prononcé). L'algorithme bidirectionnel d'Unicode réordonne le texte pour l'afficher dans le bon ordre.

La création de longs paragraphes de texte courant qui se replient automatiquement en fonction de la largeur de l'élément bloc qui les contient devient alors aisée. Il est alors nettement plus facile d'utiliser des outils comme les lecteurs d'écran.

L'algorithme bidi opère sur du texte stocké en ordre logique. Si vous préférez utiliser l'ordre visuel, vous pouvez tout de suite arrêter de lire ce tutoriel (bien que sa lecture pourrait vous faciliter grandement la vie).

fonctionnement de l'algorithme bidi

Nous allons maintenant introduire quelques concepts de base importants. Si cela vous semble fastidieux, prenez patience et persévérez car, sans une bonne compréhension de ces concepts, vous serez perdu quand il vous faudra écrire des textes bidi balisés.

Propriété directionnelle des caractères

Nous avons déjà vu qu'une suite de caractères latins est rendue (c.-à-d. affichée) de gauche à droite (comme cette page en est témoin). Par contre, l'algorithme bidi rendra une suite de caractères fortement DàG (arabes par exemple) en les affichant de droite à gauche.

Ceci ne fonctionne que parce qu'Unicode associe à chaque caractère une propriété directionnelle. La plupart des lettres ont un type directionnel GàD fort, c'est le cas des lettres latines, grecques ou cyrilliques par exemple. Les lettres des écritures bidirectionnelles ont, par contre, un type DàG fort.

Quand du texte au sein d'un élément bloc mélange plusieurs directionalités, l'algorithme bidi affiche chaque suite contiguë de caractères de même directionalité sous la forme d'un passage directionnel distinct. L'exemple ci-dessous comprend trois passages directionnels :

bahreïn مصر koweit

L'ordre d'affichage des passages directionnels dépend de leur contexte général. Pour l'exemple ci-dessus, dont le contexte général est GàD puisque il s'inscrit dans un texte en français, on lira d'abord « bahreïn » puis « مصر » (DàG) et enfin « koweit ». Aucun balisage ou stylage n'est nécessaire pour obtenir ce résultat.

Contexte directionnel

Le résultat de l'algorithme bidirectionnel dépend du contexte directionnel global du paragraphe, du bloc ou de la page auquel il s'applique.

En XHTML, si on ne précise aucun attribut `dir` sur la balise `html`, ce contexte directionnel est alors implicitement GàD. En revanche, en ajoutant `dir="rtl"` à la balise `html`, tous les éléments du document héritent d'un contexte DàG. L'utilisation de l'attribut `dir` sur un élément bloc particulier permet de modifier le contexte directionnel en vigueur dans cet élément.

Caractères neutres

Les espaces et les signes de ponctuation n'ont pas de formes GàD ou DàG dans Unicode car ils

peuvent être utilisés avec n'importe quelle écriture. C'est pourquoi on les appelle des **caractères neutres**.

C'est ici que les choses deviennent intéressantes. Quand l'algorithme bidi rencontre des caractères à propriété directionnelle neutre (comme les espaces et la ponctuation), il détermine la suite des choses en fonction des caractères voisins.

Un caractère neutre situé entre deux caractères à forte directionnalité DàG sera traité comme un caractère DàG, ce qui prolonge le passage directionnel. C'est pourquoi les trois mots arabes de la phrase GàD suivante se lisent de droite à gauche (on lit d'abord le mot arabe مفتاح, puis معايير et enfin الويب).

Le titre est مفتاح معايير الويب en arabe.

Remarquez qu'à ce stade tout balisage ou stylage est superflu. L'exemple ci-dessus ne comprend toujours que trois passages directionnels.

Les choses se corsent quand une espace ou un signe de ponctuation se trouve entre deux caractères à directionnalité forte *différente*. Le caractère neutre prend alors la directionnalité *globale du paragraphe ou du contexte*. Si plusieurs caractères neutres se trouvent entre les deux caractères de types forts différents, on associe à tous la même directionnalité résultante.

Ceci crée donc une frontière entre des passages directionnels.

quand l'algorithme a besoin d'un petit coup de pouce

Dans la majorité des cas, l'algorithme bidi traitera parfaitement les textes sans recours à du balisage ou à tout autre mécanisme, en autant qu'on aura précisé la directionnalité globale du document. Mais il est peu probable, toutefois, que vous vous en tiriez *toujours* à si bon compte.

