

01:HTML(基本編)

Webページを作るために必要なもの

- Webブラウザ・・・ユーザーがWebページを見たり、サーバーにhttpアクセスする際に使う
- Webサーバー・・・Webページのデータを置く場所、ユーザーはサーバーのサイトにアクセスします。
- ドメイン・・・ブラウザでサイトアクセスする際に必要なサイトのurlです。
- Webサイトのデータ・・・HTML、CSS、Javascriptやphpなど

webブラウザの種類

GoogleCrome、Firefox、InternetExprole、edge、Safariなど

世界や国内のユーザーが使っているブラウザのシェアなども定期的にチェックしましょう。

https://webrage.jp/techblog/pc_browser_share/

WEBサーバー

- レンタルサーバー：ロリポップ、さくらサーバー、ヘテムル..など
- クラウドサーバー：AWSやGCP(Google Cloud Platform)..など

どのサーバーのどのプランを使うか、費用感やメリット・デメリットなど概要を調べておきましょう。

ドメイン

ドメインとは「インターネット上の住所」のことで、Webサイトがどこにあるかを判別する情報として利用します。

詳しく説明すると、Webサイトがどこにあるのかを判別する情報は、ドメインに紐づく「IPアドレス」というものにあります。

IPアドレスは「000.00.00.00」のような数字で構成される文字列ですが、数字だけでは覚えづらいため、ドメイン名に変換してWebサイトの場所を示します。

住所

緯度:35.656456 経度:139.699418



東京都渋谷区桜丘町26-1

ドメイン

IP:133.130.34.12

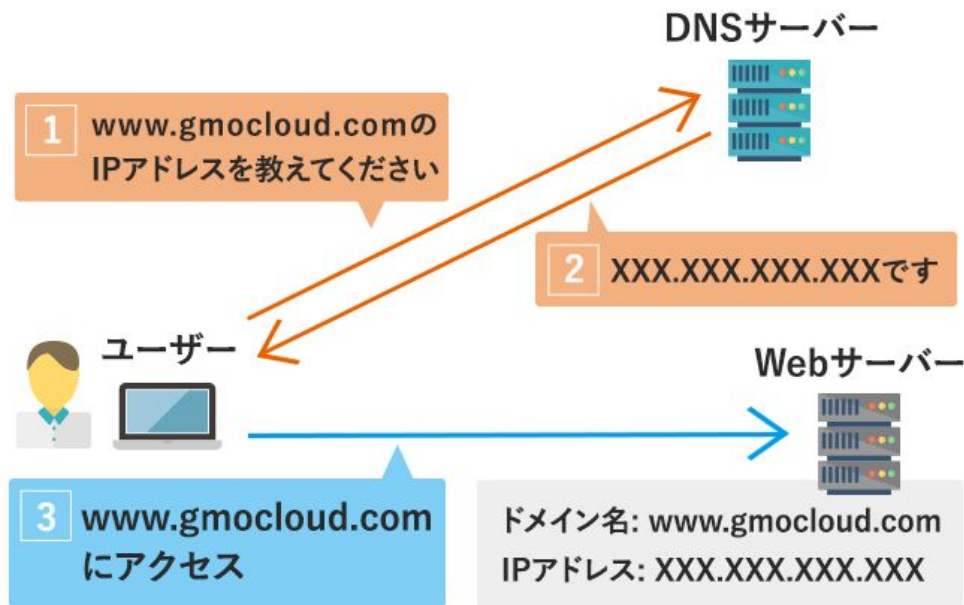


muumuu-domain.com

サイト専用のアドレス名を使ってサイトにアクセスしてもらうため、独自ドメインを取得しましょう。

ドメインを取り扱う主なサービス（レジストラ）：

- お名前ドットコム
- ムームードメイン
- さくらインターネット



Webページを構成する必要なデータファイル

デザインしたデータをwebサイトとして公開するためにデータとして組み立てる必要があります。
以下の言語を使って、ファイルを組み合わせるwebサイトやアプリケーションを作ることができます。

- HTML・・・ページを表示するためのファイル。文書構造やサイトの説明、必要情報を記述する
- CSS・・・・・・ページを装飾、レイアウトするためのファイル。色や模様、レイアウトの指示をする
- Javascript・・・ページにインタラクティブな動きを設定するファイル。

