

# Estilo orrialdeak

CSS-a (Cascading Style Sheets) web orrialdeei estiloa eta diseinua emateko erabiltzen dira.

CSS adierazpen hizkuntza bat da, non nabigatzaileak adierazpen horiek aukeratutako elementuei aplikatzen dizkie behar bezala erakusteko, webguneen itxurari formatua emanaz.

Hauek nabigatzaileari web-orriaren edukia nola bistaratu behar duten adierazten die: letra iturria aldatu, kolorea, tamaina, edukiaren arteko espazioa, zatitu hainbat zutabeetan, etabar.

Web inguruan dauden hiru teknologi nahusietako bat da CSS-a, HTML eta JavaScript-arekin batera.

CSS sortu zenean, ideia nagusia zen estilo-orriekin espezifikazio bat sortzea HTML web-orriaren edukiak eta aurkezpenaren edukiak bereizteko.

CSS gabe, hasiera baten web diseinatzaileek orriak HTML taulak bezalako teknikekin diseinatu behar zituzten, ikusmen-arazoak dituzten erabiltzaileen irisgarritasuna zailtzen.

Horrelako elementuak erabiltzen ez dituzten diseinuek asko hobetzen dituzte webaren irisgarritasuna:

- Pantaila-irakurgailuek arazo gutxiago dituzte taularik gabeko diseinuekin, egitura logikoari jarraitzen diotelako.
- Gauza bera gertatzen da bilaketa-motorren web arakatzailerekin, orrialdeak zehaztasunez sailkatu ditzakete eta erabiltzaileei bilaketa egokiak eta erraz aurkitzeko aukera emaneten dizkie.
- Diseinua (CSS) eta egitura (HTML) bereiztearen ondorioz, gailu desberdinetarako diseinu desberdinak ere eskain daitezke.

World Wide Web Consortium (W3C) organismoa arduratzen da honen zehaztapena arautzeaz.

1995ean, W3Ck bere HTML lantaldeari CSSren garapena eta estandarizazioa gehitu zizkion, eta 1996an lehen gomendio ofiziala argitaratu. Egungo estandarra **CSS3** da eta guztiz bateragarria da aurreko bertsioekin.

CSS3 gaur egungo web orrien beharrak asetzeko diseinututa dago, eta HTML5-rekin batera web garapeneko estandar berriak definitzen ditu, kodea birdiseinatzeko arazoak konpontzeko eta premia berrietara eguneratuz.

- CSS arauak idazteko modu berri asko eskaintzen ditu hautatzaile berriekin.
- Askoz errazagoa da diseinua **Flex** eta **Grid** edukiontzia erabiliz.
- **Media-query**-ak erabiltzea ahalbidetzen du. Hauek estilo-orri bat edo beste aukeratzeko aukera ematen dute ditugun pantailako propietateen arabera (mugikorra, tableta, telebista, etab.)
- Besteak beste, **kolore eta degradatu eskema** desberdinak (RGBA, HSLA, HSL, etab.), **gradientak**, **ertz biribilduak**, **itzalak**, **gardentasunak** eta animazioak eskaintzen ditu.
- Errendimendu azkarra eta bikaina eskaintzen du eta ez du hainbeste memoria erabiltzen.

Web orri bat sortzean, lehenik eta behin HTML/XHTML lengoaia erabiltzen da elementu bakoitzak orri barruan duen funtzioa izendatzeko: paragrafoak, goiburuak, testu nabarmenak, etab.

Edukiak sortu ondoren, elementu bakoitzaren formatua definitzeko CSS lengoaia.

Elementu bati estilorik ez bazaio adierazten nabigatzaileak ezarriko dizkio bere estilo lehenetsiak.

**CSS propietateen zerrenda** **CSS balio lehenetsiak**

## CSS sintaxia

Elementu bati CSS arauak gehitzeko atributu globala **<style>** da.

