

The screenshot shows the CSS Zen Garden homepage. A red box highlights the main content area, which includes sections like "The Beauty of CSS Design" and "The Road to Enlightenment". Another red box highlights the browser's developer tools, specifically the element inspector, which is displaying the underlying HTML code. The code is annotated with pink arrows pointing to specific elements: "section", "header", and "p". To the right of the code, handwritten notes explain that this is "semantic layout markup" used to "organiser le contenu".

```

<div class="page-wrapper">
  <section class="intro" id="zen-intro">
    <header role="banner">
      <h1>CSS Zen Garden</h1>
      <h2>The Beauty of <abbr title="Cascading Style Sheets">CSS</abbr> Design</h2>
    </header>
    <div class="summary" id="zen-summary" role="article">
      <p>A demonstration of what can be accomplished through <abbr title="Cascading Style Sheets">CSS</abbr></p>
      <p>Download the example <a href="/examples/index" title="This page's source"></a></p>
    </div>
    <div class="preamble" id="zen-preamble" role="article">
      <h3>The Road to Enlightenment</h3>
      <p>Littering a dark and dreary road lay the past relics of browser-specific <br> CSS</p>
      <p>We must clear the mind of the past. Web enlightenment has been achieved!<br> CSS</p>
      <p>The CSS Zen Garden invites you to relax and meditate on the important les<br> tions of CSS</p>
    </div>
  </section>
  <div class="main-supporting" id="zen-supporting" role="main">
    <div class="explanation" id="zen-explanation" role="article">
      <h3>So What is This About?</h3>
      <p>There is a continuing need to show the power of <abbr title="Cascading Style Sheets">CSS</abbr> allows complete and total control over the presentation layer</p>
    </div>
  </div>

```

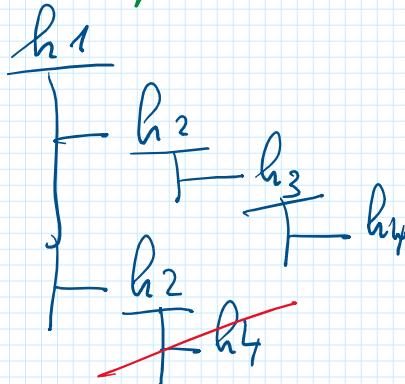
Semantic content markup

h1, h2, ..., p, img, a, table, ul, li, ...

⇒ utile pour les moteurs de recherche

= comprendre le contenu

+ hiérarchie



W3C validation ⇒ well-formed document

Semantic content markup

<time>

Baseline Widely available



L'élément HTML <time> permet de représenter une période donnée. Cet élément permet d'utiliser l'attribut `datetime` afin de traduire la date ou l'instant dans un format informatique (permettant aux moteurs de recherche d'exploiter ces données ou de créer des rappels).

Exemple interactif

HTML Demo: <time>

HTML **CSS** **OUTPUT** **Reset**

```

1 <p>
2   The Cure will be celebrating their 40th
anniversary on
3   <time datetime="2018-07-07">July 7</time>
in London's Hyde Park.
4 </p>
5
6 <p>
7   The concert starts at <time
datetime="20:00">20:00</time> and you'll be
able

```

The Cure will be celebrating their 40th anniversary on July 7 in London's Hyde Park.

The concert starts at 20:00 and you'll be able to enjoy the band for at least 2h 30m.