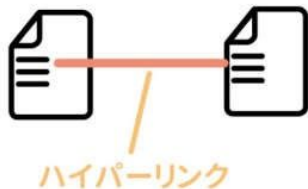
ハイパーテキスト（=ウェブ）と WWW（=ワールド・ワイド・ウェブ）

ハイパーテキストとは複数の文書を相互に結びつける仕組みのことです。
これが大規模になったものが、ワールド・ワイド・ウェブです。

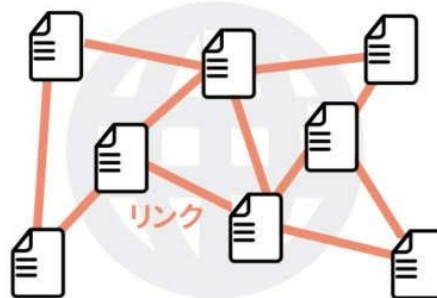
大規模なハイパーテキスト=ウェブ と呼び、
World Wide（世界的な）Web（クモの巣）なので、
世界的に広がったドキュメント同士の繋がりのことをwwwと言います。

ハイパーテキスト

複数の文書を相互に
結びつける仕組み



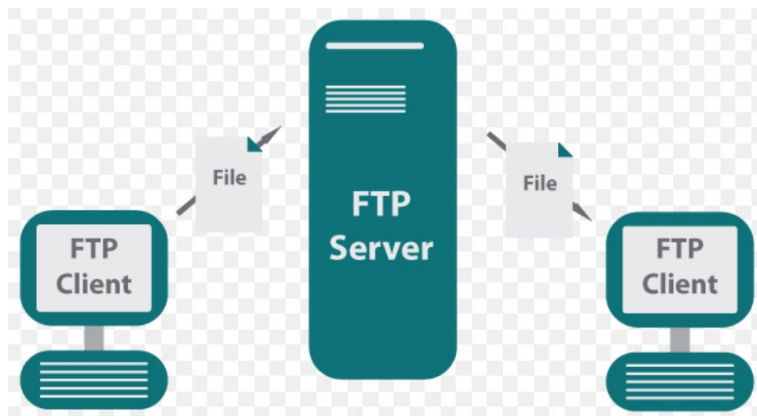
ウェブ = 大規模な
ハイパーテキスト



Webサイトの公開に必要な通信の仕組み

用途に合わせて通信の種類が切り替わります。通信のルールや規約のことをプロトコルと呼びます。
Webサイトに関わる通信の代表的なものです。

- http・・・ブラウザでドメインを入力しアクセスする際、ページが表示される際に使われる通信
- ftp・・・webサイトのデータをサーバーにアップ、ダウンロードする際に必要な通信



ftpクライアントソフトについて

サーバーとftp通信を行いデータのアップやダウンロードする際に必要なアプリケーションソフトのことです。
FileZila、cyberduck、Transmit、FFFTP、WinSCPなど

HTMLの歴史

参考文献：<https://matome.naver.jp/odai/2136874479784697001>

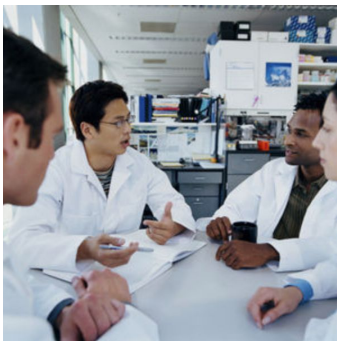
###HTMLが生まれた場所



スイス・ジュネーブ

スイス・ジュネーブにあるCERN(欧州原子核研究機構)で誕生する。
ここは、欧州12か国が共同で研究を行う機関です。
1989年にここでHTMLは開発されました。

HTMLが生まれた理由、その経緯



情報を効率よく探す必要があった

CERNは、何千人という科学者が入れ替わり立ち代わりで訪れるため、お互いの研究状況の確認、論文などの資料を探す、など必要な情報を効率よく行き渡らせる環境が課題となっていました。
これを実現するために、CERNで世界初のサーバーとブラウザが生まれます。

サーバーに情報を蓄積させ、必要な時に必要な情報をブラウザで表示出来るようになったので、効率よく情報を収集する環境が整いました。

また、情報と情報をリンクさせることで、とてもスマートに情報を整理することができるようになったのです。

HTMLの考案者



ティム・バーナーズ・リー (イギリスの計算機科学者。)