CSS arau bat hautatzaile batek eta deklarazio bloke batek osatzen dute:



- Hautatzaileak (*selector*) HTML elementura seinalatzen du estiloa, erudian *h1* elementuetara.
- Adierazpen-blokeak giltzen artean deklarazio bat edo gehiago ditu puntu eta komaz bereiztuta.
- Adierazpen bakoitzak CSS propietatearen izena eta balio bat ditu, bi puntuz bereizita.

Aurreko hautatzailearen arauak orrialdeko *h1* elementu guztiak kolore urdinez era 12 pixeleko tamainan ezarriko ditu.

## HTMLari CSSa aplikatzen

*Style* atributua erabiltzeko era desberdinak daude:

### Lerroko estiloak

*Style* atributuak hurrengo sintaxia du:

```
<tagname style="propietatea:balioa;">
```



Adibidez:

```
<!DOCTYPE html>
<html>
  <body>
    <p>I am normal</p>
    <p style="color:red;">I am red</p>
    <p style="color:blue;">I am blue</p>
    <p style="font-size:25px;">I am bigger</p>
  </body>
</html>
```

I am normal

I am red

I am blue

I am bigger

## Barne-estilo orriak

Oso ohikoa izaten da mota bereko elementu guztiei estilo bera aplikatu nahi izatea.

Araua behin era berrir ez errepikatzeko, HTML dokumentuaren goiburuan *style* definizioa ere erabili daiteke.

Orrialde beran estiloak definitizko *style* ireki eta itxeko etiketak definitzen dira:

```
<!DOCTYPE html>
<html>
  <head>
    <style></style>
  </head>
</html>
```

Sortutako *style* etiketen barruan, interesa duten elementuetarako estiloak defini daitezke.

Elementuei erreferentzia egiteko **hautatzaile** edo **selektore** bat adierazten da (elementuaren etiketa), eta giltzen artean, elementu horiei aplikatu nahi zaizkien estilo guztiak adierazten dira.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      tagname {
        property1name: property1value;
        property2name: property2value;
      }
    </style>
  </head>
</html>
```

Hurrengo ereduan:

- Parrafo guztiek letra etzana eta kolore gorria izango dute.
- Esteka guztiek salmoi atzeko-kolore bat eta letra lodia izango dute.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p {
        font-style: italic;
        color: red;
      }
      a {
        background: salmon;
        font-weight: bold;
      }
    </style>
  </head>
  <body>
    <p>I am a paragraph</p>
    <p>I am another paragraph</p>
    <a href="#">I am an hyperlink</a>
    <a href="#">I am another hyperlink</a>
  </body>
</html>
```

*I am a paragraph*

*I am another paragraph*

**I am an hyperlink**

**I am another hyperlink**

## Kanpoko estilo orriak

Elementuen estiloak *style* barruan definitzen diren heinean goiburua hazten doa; horregatik, praktika on bat izaten da estiloak HTML dokumentutik kanpo ateratzea eta fitxategi independente batean definitzea.

Hau da lan egiteko **gomendatzen** den era, ohikoena eta erabilgarriena: dokumentuaren goiburutik CSS fitxategi independente bat lotzea.

.css formatua duen dokumentu bat sortzen da, eta bertan, aplikatu nahi diren estilo guztiak zehazten dira.

HTML dokumentuaren eta kanpo-baliabide horren arteko lotura zehazteko, **<link>** etiketa erabiltzen da.

Elementu "hutsa" da, itxeko beharrik ez duena, eta atributuak baino ez ditu.

Etiketa hau orokorrean estilo orriak lotzeko erabiltzen da, baina beste gauzetarako ere erabiltzen da, adibidez webguneko dokumentuari *favicon* bat gehitzeko, etab.

