

# HTML

<b>HTML</b>	<b>1</b>
<b>Editores HTML</b>	<b>1</b>
<b>Navegadores web</b>	<b>2</b>
<b>HTML5</b>	<b>2</b>
HTML5. Guía de estilo e convencións de código	4
Validación	6
<b>Repaso HTML5</b>	<b>6</b>
Ligazóns HTML	6
Listas	7
Táboas	8
Cores	9
Imaxes HTML	10
Elementos de bloque e inline	11
Formularios	11
<b>Modelos</b>	<b>12</b>
<b>Referencias</b>	<b>12</b>

## HTML

A orixe da linguaxe HTML remóntase a finais da década dos 80, cando o físico Tim Berners-Lee, investigador do CERN (Organización Europea para a Investigación Nuclear) propoñía un novo sistema "hipertexto" para compartir documentos.

A partir de 1997, os estándares de HTML públicaos o W3C (World Wide Web Consortium). A súa primeira recomendación foi a versión HTML 3.2.

Dende a publicación de HTML 4.01, a actividade de estandarización de HTML parouse e o W3C centrouse no desenvolvemento do estándar XHTML. Por este motivo, en 2004, as empresas Apple, Mozilla, Opera e máis tarde tamén Google mostraron a súa preocupación pola falta de interese do W3C en HTML e decidiron crear unha nova asociación denominada WHATWG (Web Hypertext Association Technology Working Group).

XHTML 1.0, publicado no ano 2000, é unha adaptación de HTML 4.01 á linguaxe XML, así que mantén case todas as súas etiquetas e características pero engade algunhas restricións e elementos propios de XML. A versión XHTML 2.0 nunca chegou a ser unha recomendación para W3C, e ante o seu fracaso, a finais de 2006, tomou como base as investigacións de WHATWG para seguir traballando con HTML5. Así que HTML5 xa é un estándar recomendado polo W3C dende o 6 de agosto de 2013.

En xullo de 2012, ante as dificultades do traballo en común e a diferenza de obxectivos respecto de W3C, o WHATWG decide separarse e seguir o seu propio camiño de forma totalmente independente. A consecuencia é que, a partir de entón, o WHATWG proporá unha versión "viva" (Living Standard) do HTML e o W3C levará a cabo a normalización da linguaxe.

O HTML5 Living Standard foi publicado en 2012 e esa en continua actualización.

O WHATWG continúa coas súas investigacións independentemente do W3C, e ao HTML chámasele HTML Living Standard para non confundilo co HTML5 de W3C.

A última recomendación da W3C é o HTML5.2 publicado en decembro de 2017.

## Editores HTML

Existen unha grande cantidade de editores HTML que permiten a creación e edición de páxinas web. Algúns destes editores son: Brackets, Sublime Text, Atom, etc.

Tamén existen editores en liña que permiten escribir o código directamente dende o navegador. Algúns exemplos son: [editor da w3cschools](#), [codepen](#), [HTML online editor](#), [jseditor](#), etc.

## Navegadores web

Un navegador web (*browser*) é un software que permite a visualización de documentos de hipertexto xunto con outros tipos de arquivos multimedia, como son as imaxes, vídeos, sons e animacións.

Hai que ter en conta que non todos os navegadores interpretan o código HTML e CSS da mesma maneira, e que entre eles existen variacións na visualización da páxina web que fan que o resultado non sexa o mesmo con un que con outro. Pode incluso que algunha parte da páxina non se vexa ou non funcione.

Consideracións a ter en conta para mellorar a compatibilidade dun sitio web cos diferentes navegadores:

- Validar o código do sitio web: se o código non está ben escrito, cada navegador pode ter unha interpretación diferente para o mesmo.
- Inicializar os estilos CSS: cada elemento HTML ten un estilo CSS por defecto e non son exactamente iguais en todos os navegadores. Se se inicializan os valores por defecto, contrólase o aspecto de cada elemento de maneira máis efectiva.

## HTML5

HTML5 é a versión actual da linguaxe HTML.

Algunhas das características de HTML5 son:

- Permite describir con maior precisión cal é o contido dunha páxina web dotándoa de **semántica**. A web semántica propón a inclusión de metadatos na web para que, por exemplo, os robots dos motores de busca poidan procesar o seu contido e facer deducións lóxicas para dar unha resposta máis acertada. Elementos semánticos: <header>, <footer>, <article> e <section>.
- Proporciona unha maior optimización da velocidade e un maior uso do hardware a través da introdución dun deseño adaptativo (*responsive*). Dispón de técnicas para que unha web sexa visible de maneira óptima en todo tipo de dispositivos capaces de visualizar páxinas web.
- Utiliza CSS3 que ofrece unha grande variedade de opcións para facer deseños máis sofisticados.
- Proporciona soporte para utilizar contido multimedia como audio e vídeo.
- Proporciona unha ampla gama de novas características para visualizar gráficos na web en 2D e 3D.