format date =>

visible par une machine

Important SEO

attribut `datetime`

Valeurs valides

FR

Une chaîne de caractères représentant une année

2011 ou 0001

Une chaîne de caractères représentant une année et un mois

2011-11

Une chaîne de caractères représentant une date

2011-11-18

Une chaîne de caractères représentant une date sans l'année

11-18

Une chaîne de caractères représentant une semaine

2011-W47

Valid datetime values

EN

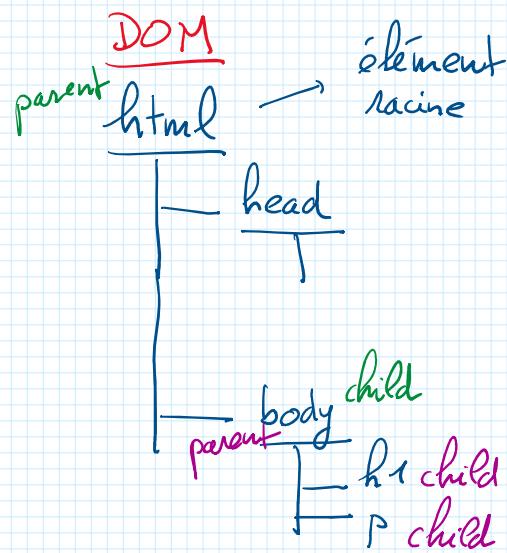
| Description | Microsyntax | Examples |
|----------------------------|--|--|
| Valid month string | YYYY-MM | 2011-11, 2013-05 |
| Valid date string | YYYY-MM-DD | 1887-12-01 |
| Valid yearless date string | MM-DD | 11-12 |
| Valid time string | HH:MM HH:MM:SS HH:MM:SS.mmmm | 23:59 12:15:47 12:15:52.998 |
| | YYYY-MM-DD HH:MM YYYY-MM-DD HH:MM:SS YYYY-MM-DD HH:MM:SS.mmmm | 2013-12-25 11:12 1972-07-25 13:43:07 1972-07-25 13:43:07.123 |

Modélisation x HTML struct

```

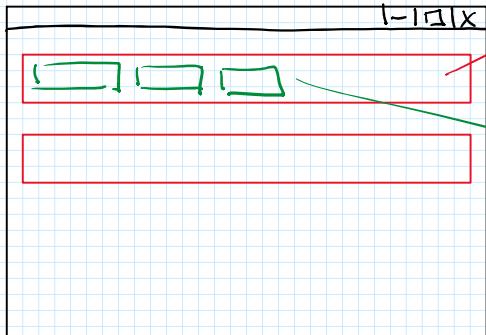
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="scale=1.0">
6      <title>Document</title>
7    </head>
8    <body>
9      <h1>Titre</h1>
10     <p>Texte</p>
11   </body>
12 </html>

```



1) Permitted Parent/Child Relationship

2) Behaviour : block
inline



block : h1, h2, p, ...

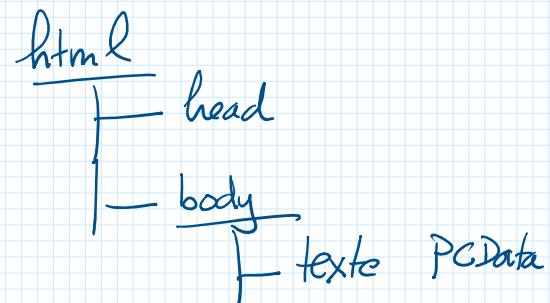
(CR, nouvelle ligne)

inline : b, a, i, ...

CR/LF

Case Study N°1 : text as a child of body
xHTML, then HTML5 validation

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6 </head>
7 <body>
8 Lorem ipsum dolor sit amet, con
9 </body>
10 </html>
```



Error found while checking this document as XHTML 1.0 Strict!

Validation Output: 1 Error

✖ Line 8, Column 1: character data is not allowed here

 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur eleifend, ma...

 You have used character data somewhere it is not permitted to appear. Mistakes that can cause this error include:

- putting text directly in the body of the document without wrapping it in a container element (such as a <p>paragraph</p>) or

<p>

Éléments parents

Les éléments affichés en rouge doivent obligatoirement contenir des sous-éléments.

blockquote body button dd del div fieldset form ins k map noscript object td th

Éléments enfants

Les éléments affichés en rouge doivent obligatoirement contenir des sous-éléments.

#PCDATA a abbr acronym b bdo big br button cite code del dfn em i img input ins kbd label map object q samp

Éléments enfants

Les éléments affichés en rouge doivent obligatoirement contenir des sous-éléments.

#PCDATA a abbr acronym b bdo big br button cite code del dfn em i img input ins kbd label map object q samp
script select small span strong sub sup textarea tt var

↑
≡ texte

<body>

Éléments parents

Les éléments affichés en rouge doivent obligatoirement contenir des sous-éléments.

html

Éléments enfants

Les éléments affichés en rouge doivent obligatoirement contenir des sous-éléments.

addressblockquotedeldivdlfieldsetformh1h2h3h4h5h6hrinsnoscriptolpprescripttableul

! pas de texte !

Mais OK en HTML5 !

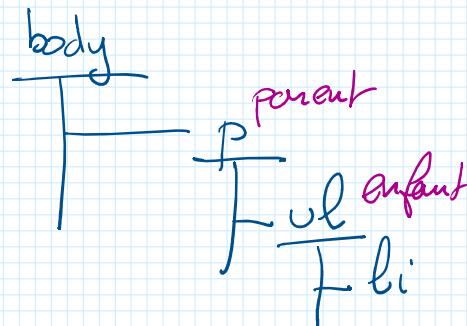
Document checking completed. No errors or warnings to show.

pas une bonne pratique

- **Search engine optimization:** Machines reading your content (e.g. Google's search index crawler) will **know** that this is a list and can react accordingly. This is better understandable when looking at headings: an <h1> element will be treated by search engines with higher importance than a element (which doesn't have any semantic meaning).
- **Accessibility:** Assistive software (e.g. screen readers for blind or vision impaired surfers) can tell that this is a list, and users can navigate it accordingly (e.g. jump from one list item to the next).

Case Study N°2 : list as a child of p

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Document</title>
6 </head>
7 <body>
8 <p>Lorem ipsum dolor sit amet, cor
9
10 Item 1
11 Item 2
12
13 </p>
14 </body>
15 </html>



Error found while checking this document as XHTML 1.0 Strict!

Line 9, Column 4: document type does not allow element "ul" here; missing one of "object", "ins", "del", "map", "button" start-tag

en HTML 5!

</p> vu

non vu

1. Error No p element in scope but a p end tag seen.

From line 13, column 1 to line 13, column 4

<p></p>

Ouverture ?