LINK tag

Hona hemen kanpoko estilo orrialde bat erabiltzeko sintaxia:

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

- Rel:

Derrigorrezko atributua da eta uneko dokumentuaren eta estekatutako agiriaren arteko lotura zehazten du

Ereduan estilo-orri bat dela (*stylesheet*) adierazten da, baina beste balio askorekin ere agertu daiteke.

- Href:

Atributuak estekatutako dokumentuaren kokapena zehazten du.

Aurreko adibidearekin jarraituz, honako hau litzateke emaitza:

<pre>index.html &lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;link rel="stylesheet" href="css/style.css"&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;p&gt;I am a paragraph&lt;/p&gt;     &lt;p&gt;I am another paragraph&lt;/p&gt;     &lt;a href="#"&gt;I am an hyperlink&lt;/a&gt;     &lt;a href="#"&gt;I am another hyperlink&lt;/a&gt;   &lt;/body&gt; &lt;/html&gt;</pre>	<pre>css/style.css p {   font-style: italic;   color: red; } a {   background: salmon;   font-weight: bold; }</pre>	<p><i>I am a paragraph</i></p> <p><i>I am another paragraph</i></p> <p><b>I am an hyperlink</b> <b>I am another hyperlink</b></p>
--	---	---

Kanpoko estiloak erreferentzatu ahal izateak Interneten diren beste orrialdeak partekatzeko aukera ematen du.

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css">
```

## Klaseak

Orain arteko eredueta, HTMLn jasotzen duten elementu-izenean oinarritzen diren izenak dituzten elementuetarako hautagailuak erabili dira.

Honek balio du mota bereko elementu guztiek dokumentuan itxura bera izatea nahi denean.

Hala ere, gehienetan ez da hala gertatzen, eta, beraz, elementu-azpimultzo bat aukeratzeko modu bat aurkitu behar da, besteak aldatu gabe.

Hori egiteko modurik ohikoena HTML elementuari atributuen bidez klasea bat gehitzea eta zehaztea da.

Horretarako, globala den **class** atributua definitzen da, eta honen bidez, elementu jakin batean izenen zerrenda bat (espazioka bereiztutak) definitu daiteke. Hautagailuen bidez izen horietara heldu daiteke elemetu hauek aukeratzeko.

Class atribututaren sintaxia:

```
<tagname class="name1 name2 name3">
```

CSSn, klase bat hauta daiteke, amaierako puntua eta klasearen izena jarraian dituen hautagailu bat sortuz.

```
.name1 { ... }
.name2 { ... }
.name3 { ... }
```

Hurrengo adibidean, *normal* eta *special* klaseak adierazten dira CSS arauetan, bakoitza ezaugarri desberdinak ezartzeko.

klasea bakoitza duten edozein elementuk estilo orrian klase hoietarako definitutako formatua hartzen dute.

Halaber, klaseekin batera HTML elementu hautatzaileak konbina daitezke.

Hurrengo adibidean, lehen bezala aurreko klaseak duten elementu guztiek haiek definitutako formatua hartzen dute.

Hirugarren araua berriz, klasearekin lotutako esteka elementuetan soilik aplikatuko da, paragrafoan ordea ez.

```

<!DOCTYPE html>
<html>
  <head>
    <style>
      .normal { background: salmon; }
      .special { background: lightgreen; }
    </style>
  </head>
  <body>
    <ul>
      <li>
        <a href="#" class="normal">
          Normal list item
        </a>
      </li>
      <li>
        <a href="#" class="normal">
          Another normal list item
        </a>
      </li>
      <li>
        <a href="#" class="special">
          Special list item
        </a>
      </li>
    </ul>
    <p class="normal">Normal paragraph</p>
    <p class="special">Special paragraph</p>
    <p>
      <a href="#" class="special">Special hyperlink</a>
    </p>
  </body>
</html>

```

- Normal list item
- Another normal list item
- Special list item

Normal paragraph

Special paragraph

Special hyperlink