『ロバート・カイリューとともに、World Wide Web (WWW) を考案し、ハイパーテキストシステムを実装・開発した人物である。URL、HTTP、HTML の最初の設計は彼によるもの』 - Wikipediaより
現在ではCSSによる装飾がセットとして存在しているHTMLですが、そもそも情報管理・整理が目的なため、当時は文書構造さえ記述できれば要件は満たされていました。

##HTMLの歴史の流れ

1989年 HTML開発

ティム・バーナーズ・リーによって開発される。

1990年 サーバーとブラウザが誕生

世界初のサーバーとブラウザが誕生。

HTTPという通信手段を使って、URLでファイルが保管されているサーバーの特定の場所へアクセスし、ブラウザでHTMLを人間の目でキレイに成形する仕組みです。

1993年 Mosaicブラウザが登場

アメリカのイリノイ大学のとある大学生によって世界初の画像表示ブラウザ「Mosaic1.0」が発明されます。

この大学生の名はマーク・アンドリーセン(後のNetscape開発者)。このブラウザの発明を契機に大きなブラウザ戦争の歴史が幕を上げるのですが、これについては改めてまとめます。

1994年 W3Cの設立

ティム・バーナーズ・リーが、W3Cという非営利団体を設立します。

W3Cとは、簡単にいえば「ウェブをみんなで使えるように、使う人や場所や環境に差異がないようにしようね。そういう仕様や技術書を作成したり勧告するよ」という機関です。

1996年 CSS1.0を勧告

今まで、HTMLでも簡単な文字の装飾などが出来ていましたが、W3Cが「HTMLって本当は文書構造を記述するためのものだから、装飾したい場合はCSSを使おうよ」と勧告します。

1997年 HTML4.0を勧告

CSS1.0に続き、HTML4.0の仕様をW3Cが勧告。ここから「HTMLは文書構造を記述するもの、CSSは体裁を整えるためのもの」という切り分けが明確になっていきます。

1998年 XML1.0を勧告

HTMLの発展型としてXMLという規格がW3Cから勧告されます。これは、「この数値は商品価格、この数値は製造年月日ね」という具合に、任意の箇所を要素として定義してデータをやりとりするためのもの。

皆さんご存知のitunesの曲リストとかも、このXMLを使って「ここは歌手名、ここは曲名のデータだよ」というやりとりがされてます。

2000年 XHTML1.0を勧告

「XMLはデータのやりとりに最適。だから段階を経てXMLへ移行させよう。ひとまず、今までのHTMLの仕様を、このXMLに準拠したものに新しく書き直したよ」ということで、HTML4.0を基にしたXHTML1.0が勧告されました。

2004年 CSS2.1を勧告

W3Cが新しいCSSの仕様を勧告。同年、世界にはブログが広く普及しXHTML+CSSが流行。また、ウェブアプリが勢力を拡大していました。

2004年 W3Cワークショップにて

ブログの普及により、XHTMLは浸透しつつもその用途はXMLよりもHTMLとして利用されることがほぼ。Mozilla(Firefox)やApple(Safari)などのブラウザベンダーは、世間に浸透していないその流れを考慮し、W3Cに提案します。

ベンダー「HTMLを発展させません？」
W3C「うちはXML普及が優先事項です。」
ベンダー「でも現場のニーズも考えてよ。」
W3C「うちでは出来なすすって。」
ベンダー「まじか〜っ！」

ということで、Mozilla・Apple・Operaに所属するメンバーによって「WHATWG」が設立。現状のHTML機能を拡張させる取り組みが独自で進みます。

2006年 XML移行プランの失敗

2004年設立から2年。WHATWGの仕様は評価されていき、ティム・バーナーズ・リーが「世界をXMLへとスイッチさせる試みはうまく機能しなかった。」と自身のブログで発言。翌2007年からは、W3CとWHATWGが協力して、HTML5の標準化を進めます。

2008年 HTML5草案を発表

ようやく漕ぎ着けた新HTML。従来のHTMLとの違いは、その拡張性と互換性。元々、ブラウザベンダーやそれに賛同した人たちが主体で進行していたプロジェクトだったため、現場の声がよく反映されている。文書構造という概念はもちろんありますが、次世代というか、現代的な香りのする仕様になっています。