Estrutura dunha páxina HTML:



Algunhas das novidades de HTML5 son:

- O tipo do documento

**<!doctype html>**

- A codificación de caracteres:

**<meta charset="UTF-8">**

- Elementos gráficos: <svg> (scalable vector graphics) e <canvas> (usando scripting)
- Elementos multimedia: <audio> e <video>.

Exemplo HTML5:

```
<!doctype html>
<html lang="gl">

<head>
<meta charset="UTF-8" />
<meta name="keywords" content="HTML, CSS, XML" />
<meta name="description" content="Descrición da páxina" />
<title>Título do documento</title>
</head>

<body>
Contido do documento.....
</body>

</html>
```

A declaración do idioma na etiqueta html axuda ao software de lectura que se executa sobre a páxina.

Máis información de HTML5: <https://www.w3schools.com/html/default.asp>

## HTML5. Guía de estilo e convencións de código

HTML5 non é tan estrito como XHTML, sen embargo, recoméndase seguir as convencións de código:

- Usar minúsculas nos nomes dos elementos.
- Cerrar todas as etiquetas de elementos.
- Cerrar elementos baleiros. En HTML5 é opcional, sen embargo se algún software XML vai acceder á páxina, é recomendable cerralos.

**<meta charset="UTF-8" />**

- Usar minúsculas para os nomes dos atributos.

**<div class="menu">**

- Entrecomillar valores de atributos. Non é obrigatorio en HTML5, mais é recomendable:

**<table class="striped">**

- HTML5 permite espazos ao redor do símbolo "=", sen embargo evitalos fai que a lectura sexa máis fácil.

**<link rel="stylesheet" href="styles.css">**

- Engadir sempre o atributo **alt** ás imaxes. Este atributo é útil cando a imaxe non pode ser amosada por calquera motivo.

Tamén se recomenda definir o ancho e alto da imaxe para evitar parpadeos ao cargar a páxina, pois o navegador pode reservar o espacio antes de cargar a imaxe:

```

```

**NOTA:** o alto e ancho tamén se poden especificar cos atributos **width** e **height**, aínda que se recomenda usar estilos para evitar que unha folla de estilos cambie o tamaño das imaxes.

- Evitar liñas de máis de 80 caracteres.
- As liñas en branco favorecen a lectura cando separan bloques de código longos. Non engadir liñas en branco sen motivo.
- Para a sangría son suficientes dous espazos en branco.
- Aínda que HTML5 permite a omisión de certas etiquetas (<html>, <body>, <head>) non é recomendable facelo.
- Comentarios:

```
<!-- Este é un comentario -->
```

```
<!--
```

```
Exemplo dun comentario longo. Exemplo dun comentario longo.
```

```
Exemplo dun comentario longo. Exemplo dun comentario longo.
```

```
-->
```

- Exemplos para indicar estilos:

```
<link rel="stylesheet" href="styles.css">
```

```
p.intro {font-family: Verdana; font-size: 16em;}
```

```
body {
  background-color: lightgrey;
  font-family: "Arial Black", Helvetica, sans-serif;
  font-size: 16em;
  color: black;
}
```

- A chave de apertura debe ir na mesma liña que o selector.
- Usar un espacio antes da chave de apertura.
- Usar dous espazos para a sangría.
- Usar punto e coma (;) despois de cada par **propiedade-valor**, incluída a última.
- Só usar comiñas se o valor contén espazos.
- Colocar a chave de cerre nunha nova liña sen espazos diante.
- Evitar liñas de máis de 80 caracteres.

- Cargar JavaScript:

**<script src="myscript.js">**

- Usar minúsculas nos nomes dos arquivos. Algúns servidores web son sensibles ás maiúsculas. Para evitar erros, recoméndase usar nomes en minúsculas.
- Caracteres reservados en HTML. Se se quere usar o carácter <, este pode ser confundido con unha etiqueta. En vez de < deberase escribir **&lt;** ou **&#60;**.
- *Non-breaking Space (&nbsp;)*. O carácter **&nbsp;** representa un espacio que non suporá un salto de liña.