```

<!DOCTYPE html>
<html>
  <head>
    <style>
      .normal { background: salmon; }
      .special { background: lightgreen; }
      a.special { background: orange; }
    </style>
  </head>
  <body>
    <ul>
      <li>
        <a href="#" class="normal">
          Normal list item
        </a>
      </li>
      <li>
        <a href="#" class="normal">
          Another normal list item
        </a>
      </li>
      <li>
        <a href="#" class="special">
          Special list item
        </a>
      </li>
    </ul>
    <p class="normal">Normal paragraph</p>
    <p class="special">Special paragraph</p>
    <p>
      <a href="#" class="special">Special hyperlink</a>
    </p>
  </body>
</html>

```

- Normal list item
- Another normal list item
- Special list item

Normal paragraph

Special paragraph

Special hyperlink

## IDak

ID atributuak HTML elementu bat identifikatzen du orrialde osoan.

HTML elementu batek id atributu bakarra izan ahal du, eta honek balio bakarra izan behar du orrian osoan zehar, ezin dira bi ID errepikatu.

Hau dela eta, diseinuaren aldetik elementu zehatz bat hautatu nahi denean id atributua erabili daiteke.

klase baten antzera egiten da, baina kasu honetan # sinboloa CSS hautatzailean adierazi behar da elementua identifikatzeko erabiltzen dugun balioaren aurrealdean.

```

<!DOCTYPE html>
<html>
  <head>
    <style>
      #list-item-1 { background: salmon; }
      #list-item-2 { background: lightgreen; }
      #list-item-3 { background: orange; }
    </style>
  </head>
  <body>
    <ul>
      <li id="list-item-1">List item 1</li>
      <li id="list-item-2">List item 2</li>
      <li id="list-item-3">List item 3</li>
    </ul>
  </body>
</html>

```

- List item 1
- List item 2
- List item 3

Ikusten denez, erabilera *class*-en oso antzekoa da. Hala ere, *class* batek hainbat elementu defini ditzake eta elementu bakar batek hainbat *class* hautatzaile izan ditzake; *id* atributuak, berriz, elementu bakar bat identifikatzen du.

Gehienetan, estiloetan *id* atributua baino hobea da *class* erabiltzea.

## Kaskada eta herentzia

Kontzeptu hauek ulertzeak estilo-orri oso indartsuak idazteko aukera ematen du eta, gainera, denbora aurreztuko da CSS arau gutxiago idatziz.

### Herentzia

CSSn, estilo batzuk HTML dokumentuaren zuhaitzean hedatzen dira; beste batzuk, aldiz, ez.

Honen diseinuaren atzean dagoen printzipioa CSS arau gutxiago idaztea dago.

Iturriaren jabetza heredatu egiten da. Horregatik erabiltzen dugu <body> elementua gure iturri-estiloak eransteko.

Gure HTMLaren body elementua gure beste HTML elementu guztien aita da (<head> atala izan ezik). Orrialdearen iturrian propietatea ezartzeak aukera ematen du dokumentuaren gainerakoari iturriaren definizioa jaraunsteko.

Herentziaren bidez hedatzen diren estiloak, oro har, dokumentuaren testuaren estiloarekin lotuta daude.

Adibidez, elementu batentzat kolorea (kolorea) eta letra-mota (font-family) ezartzen badira, elementu bakoitza ere kolore horrekin eta letra mota horrekin agertuko da, zuzenean kolore eta letra-mota desberdinak aplikatu ez bazaizkie ere.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        color: blue;
      }
      span {
        color: black;
      }
    </style>
  </head>
  <body>
    <p>As the body has been set to have a color of blue this is
    inherited through the descendants.</p>
    <p>We can change the color by targeting the element with a selector,
    such as this <span>span</span>.</p>
  </body>
</html>
```

As the body has been set to have a color of blue this is inherited through the descendants.

We can change the color by targeting the element with a selector, such as this span.