##HTMLの考え方・構造

###HTML = マークアップ言語

HTML（HyperText Markup Language / ハイパーテキスト マークアップ ランゲージ）は、Webサイト上の文書を作成するためのマークアップ言語です。

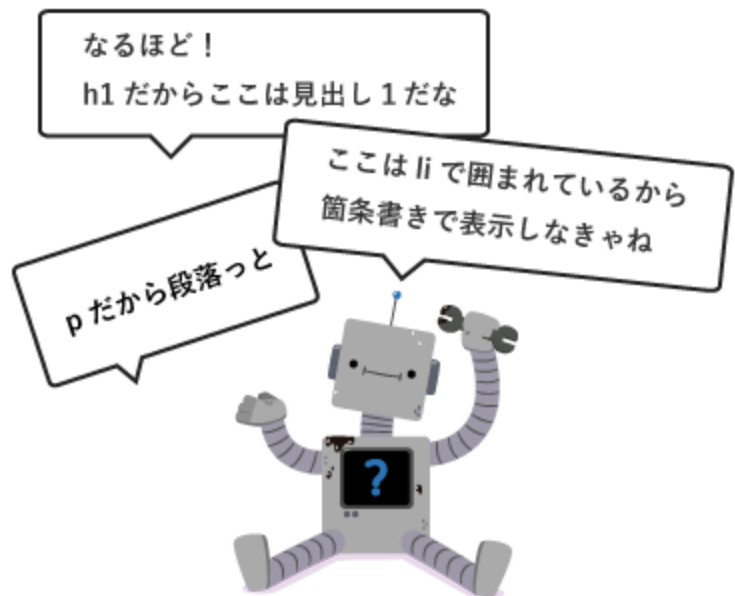
###マークアップ言語とは

タグで囲むことで**意味**をつけていく それを「**マークアップ**」と呼ぶ

```
<h1> これは大見出し </h1>

<ul>
  <li> 箇条書き 01 </li>
  <li> 箇条書き 02 </li>
  <li> 箇条書き 03 </li>
</ul>

<p> ここはテキストです </p>
<p> ここはテキストです </p>
```



Webサイト上の文書作成とは何か？というと、「テキスト情報の記述」「画像（動画・音声）情報の配置」「リンクの設置」などがそれに当たります。

またそれぞれの情報に「見出し」や「段落」などといった、役割を与えるためのタグ付け（マークアップ）を行います。

これによってブラウザや検索エンジンに対し、どの情報がどのような意味や役割を持つかを伝えることができ、コンピューターにとって読みやすい文書を作ることができます。

→検索エンジンに良い評価してもらうことで、検索にかかりやすいサイトになります。

検索に影響があることですので、正しい書式で良いマークアップができるよう心がけましょう。

HTMLで行わないこと

逆にそれ以外のこと（色や大きさ、レイアウトの指定 / マウス操作に合わせた動き等）はHTMLという言語では記述しないという考え方が一般的で、それ以外のことを記述する場合は他の言語（CSSやJavascript）を使用します。

そのためHTMLのみではデザイン性や機能性において優れたサイトを作ることはできません。優れたWebサイトを制作するためにはその他の言語を使用することが必須となります。



様々なWeb言語

様々なWeb言語を使って一つのサイトを作成するというのが、現在のWeb制作における一般的な考え方となっています。

Webサイト制作で使用される代表的な言語の種類と役割は以下のようになります。

- HTML（文書を作成）
- CSS（装飾の指定）
- Javascript（動的な仕組みの作成）

HTMLによるマークアップ / マークアップ言語とは

前の記事で書いた通り、HTMLはマークアップ言語と呼ばれる言語です。

簡単に要約してしまうと「文章中の語句に役割を与えること」をマークアップと考え、「マークアップのために使用する言語をマークアップ言語」と考えると分かりやすいでしょう。