Neutres mal placés

Ajoutons un signe de ponctuation à la fin de la phrase arabe. En absence de balisage, on verra :

Le titre est « !مفتاح معايير الويب » en arabe.

Les guillemets sont bien placés, mais le point d'exclamation est à la mauvaise place. Il devrait apparaître à la fin du texte arabe, c'est-à-dire à sa gauche :

Le titre est « مفتاح معايير الويب! » en arabe.

Grâce à notre compréhension de l'algorithme bidi, il est facile de comprendre pourquoi ceci se produit. En effet, le point d'exclamation se trouvant en mémoire entre la dernière lettre DàG « ب » (à gauche) et la lettre GàD « e » (du mot « en »), c'est le contexte global (ici GàD) qui détermine sa directionnalité. Remarquons qu'il importe peu que l'on trouve à cet endroit deux caractères de ponctuation et deux espaces, car ils sont tous neutres et sont donc affectés de la même façon. Le point d'exclamation étant considéré GàD, il se joint au passage directionnel qui comprend le texte « en arabe ».

Comment faire alors pour remettre la ponctuation à sa place ? Dans ce genre de situation, vous aurez sans doute entouré la citation arabe de balises pour la désigner comme une citation ou pour y adjoindre une information linguistique. Dans ce cas-ci, il existe une solution simple : ajouter un attribut à la balise pour en préciser la directionnalité DàG.

Voilà à quoi cela pourrait ressembler en XHTML :

```
Le titre est «&nbsp;<span dir="rtl" lang="ar" xml:lang="ar">مفتاح معايير الويب!</span>&nbsp;>»  
en arabe.
```

Observez bien que la balise span se trouve à l'intérieur des guillemets car ces derniers font partie du texte français.

Une autre solution serait de faire suivre le point d'exclamation d'un caractère invisible à forte directionnalité DàG. De la sorte, le point d'exclamation serait interprété comme DàG et ferait dès lors partie du passage directionnel arabe.

Or, il se trouve qu'Unicode contient un tel caractère — le caractère U+200F, appelé MARQUE DROITE-À-GAUCHE (MDAG). Sa contrepartie GàD existe également, il s'agit de U+200E MARQUE GAUCHE-À-DROITE (MGAD).

Ce caractère étant invisible, il vaut mieux le saisir sous la forme d'un appel de caractère numérique (‏) ou — si possible — d'un appel d'entité (par exemple ‏ en XHTML). Dans l'exemple suivant, on a ajouté ‏ après le point d'exclamation, le résultat est alors correct :

Le titre est « مفتاح معايير الويب! » en arabe.

Neutres en pleine crise d'identité

Dans notre prochain exemple, l'ordre de la liste est incorrect car les deux premiers mots arabes devraient être intervertis et la virgule qui se trouve entre eux et qui fait partie du texte français devrait apparaître à la droite immédiate du premier mot.

Les noms de ces États en arabe sont respectivement le الكويت, مصر, البحرين.

Le résultat escompté était :

Les noms de ces États en arabe sont respectivement الكويت, البحرين, مصر.

Ce résultat inattendu s'explique par la présence de chaque côté de la virgule de caractères à forte directionnalité droite-à-gauche (DàG). En présence de ces caractères, l'algorithme bidirectionnel considère la virgule neutre comme faisant partie du texte arabe alors qu'elle fait partie du texte français et devrait marquer la frontière entre deux passages directionnels arabes.

Alors que, dans la section précédente, le caractère neutre pensait faire partie du contexte global et qu'il n'en faisait pas partie, dans ce cas-ci le neutre pense faire partie du passage contra-directionnel alors qu'il fait partie du contexte global ! Personne n'a dit que la vie était simple...

Une solution élégante consiste à insérer à côté de la virgule un autre caractère Unicode invisible, cette fois-ci il s'agit de la MARQUE GAUCHE-À-DROITE. Ceci place notre virgule entre deux caractères à directionnalité forte, l'un DàG et l'autre GàD, ce qui la force à prendre la directionnalité du contexte global, c'est-à-dire le flot GàD français. Ceci sépare les mots arabes pour former deux passages distincts qui sont affichés en ordre GàD conformément à la direction dominante du paragraphe.

À nouveau, il se peut que vous préféreriez utiliser un ACN (‎) ou un appel d'entité (comme &llm;) si possible.