Propietate batzuk ez dira heredatzen. Adibidez, elementu baten zabalera, *margin*, *padding* edo *border* bat ezartzen bada, ondorengoek ez dituzte gurasoengan definitutako propietate hauek ezarrita izango, horrela balitz oso frustragarria izango litzateke CSSa erabiltzea.

## Kaskada

CSSn estilo-orrialde baten arauen ordenak axola duela esan nahi du kaskadak.

CSS adierazpenak aplikatzeko ordena kontrolatzen duten hiru kontzeptu nagusi daude:

- Garrantzia
- Espezifikotasuna
- Arauen ordena

### Garrantzia

CSS adierazpen baten garrantzia non zehaztu den arabera da.

- Lehenetsuna 3: nabigatzaileen estilo lehenetsiak.

Nabigatzaileek bere arauak dituzte HTML elementu ezberdinak bistaratzeko moduari buruz orriaren diseinatzaileek estilorik zehazten ez dituztenean: adibidez, bisitatu gabeko estekak urdinak eta azpimarratuak izaten dira.

- Lehenetsuna 2: kanpoko eta barneko estilo-orriak.

Orrialdearen autoreak kanpoko edo barneko estiloak zehazten dituenean, aurrekoak baino garrantzi handiagoa izango dituzte: adibidez, diseinatzaileak esteka elementuei beste kolorea aplikatuz gero nabigatzaileak adieraztutakoak gainidazten ditu.

- Lehenetsuna 1: lerroko estiloak.

Nabigatzaile batek HTML kodea aztertzen duenean goikoak aztertzen ditu definitutako estiloak aplikatuz, hala ere, barne-estilo moduan style atributua definituta duen elementura iristen denean, aurrez definitutako edozein estilo gainidatziko du.

Nabigatzaileen estilo lehenetsiak.

Kanpoko eta barneko estilo-orriak.

Lerroko estiloak.

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <p>
      <a href="#">
        I am an hyperlink
      </a>
    </p>
    <p>
      <a href="#">
        I am another hyperlink
      </a>
    </p>
  </body>
</html>
```

[I am an hyperlink](#)

[I am another hyperlink](#)

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      a {
        text-decoration: none;
        background: salmon;
        color: red;
      }
    </style>
  </head>
  <body>
    <p>
      <a href="#">
        I am an hyperlink
      </a>
    </p>
    <p>
      <a href="#">
        I am another hyperlink
      </a>
    </p>
  </body>
</html>
```

[I am an hyperlink](#)

[I am another hyperlink](#)

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      a {
        text-decoration: none;
        background: salmon;
        color: red;
      }
    </style>
  </head>
  <body>
    <p>
      <a href="#">
        I am an hyperlink
      </a>
    </p>
    <p>
      <a href="#"
        style="background: pink;">
        I am another hyperlink
      </a>
    </p>
  </body>
</html>
```

[I am an hyperlink](#)

[I am another hyperlink](#)

### Espezifikotasuna

Orrialde bereko arau batzuek hautatzaile desberdinak badituzte, baina elementu berari aplika dakizkioke, nabigatzaileak zein arau aplikatzen den erabakitze duen modua espezifikotasuna da.

Funtsean, **espezifikotasunak** hautatzaile baten zehaztasuna zein espezifikoa den neurtzen du :

- Lehenetsua 3: Elementu mota hautatzaileak.
- Lehenetsua 2: Klase hautatzaileak.
- Lehenetsua 1: ID hautatzaileak.

bi arauk espezifikotasun bera dutenean, CSSn azkena agertzen dena aplikatzen da.

#### CSS espezifikazioaren propietateak

Behoko adibidean esteka elementuei aplika dakizkiokeen bi arau ditugu.

Estekak urdin kolorezkoak izaten amaitzen dute arau biek hautatzaile berdina dutelako eta, beraz, espezifikotasun bera dutelako.

Hau dela eta agertzen den azkena aplikatzen da.

Beste behin esteka elementuei aplika dakizkiokeen bi arau aurkitzen dira.

Oraingoan estekak gorriak izaten amaitzen dira: klase-hautatzaileak lehenengo arauari ematen dio espezifikotasun handiagoa, beraz, hau aplikatuko da elementu-hautatzailearen araua kontuan hartu gabe nahiz eta kodearen ordenan azpian zerrendatzen bada ere.

Azken eredu honetan ID hautatzaile bat gehitu da CSS auaruen artean.

Oraingo honetan, lehenengo estekarentzako, klase-hautatzailearen kolorearen definizioa ez da kontuan hartuko, elementu horretarako beste id-hautatzaile zehatzagoa bait dago beste kolore bateko adierazpenarekin.

Klase-hautatzailean adierazitako beste propietate guztiak bai hedatuko dituela.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      a {
        color: red;
      }
      a {
        color: blue;
      }
    </style>
  </head>
  <body>
    <p>
      <a href="#"
        id="link-1"
        class="my-link">
        I am an hyperlink
      </a>
    </p>
    <p>
      <a href="#"
        id="link-2"
        class="my-link">
        I am another hyperlink
      </a>
    </p>
  </body>
</html>
```