```

1  !DOCTYPE html
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Document</title>
6  </head>
7  <body>
8  <p>Lorem ipsum dolor sit amet, cor
9  <ul>
10     <li>Item 1</li>
11     <li>Item 2</li>
12 </ul>
13 </p> Formeture vue !
14 </body>
15 </html>

```

(en situation) a)

xHTML

Pour comprendre

Modélisation HTML 5 b)

Relations permises parent / enfant en xHTML

Éléments parents

Les éléments affichés en rouge doivent obligatoirement contenir des sous-éléments.

blockquote body button dd del div fieldset form ins li map noscript object td th

xHTML
pour <p>

Éléments enfants

Les éléments affichés en rouge doivent obligatoirement contenir des sous-éléments.

#PCDATA a abbr acronym b bdo big br button cite code del dfn em i img input ins kbd label map object q samp

script select small span strong sub sup textarea tt var

pas de !

douc

solution

<p> texte </p>

 —

en Modélisation HTML 5

Résumé technique

éléments enfants

| | |
|-----------------------|--|
| Catégories de contenu | Contenu de flux, contenu tangible. |
| Contenu autorisé | Contenu phrasé. |
| Omission de balises | La balise de début est obligatoire. La balise de fin peut être absente si l'élément <code><p></code> est immédiatement suivi d'un élément <code><address></code> , <code><article></code> , <code><aside></code> , <code><blockquote></code> , <code><div></code> , <code><dl></code> , <code><fieldset></code> , <code><footer></code> , <code><form></code> , <code><h1></code> , <code><h2></code> , <code><h3></code> , <code><h4></code> , <code><h5></code> , <code><h6></code> , <code><header></code> , <code><hr></code> , <code><menu></code> , <code><nav></code> , <code></code> , <code><pre></code> , <code><section></code> , <code><table></code> , <code></code> ou d'un autre élément <code><div></code> ou s'il n'y a plus de contenu dans l'élément parent et que l'élément parent n'est pas un élément <code><a></code> . |
| Parents autorisés | Tout élément qui accepte du contenu de flux. |
| Rôles ARIA autorisés | Tous les rôles sont autorisés. |
| Interface DOM | <code>HTMLParagraphElement</code> (angl.) |

éléments parents

Solution

`<p> texte` pas de `</p>`

$\Rightarrow <\ul> \li - \li $

~~`</p>`~~ pas de `<p>` vu !

pour `` en xHTML

Éléments parents

Les éléments affichés en rouge doivent obligatoirement contenir des sous-éléments.

blockquote body button dd del div fieldset form ins li map noscript object td th

pas de `<p>` !

Éléments enfants

Les éléments affichés en rouge doivent obligatoirement contenir des sous-éléments.



en HTML5

Phrasing content

Phrasing content, a subset of flow content,
Sequences of phrasing content make up pa

The phrasing elements are:

- `<a>`
- `
`
- ``
- `<big>`
- ``
- `<i>`
- `<small>`
- ``
- `<u>`

Contenu phrasé

The phrasing elements are:

- [`<abbr>`](#)
- [`<audio>`](#)
- [``](#)
- [`<bdi>`](#)