人間がどのように情報を読むか

本や雑誌などを見てみると必ずいくつかのレベルの「見出し」が存在します。

またその見出しに対しての本文（段落）が存在します。

私たち人間はまずいくつかの「見出し」を比較して興味がありそうな場所を探し、興味のある見出しに対する本文(段落)を読みます。

そのようにすることで自分にとって必要のない場所を読み飛ばして、最小限の時間で必要な情報のみを読むことができます。

コンピューター（ブラウザ・検索エンジン）にも人間と同様に情報を理解させる

Webにおける文章にも「見出し」や「本文（段落）」などの役割が存在します。

人間であれば文字の大小や文脈などからそれを判断できますが、コンピューターは理解することができません。

HTMLのタグを用いることで文章中の語句に役割を与えることができ、役割を与えられたページの情報はブラウザや検索エンジンにとって、読みやすく理解しやすいページとなります。

マークアップ言語にはHTMLの他にSGML、XML、XHTML等の言語が存在します。

HTMLの基本文法

どのようにして役割を与えるのか？

下記にマークアップされていない文章と、マークアップされた文章の例を記述します。

【マークされていない文章】

Webサイトのタイトル

ページの中見出し

説明文など段落文章

【マークされている文章】

`<h1>Webサイトのタイトル</h1>`

`<h2>ページの中見出し</h2>`

`<p>説明文など段落文章</p>`

タグ（記号）で囲む

情報の先頭に付けるタグを「開始タグ」、後方に付けるタグを「終了タグ」と呼びます。

HTMLの基本文法はいたってシンプルで、情報を開始タグと終了タグで囲むということになります。

またタグの種類によっては終了タグが存在せず、タグが一つで完結するものもあります。（改行や画像挿入のタグなど）

HTMLの基本構造

```
1  <!DOCTYPE html>
2  <html lang="ja">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Document</title>
8  </head>
9  <body>
10
11 </body>
12 </html>
```

どのブラウザでも正しい表示、検索エンジンにも正しく文書構造を読み取ってもらうために、htmlの基本構造に沿って作成する必要があります。

大枠として必要な要素は以下の内容です。

- ドキュメント宣言・・・このファイルはhtmlで書かれていますという証明
- html要素・・・ファイルの内容は全てこの囲みの中に書きます。
- -head要素・・・ページのタイトルや説明、設定などの付加情報を記載する囲みです。
- -body要素・・・ブラウザに表示される内容を記載する囲みです。

様々な役割のタグ

HTMLのタグは様々な種類が用意されており、それぞれのタグにはそれぞれの役割があります。適切な役割のタグでマークアップすることが、正しいHTML文書の作成に繋がります。以下がWebサイト制作で頻繁に使用される代表的なタグとなります。

- 見出し
- 段落
- リスト
- アンカー（リンク）
- 画像挿入
- 改行
- 範囲指定（構造を表すタグ）

hタグ＝見出し

<h1>,<h2>,<h3>,<h4>,<h5>,<h6>

見出しを定義するタグです。

見出しは階層構造のように構成し6段階のレベルに分けられます。

h1が最も大きな見出しで、h2、h3・・・h6と徐々に見出しのレベルが低くなっていきます。

H1に関しては「ページ内の最も大きな見出し」ということなので、1ページ内に一度しか記述できません。

見出しはh1,h2,h3と番号順に出現しないといけません。

注意：h1はページで1つです。（例外もあります）

注意：h1の次がh3という記述はNGです。

注意：h2の次がh1という記述はNGです。

pタグ=段落

<p>

段落を定義するタグです。文章を一塊として扱います。

※見出し以外のテキストは基本的にpタグでくるものだと考えるとよいでしょう。

```
<body>
  <h1>大見出し</h1>

  <h2>項目の中見出し</h2>
  <p>段落テキスト</p>

  <h2>項目の中見出し</h2>
  <p>段落テキスト</p>
  <h3>小見出し</h3>
  <p>段落テキスト</p>
  <h3>小見出し</h3>
  <p>段落テキスト</p>

  <h2>項目の中見出し</h2>
  <p>段落テキスト</p>
</body>
```

brタグ=改行

Hタグやpタグの中で、テキストの改行が必要な場合に置きます。

※brタグを置かないとソースコード上で改行したつもりでも、ブラウザの表示では改行されません。

```
<p>文章の改行、文末の句読点「。」の後に改行タグを置きましょう。<br>  
文章の改行、文末の句読点「。」の後に改行タグを置きましょう。<br>  
文章の改行、文末の句読点「。」の後に改行タグを置きましょう。</p>
```

ulタグ liタグ=リスト

順序の無いリストをまとめるタグです。

ulタグで囲まれたリストには頭に中黒が付きます。

```
<ul>  
  <li>箇条書き項目のリスト</li>  
  <li>箇条書き項目のリスト</li>  
  <li>箇条書き項目のリスト</li>  
</ul>
```