[I am an hyperlink](#)

[I am another hyperlink](#)

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .my-link {
        color: red;
      }
      a {
        color: blue;
      }
    </style>
  </head>
  <body>
    <p>
      <a href="#"
        id="link-1"
        class="my-link">
        I am an hyperlink
      </a>
    </p>
    <p>
      <a href="#"
        id="link-2"
        class="my-link">
        I am another hyperlink
      </a>
    </p>
  </body>
</html>
```

[I am an hyperlink](#)

[I am another hyperlink](#)

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      #link-1 {
        color: green;
      }
      .my-link {
        color: red;
      }
      a {
        color: blue;
      }
    </style>
  </head>
  <body>
    <p>
      <a href="#"
        id="link-1"
        class="my-link">
        I am an hyperlink
      </a>
    </p>
    <p>
      <a href="#"
        id="link-2"
        class="my-link">
        I am another hyperlink
      </a>
    </p>
  </body>
</html>
```

[I am an hyperlink](#)

[I am another hyperlink](#)

## Elementuaren kokapenaren arabera formatua

Batzuetan zerbait dokumentuan dagoen tokiaren arabera desberdina izatea nahiko da.

Horretarako, CSS hautagailu bat hautatzaile sinple bat baino gehiagoko konbinazioa izan daiteke.

Hautatzaile sinpleen artean konbinatzaile desberdinak sar daitezke (testu-espazioa, >, +, ~, ~).

### Espazioa (ondorengo hautagailua)

Espazioa da CSSetan gehien erabiltzen den hautagailua. Beste bi hautatzaile sinpleen arteko espazio baten forma hartzen du, besterik gabe.

Beste elementu batzuen barruan habiatuta dauden elementuak aukeratzeko erabiltzen da.

```
parentTag childTag { ... }
```

Hurrengo adibidean, `<span>` elementu guztiek letra-kolore bat hartzen dute, baina, horretaz gain, `</>` elementu baten barruan habiatuta dauden `<span>` elementuek atzeko-kolore bat ere hartzen dute.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      span {
        color: red;
      }
      li span {
        background: salmon;
      }
    </style>
  </head>
  <body>
    <ul>
      <li>
        <span>
          I am a span
        </span>
      </li>
    </ul>
  </body>
</html>
```

- This **span** is inside an hyperlink.