Dúas palabras separadas por un **&nbsp;** permanecerán xuntas na mesma liña. Isto úsase cando non se poden separar dúas palabras. Por exemplo 10 km/h ou 10:00 pm.

Tamén se usa para evitar que os navegadores eliminen os espazos en branco por defecto. Se se quere engadir espazos en branco, debe usarse este carácter.

Poden consultarse outros caracteres reservados na seguinte [páxina](#).

## Validación

Resulta de grande utilidade poder verificar os erros que pode haber nas páxinas.

Existe unha ferramenta en liña accesible dende <https://validator.w3.org/> que permite validar o código, ben indicando a URL, subindo o documento HTML ou escribindo o código directamente.

## Repaso HTML5

Na páxina web <https://www.w3schools.com/html/default.asp> hai un tutorial completo de HTML5.

Nos seguintes apartados vaise facer un repaso dos principais elementos.

## Ligazóns HTML

A sintaxe dunha ligazón é:

**<a href="url">texto da ligazón</a>**

Por exemplo:

**<a href="<https://www.w3schools.com/html/>">Visita o noso tutorial</a>**

Nota: se non se engade a barra do final (/), pódense xerar dúas peticións ao servidor. Moitos servidores engadirán automaticamente a barra á dirección e crearán unha nova petición.

Valores que pode tomar o atributo href:

- Unha páxina da nosa web: recoméndase usar rutas relativas.

**href="paxina2.html"**

- Unha páxina doutra web.

**href="http://www.iessanclemente.net"**

- Un marcador na mesma páxina:

O marcador ponse co atributo id: **<p id="marcador1">**

A ligazón incluíría: **href="#marcador1"**

- Un marcador dentro doutra páxina web.

**href="paxina2.html#marcador1"**

O atributo **target** especifica onde abrir a ligazón:

- **\_blank**: nunha nova ventá.
- **\_self**: na mesma ventá onde se fixo click.
- **\_parent**: no frame pai.
- **\_top**: na ventá completa.

Exemplo de creación dunha ligazón a un marcador:

**<h2 id="C4">Capítulo 4</h2>**

**<a href="#C4">Ir ao Capítulo 4</a>**

**<a href="html\_demo.html#C4">Ir ao capítulo 4</a>**

**NOTA:** o atributo id debe ser único en todo o documento HTML.

## Listas

Listas desordenadas:

```
<ul style="list-style-type:disc;">
  <li>Elemento 1</li>
  <li>Elemento 2</li>
  <li>Elemento 3</li>
</ul>
```

A propiedade CSS *list-style-type* pode ser: *disc*, *circle*, *square*, *none*.



Listas ordenadas:

```
<ol type="1">
  <li>Elemento 1</li>
  <li>Elemento 2</li>
  <li>Elemento 3</li>
</ol>
```

O atributo *type* define o tipo da lista. Este atributo pode tomar os valores: 1, A, a, I, i.

Listas de descrições ou definiciones:

```
<dl>
  <dt>Elemento 1</dt>
  <dd>- definición elemento 1</dd>
  <dt>Elemento 2</dt>
  <dd>- definición elemento 1</dd>
</dl>
```

## Táboas

Exemplo de táboa en HTML5:

```
<table style="width:100%">
  <caption>Persoaxes</caption>  <!-- título da táboa -->
  <tr>                             <!-- inicio de fila -->
    <th>Nome</th>                 <!-- cabeceira -->
    <th>Apelido</th>
  </tr>
  <tr>
    <td>Ada</td>                  <!-- inicio de cela -->
    <td>Lovelace</td>
  </tr>
</table>
```

Para definir o borde dunha táboa en HTML5 hai que usar CSS. Recordar definir o borde para a táboa, a cabeceira e as celas.

```
table, th, td {
  border: 1px solid black;
  padding: 15 px;    <!-- espacio entre o contido da cela e o seu borde -->
  border-spacing: 5px;    <!-- espacio entre celas -->
}
```

Outros atributos:

- *colspan*: especifica que unha cela se expanda por varias columnas.
- *rowspan*: especifica que unha cela se expanda por varias filas.