Entourer la virgule de balises dans ce cas-ci reviendrait à écraser une mouche avec un marteau.

Encore une fois, tous ensemble !

Les exemples que nous avons montrés jusqu'à présent étaient en français et avaient donc une directionnalité globale GàD. Les mêmes principes s'appliquent aux textes DàG écrits dans des langues comme l'hébreu ou l'arabe. Un exemple supplémentaire nous aidera à mieux comprendre.

Voici ce qu'on voudrait voir, dans un paragraphe dont la direction de base est réglée à DàG.

.ERCIM - מעביר את שירותי הארחה באירופה ל - W3C (World Wide Web Consortium)

Malheureusement, sans aide l'algorithme bidirectionnel crée un véritable fouillis :

.ERCIM - W3C (World Wide Web Consortium) מעביר את שירותי הארחה באירופה ל -

On peut croire qu'il sera extrêmement compliqué de régler ce gâchis, mais en fait la solution est des plus simples. Il suffit d'insérer une MDAG après « W3C » et le tour est joué. C'est vraiment aussi simple !

Si vous n'êtes pas convaincu, voici l'explication. La MDAG après « W3C » sépare ce morceau de texte GàD du passage directionnel entre parenthèses qui le suit. Souvenez-vous que les passages directionnels s'afficheront ici de droite à gauche, qui est la direction générale du paragraphe. Le « W3C » ayant été saisi en premier lieu, il apparaît le plus à droite. La parenthèse se trouve maintenant entre des caractères de forte directionnalité différente et prend donc la directionnalité globale du paragraphe. Elle suit donc. Puis vient le passage directionnel GàD continu, c'est-à-dire tout ce qui se trouvait à l'intérieur des parenthèses.

Note: Afin de mieux comprendre la suite de caractères des exemples ci-dessus, il vaudrait peut-être la peine de déplacer le curseur à travers le source de ce texte dans un éditeur plutôt que de simplement examiner l'écran.

(Le changement de glyphe de la parenthèse située le plus à droite est automatique. Le glyphe utilisé pour ces « caractères miroités » se transforme automatiquement en fonction de leur directionnalité. Il s'agit cependant du même caractère.)

Emboîtement des passages directionnels

L'algorithme bidi d'Unicode et les marques directionnelles font l'affaire quand on n'est en présence que d'un seul niveau de texte à directionnalité contraire (appelé parfois **palier directionnel**). En présence de deux ou plus de deux niveaux imbriqués (paliers) de texte directionnel, une autre solution s'impose. Mais d'abord un exemple de texte mal ordonné :

Le titre est « פעילות הבינאום, W3C » en hébreu.

L'ordre des deux mots hébreux est correct, mais le terme «W3C» devrait apparaître du côté gauche de la citation et la virgule devrait se trouver entre le texte hébreu et « W3C ». En d'autres mots, le résultat escompté est :

Le titre est « W3C, פעילות הבינאום » en hébreu.

Ce problème se présente parce que le contexte GàD du paragraphe détermine l'ordre des passages directionnels. À l'intérieur de la citation en hébreu, toutefois, l'ordre implicite devrait être DàG.

Pour résoudre ce problème, il faut déclarer un nouveau palier directionnel. Pour ce faire, en XHTML, on encadre la citation à l'aide de balises et on lui affecte une directionnalité DàG grâce à l'attribut `dir`.

Le titre est

```
<&nbsp;<span dir="rtl">W3C , פעילות הבינאום</span>&nbsp;>
```

en hébreu.

Pour des langages de balisages autres que XHTML/HTML, il se peut qu'un attribut similaire existe auquel vous pourrez associer une mise en forme particulière pour obtenir le résultat souhaité. Si un tel attribut n'existe pas, il faudra probablement vous résoudre à appliquer manuellement du stylage à certains éléments, mais il vaudrait sans doute mieux que fassiez pression auprès du concepteur de votre langage de balisage pour qu'il vous fournisse un tel attribut.