用途としては、箇条書きの項目やナビゲーションなどに使われます。

olタグ liタグ=順序付きリスト

順序のあるリストをまとめるタグです。

olタグで囲まれたリストには頭に通し番号が付きます。

```
<ol>  
  <li>順序指定付き項目のリスト</li>  
  <li>順序指定付き項目のリスト</li>  
  <li>順序指定付き項目のリスト</li>  
</ol>
```

用途としては、手順やフローなど順番を守って読んでもらいたい箇条書きの項目に使われます。

dlタグ dtタグ ddタグ =定義リスト

<dl><dt><dd>

用語の解説を作成するタグです。下記のように記述して定義リストを作成します。

用途としては商品と商品説明、人物と人物説明といったような組み合わせで使われるものだと考えましょう。

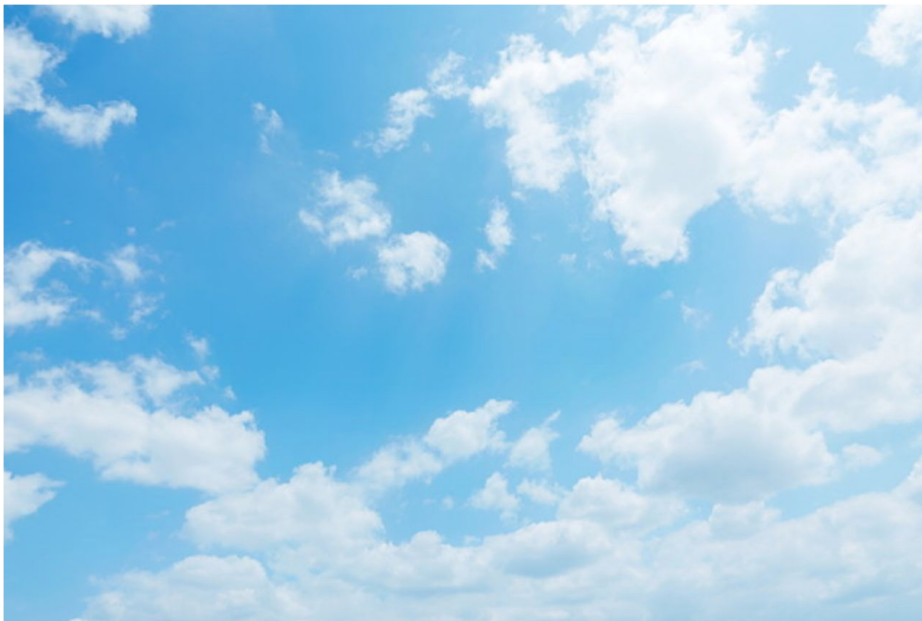
```
<dl>
  <dt>定義する用語(タイトル)</dt>
  <dd>用語の解説</dd>
  <dt>定義する用語(タイトル)</dt>
  <dd>用語の解説</dd>
</dl>
```

imgタグ = 画像

画像を配置するためのタグです。テキスト情報などと同様に扱い、画像の説明なども記載しましょう。

src=""属性は画像のパスを記述します。 alt=""に画像の説明を記述します。

```
<p></p>
```



画像設置例：

```
<p></p>
```

figureタグ、figcaption = 図表、図表の説明

<figure><figcaption>

画像などの図表をマークアップするタグです。

Figureの中にimgタグとfigcaptionを置くことで、画像とそのキャプションという関係性で1組の情報としてマークアップすることができます。

```
<figure>
  
  <figcaption>画像のキャプション</figcaption>
</figure>
```

timeタグ = 時間を表す

<time>

お知らせ一覧など投稿時間の表記（タイムスタンプ）などに対してのマークアップに使用します。

こちらのタグはセットで、**datetime=""**属性が必要です。

例の時間の表記を参考にしてください。

例：2018年7月7日20時だった場合

※時間の前の「T」は無し＝半角スペースでも大丈夫です。

```
13      <time datetime="2018-07-07T20:00:00">20:00</time>
```