## Cores

En HTML5 as cores poden especificarse utilizando:

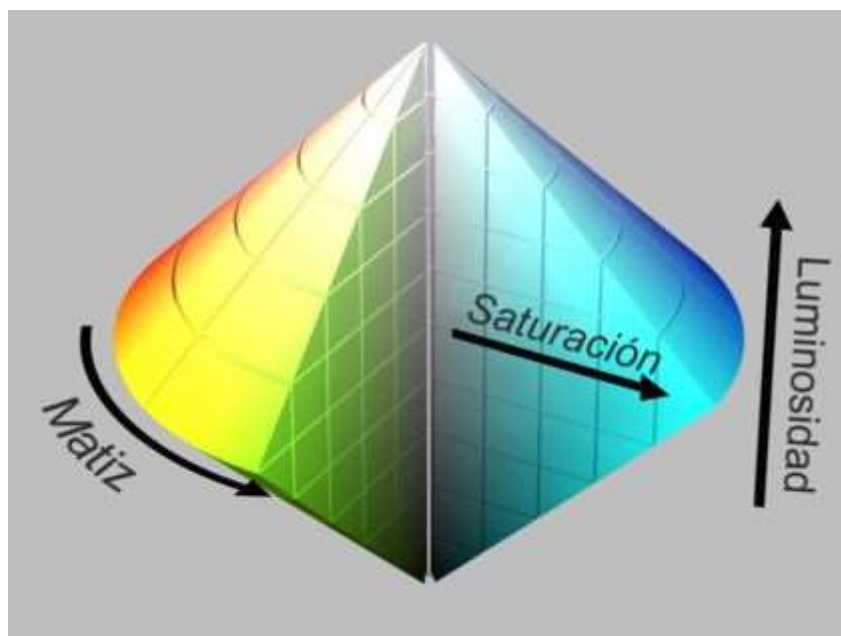
- O seu nome: os navegadores modernos soportan os seguintes 140 [nomes de cores](#).
- O valor RGB, o código hexadecimal, o valor RGBA ou o valor HSLA. Así por exemplo a cor “tomato” corresponde:
  - **rgb(255, 99, 71)**
  - **#ff6347**
  - **hsl(9, 100%, 64%)**

A mesma cor cunha transparencia do 50%.

- **rgba(255, 99, 71, 0.5)**
- **hsla(9, 100%, 64%, 0.5)**

HSL (Hue, Saturation, Lightness - Matiz, Saturación, luminosidade) é un modelo de cor en base aos seus compoñentes constituíntes.

O modelo HSL represéntase graficamente como un cono dobre. Os dous vértices correspóndese co branco e o negro, o ángulo correspóndese co matiz, a distancia ao eixe corresponde coa saturación e a distancia ao eixe branco-negro corresponde coa luminosidade.



O matiz é un grao nunha roda de cores de 0 a 360 graos. Vermello é 0, verde é 120 e azul é 240. [Exemplo](#).

A saturación pode describirse como a intensidade de cor. 100% é cor puro e 50% é metade de gris, pero aínda pode verse a cor. 0% é completamente gris. [Exemplo](#).

A luminosidade indica canta luz ten unha cor. Exprésase como unha porcentaxe. Negro significa sen luz e represéntase co 0% e branco significa completamente iluminado e represéntase co 100%. [Exemplo.](#)

O modelo RGBA baséase no modelo RGB ao que engade un canal alfa que especifica a opacidade da cor. O valor alfa é un número entre 0.0 (totalmente transparente) e 1.0 (completamente opaco). [Exemplo.](#)

O modelo HSLA é unha extensión do modelo HSL ao que se engade un canal alfa que especifica a opacidade da cor. O valor alfa é un número entre 0.0 (totalmente transparente) e 1.0 (completamente opaco). [Exemplo.](#)

## Imaxes HTML

Un exemplo de inclusión dunha imaxe en HTML5 é o seguinte:

```

```

Recoméndase usar rutas relativas sempre que sexa posible, así non se depende do directorio raíz no que estea aloxada a páxina.

Debe engadirse sempre o atributo *alt*, que aparecerá sempre que non poida mostrarse a imaxe.

Por defecto a imaxe móstrase co seu tamaño orixinal. Poden usarse os atributos *width* e *height* para modificar ese tamaño, ou para que o navegador xa saiba o tamaño antes de descargar a imaxe e así deixe o oco adecuado para a mesma.

HTML5 introduciu o elemento **<picture>** para engadir flexibilidade. Este elemento contén un número de fontes (**source**), cada un referíndose a unha imaxe distinta para que cada navegador escolla a imaxe que mellor se adapte ao dispositivo. Cada **source** ten atributos que describen cando será máis axeitado escoller esa fonte. O navegador escollerá o primeiro **source** que se adapte, ignorando o resto.