- [`<bdo>`](#)
- [`
`](#)
- [`<button>`](#)
- [`<canvas>`](#)
- [`<cite>`](#)
- [`<code>`](#)
- [`<data>`](#)
- [`<datalist>`](#)

Contenu phrasé

Le contenu phrasé définit le texte et le balisage qu'il contient. Des séquences de contenu phrasé constituent des paragraphes.

Les éléments appartenant à cette catégorie sont [`<abbr>`](#), [`<audio>`](#), [``](#), [`<bdo>`](#), [`
`](#), [`<button>`](#), [`<canvas>`](#), [`<cite>`](#), [`<code>`](#), [`<command>`](#), [`&`](#), [`<data>`](#), [`<datalist>`](#), [``](#), [`<details>`](#), [`<dfn>`](#), [`<div>`](#), [`<dl>`](#), [``](#), [`<embed>`](#), [`<fieldset>`](#), [`<figure>`](#), [`<footer>`](#), [`<form>`](#), [`<h1>`](#), [`<h2>`](#), [`<h3>`](#), [`<h4>`](#), [`<h5>`](#), [`<h6>`](#), [`<header>`](#), [`<hgroup>`](#), [`<hr>`](#), [`<i>`](#), [`<iframe>`](#), [``](#), [`<input>`](#), [`<ins>`](#), [`<kbd>`](#), [`<label>`](#), [`<main>`](#), [`<map>`](#), [`<mark>`](#), [`<math>`](#), [`<menu>`](#), [`<meter>`](#), [`<nav>`](#), [`<noscript>`](#), [`<object>`](#), [``](#), [`<output>`](#), [`<p>`](#), [`<picture>`](#), [`<pre>`](#), [`<progress>`](#), [`<q>`](#), [`<ruby>`](#), [`<s>`](#), [`<samp>`](#), [`<script>`](#), [`<section>`](#), [`<select>`](#), [`<small>`](#), [``](#), [``](#), [`<sub>`](#), [`<sup>`](#), [`<svg>`](#), [`<table>`](#), [`<template>`](#), [`<textarea>`](#), [`<time>`](#), [``](#), [`<var>`](#), [`<video>`](#), [`<wbr>`](#) et du texte brut (n'étant pas une simple suite de blancs).

Contenu de flux

Les éléments appartenant à la catégorie de contenu de flux contiennent généralement du texte ou du contenu intégré. Ces éléments sont : [`<a>`](#), [`<abbr>`](#), [`<address>`](#), [`<article>`](#), [`<aside>`](#), [`<audio>`](#), [``](#), [`<bdo>`](#), [`<bdi>`](#), [`<blockquote>`](#), [`
`](#), [`<button>`](#), [`<canvas>`](#), [`<cite>`](#), [`<code>`](#), [`<command>`](#), [`&`](#), [`<data>`](#), [`<datalist>`](#), [``](#), [`<details>`](#), [`<dfn>`](#), [`<div>`](#), [`<dl>`](#), [``](#), [`<embed>`](#), [`<fieldset>`](#), [`<figure>`](#), [`<footer>`](#), [`<form>`](#), [`<h1>`](#), [`<h2>`](#), [`<h3>`](#), [`<h4>`](#), [`<h5>`](#), [`<h6>`](#), [`<header>`](#), [`<hgroup>`](#), [`<hr>`](#), [`<i>`](#), [`<iframe>`](#), [``](#), [`<input>`](#), [`<ins>`](#), [`<kbd>`](#), [`<label>`](#), [`<main>`](#), [`<map>`](#), [`<mark>`](#), [`<math>`](#), [`<menu>`](#), [`<meter>`](#), [`<nav>`](#), [`<noscript>`](#), [`<object>`](#), [``](#), [`<output>`](#), [`<p>`](#), [`<picture>`](#), [`<pre>`](#), [`<progress>`](#), [`<q>`](#), [`<ruby>`](#), [`<s>`](#), [`<samp>`](#), [`<script>`](#), [`<section>`](#), [`<select>`](#), [`<small>`](#), [``](#), [``](#), [`<sub>`](#), [`<sup>`](#), [`<svg>`](#), [`<table>`](#), [`<template>`](#), [`<textarea>`](#), [`<time>`](#), [``](#), [`<var>`](#), [`<video>`](#), [`<wbr>`](#) et le texte.