Il existe des caractères de commande Unicode qui fourniraient le même résultat, mais ils créent des régions aux frontières invisibles et **ne sont donc pas recommandés**.

les chiffres, un cas à part

Les chiffres dans les écritures DàG s'écrivent de gauche-à-droite à l'intérieur de passages droite-à-gauche, mais l'algorithme bidi les traite quelque peu différemment des mots. On dit qu'ils ont une directionnalité faible. Les deux exemples ci-dessous illustrent cette distinction. Si l'on compare les deux lignes, on remarque que les mots arabes qui entourent le quatrième élément ont été intervertis. La seule différence en mémoire entre ces deux lignes est l'emploi de « 1234 » à la place de « quatre ».

un deux ثلاثة quatre خمسة

un deux خمسة 1234 ثلاثة

Dans le premier exemple, les lettres de « quatre » ont un type fort et elles séparent les mots arabes pour former deux passages directionnels distincts affichés de gauche à droite conformément au contexte du paragraphe.

Dans le second exemple, le nombre à type faible « 1234 » est considéré comme faisant partie du texte arabe, il ne rompt pas le passage directionnel arabe : les deux mots arabes et « 1234 » forment un même passage, même si les chiffres s'affichent de gauche à droite à l'écran.

Note: Dans certaines versions des navigateurs de Netscape et Mozilla, le quatrième élément de la deuxième rangée apparaîtra sous la forme de ١٢٣٤ plutôt que 1234. Les deux chaînes représentent le même nombre. Notez que Mozilla 1.4 considère à nouveau la forme européenne des chiffres comme l'œil implicite à utiliser.

Ceci ne se produit que dans des textes DàG.

Complicé ? Ne vous en faites pas, d'habitude l'algorithme bidi s'occupera de tout pour vous. Je n'ai inclus cette section que pour ceux qui remarqueraient la différence et se demanderait it ce qui se passe.

Enfin, notez que d'autres caractères neutres accolés à un nombre, comme les symboles de devises monétaires, sont considérés comme faisant partie du nombre plutôt que comme des neutres.

désactiver l'algorithme

Il peut arriver que vous vouliez empêcher l'algorithme bidi d'effectuer son travail de réordonnement sur un passage de texte. Il faut alors entourer ce passage de balisage supplémentaire.

En XHTML 1.0, on utilise pour ce faire l'élément `bdo`. En XHTML 2, l'élément `bdo` sera probablement remplacé par les valeurs `rlo` et `lro` ajoutées à l'attribut `dir`, ce qui permettra de l'appliquer à n'importe quel élément. À nouveau, il existe des caractères de commande Unicode qui permettent d'obtenir le même résultat, mais puisqu'ils créent des régions aux frontières invisibles, ils **ne sont pas recommandés**.

Les exemples de cet article qui montrent l'ordre des caractères en mémoire utilisent la balise `bdo` pour obtenir cet effet. Ainsi, pour illustrer la suite de caractères correspondant à :

פעילות הבינאום, W3C

il suffit d'utiliser le balisage suivant en XHTML 1.0 :

```
<p><bdo dir="ltr">פעילות הבינאום, W3C</bdo></p>
```

Le résultat, écrit de gauche à droite, est :

פעילות הבינאום, W3C

pour approfondir

- [Unicode et ISO 10646 en français](http://hapax.qc.ca) *http://hapax.qc.ca*
- [Bidi techniques in HTML \(en dev[†]\)](http://www.w3.org/International/geo/html-tech/tech-bidi.html) *http://www.w3.org/International/geo/html-tech/tech-bidi.html*
- [The Unicode Bidirectional Algorithm](http://www.unicode.org/reports/tr9/) *http://www.unicode.org/reports/tr9/*
- [Non-Latin scripts tutorial](http://people.w3.org/rishida/scripts/tutorial/) *http://people.w3.org/rishida/scripts/tutorial/*
- [Other W3C I18N resources relating to Bidirectional text](http://www.w3.org/International/resource-index.html#bidi) *http://www.w3.org/International/resource-index.html#bidi*

remerciements

Mes plus vifs remerciements à Martin Dürst pour ses nombreux commentaires judicieux au sujet d'une ébauche de ce document.

Auteur : Richard Ishida, W3C. Traducteur : P. Andries, [Hapax](#).

Traduit de la version anglaise datée du 30 janvier 2004. La traduction a été modifiée pour la dernière fois le 2005-02-18 14:50 TU.

Copyright © 2004 W3C[®] (MIT, ERCIM, Keio), All Rights Reserved. W3C [liability](#), [trademark](#), [document use](#) and [software licensing](#) rules apply. Your interactions with this site are in accordance with our [public](#) and [Member](#) privacy statements.