Exemplo:

```
<picture>
  <source media="(min-width: 650px)" srcset="img_pink_flowers.jpg">
  <source media="(min-width: 465px)" srcset="img_white_flower.jpg">
  
</picture>
```

Recoméndase especificar o elemento `<img>` como última fonte. Esta será usada en navegadores que non soporten o elemento `<picture>` ou ningunha das outras `<source>` encaixan.

## Elementos de bloque e inline

Cada elemento en HTML ten unha visualización predeterminada que depende do propio elemento.

Os valores de visualización son: bloque e *inline*.

- Os elementos de bloque sempre empezan nunha nova liña e ocupan todo espazo dispoñible de esquerda a dereita. Exemplo: <div>.
- Os elementos *inline* non empezan nunha nova liña e só ocupan o espazo necesario. Exemplo: <span> (usado habitualmente para aplicar estilo ao texto).

Poden consultarse os elementos en bloque e *inline* nesta [páxina](#).

## Formularios

Un formulario <form>, contén diferentes elementos:

- <input type="text" name="nome">: campo de entrada de texto dunha soa liña
- <input type="radio" name="profesion">: botón de opción para seleccionar só unha opción.
- <input type="checkbox" name="profesion">: caixas de verificación.
- <input type="submit" name="enviar">: botón de envío. A páxina que vai procesar os datos do formulario especificase no atributo **action** do formulario.
- <select name="coches">: define unha lista despregable [Exemplo](#).
- <textarea name="mensaxe" ...>: define un área de texto.
- <datalist>: define unha lista con opcións predefinidas que permite o autocompletado dun campo de texto. Está asociada cun elemento *input*. [Exemplo](#).

Se se omite o atributo **name**, non se envía a información.

Na seguinte [páxina](#) poden verse máis elementos que permiten introducir datos nun formulario.

O método de envío empregado polo formulario especificase no atributo **method**. Os seus valores poden ser GET ou POST. O método por defecto de envío do formulario é GET. Cando se usa este método, a información enviada é visible no campo de dirección da páxina:

*/action\_page.php?firstname=Mickey&lastname=Mouse*

Notas sobre GET:

- A lonxitude da URL está limitada a 3000 caracteres.
- Non se debe usar GET para enviar información sensible.

Recoméndase usar POST cando a información a enviar é sensible.

HTML5 engadiu o atributo **autocomplete** para o formulario que especifica ao navegador que complete os elementos do formulario cos valores introducidos polo usuario anteriormente. Este atributo pode estar activo ou non. Tamén pode estar activo para o formulario e desactivarse para algúns elementos do formulario.

## Modelos

Un modelo web (*template*) é un conxunto de documentos web predeseñados que se poden utilizar para inserir contido e imaxes para desenvolver a interface do sitio web.

A principal vantaxe do uso dun modelo é que permite reducir considerablemente o tempo de desenvolvemento e, en consecuencia, o seu custe. Sen embargo, hai que avaliar se a súa utilización é apropiada, xa que en ocasións non permiten un deseño e maquetación do sitio personalizado.

Consideracións á hora de escoller un modelo:

- O deseño do modelo debe ser adaptable a diferentes tipos de dispositivos.
- Que estea optimizado para o posicionamento SEO (*Search Engine Optimization*), polo menos que non teña erros que impida aparecer nos primeiros resultados dos buscadores.
- Que teña soporte para diferentes idiomas.
- Que sexa compatible coa maioría dos navegadores.
- Hai que ter en conta a usabilidade do modelo, en canto ao tempo de carga e facilidade de modificación.

NOTA: hai modelos de uso gratuíto e outros de pago.

Páxinas de descarga de modelos:

- <https://html5up.net/>
- <https://templated.co/>
- <https://www.zertheme.com/>

Tamén se poden obter modelos para traballar cos diferentes *frameworks*, como por exemplo **Bootstrap**:

- <https://startbootstrap.com/>
- <https://wrapbootstrap.com/tag/seo>
- <http://themesseo.com/cat/bootstrap>

## Referencias

Para a elaboración deste material utilizáronse, entre outros, os recursos que se enumeran a continuación:

- [Diseño de interfaces web \(xtec.cat\)](https://ioc.xtec.cat/materials/FP/Recursos/fp_daw_m09_/web/fp_daw_m09_htmlindex/index.html)  
[https://ioc.xtec.cat/materials/FP/Recursos/fp\\_daw\\_m09\\_/web/fp\\_daw\\_m09\\_htmlindex/index.html](https://ioc.xtec.cat/materials/FP/Recursos/fp_daw_m09_/web/fp_daw_m09_htmlindex/index.html)
- <https://www.w3schools.com/html/default.asp>