```

</style>
</head>
<body>
  <ul>
    <li><a href="">This <span>span</span> is inside an hyperlink.</a></li>
    <li>This <span>span</span> is inside a list item.</li>
  </ul>
  <p>This <span>span</span> is inside a paragraph.</p>
  <span>This is only a span</span>
  <span>This is another span</span>
</body>
</html>

```

- This **span** is inside a list item.

This **span** is inside a paragraph.

This is only a span This is another span

## > Sinboloa (Zuzeneko seme-alaben hautatzailea)

Konbinagailu hau beste bi hautagailu sinpleen artean kokatzen da CSS arauak elementu jakin baten seme-alaba zuzenak diren elementuei soilik aplikatzeko.

```
parentTag > childTag { ... }
```

Formatua aplikatzeko, elementuak aurretik duen beste batetik egon behar du zuzenean, hierarkian inolako bitarteko elementurik gabe.

Aurreko adibidearekin jarraituz, ikus daiteke nola CSS arau berria bakarrik aplikatzen zaiola zuzenean `<li>` elementuaren atzetik dagoen `<span>` elementuari.

Zerrendako beste `<span>` elementuak `<li>` eta bere artean esteka bat du, beraz, ez du arau berria betetzen eta ez zaio formatua aplikatzen.

```

<!DOCTYPE html>
<html>
  <head>
    <style>
      span {
        color: red;
      }
      li span {
        background: salmon;
      }
      li > span {
        background: lightgreen;
      }
    </style>
  </head>
  <body>
    <ul>
      <li><a href="">This <span>span</span> is inside an hyperlink.</a></li>
      <li>This <span>span</span> is inside a list item.</li>
    </ul>
    <p>This <span>span</span> is inside a paragraph.</p>
    <span>This is only a span</span>
    <span>This is another span</span>
  </body>
</html>

```

- This **span** is inside an hyperlink.

- This **span** is inside a list item.

This **span** is inside a paragraph.

This is only a span This is another span

## + Sinboloa (alboko anai-arreben hautagailua)

Konbinagailu hau beste bi hautagailu sinpleen artean kokatzen da, elementu zehatz baten alboko anai-arrebak diren elementu guztiei CSS arauak aplikatzeko.

```
parentTag + childTag { ... }
```

Elementu anaiak elementu nagusi bera izan behar dute. "Alboko" hitzak "berehala" esan nahi du.

Hurrengo adibidean, `<p>` elementuaren ondoren datorren `<span>` elementuari soilik aplikatuko zaio araua.

```

<!DOCTYPE html>
<html>
  <head>
    <style>
      span {
        color: red;
      }
      p + span {
        background: salmon;
      }
    </style>
  </head>
  <body>
    <ul>
      <li><a href="">This <span>span</span> is inside an hyperlink.</a></li>
      <li>This <span>span</span> is inside a list item.</li>
    </ul>
    <p>This <span>span</span> is inside a paragraph.</p>
    <span>This is only a span</span>
    <span>This is another span</span>
  </body>
</html>

```

- This **span** is inside an hyperlink.

- This **span** is inside a list item.

This **span** is inside a paragraph.

This is only a span This is another span

## ~ Sinboloa (alboko anai-arreben hautagailu orokorra)

Konbinagailu hau beste bi hautatzaile sinpleen artean kokatzen da eta aurrekoaren antzekoa da, baina, kasu honetan elementu espezifiko baten anai-arrebak diren hurrengo elementu guztiak hautatzen ditu.

```
parentTag ~ childTag { ... }
```

En el siguiente ejemplo, al contrario del anterior, se seleccionan todos los siguientes elementos `<span>` que son hermanos de los elementos `<p>`.

Hurrengo adibidean, aurrekoan ez bezala, `<p>` elementuen anai-arrebak diren `<span>` elementu guztiak hautatzen dira.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      span {
        color: red;
      }
      p ~ span {
        background: salmon;
      }
    </style>
  </head>
  <body>
    <ul>
      <li><a href="">This <span>span</span> is inside an hyperlink.</a></li>
      <li>This <span>span</span> is inside a list item.</li>
    </ul>
    <p>This <span>span</span> is inside a paragraph.</p>
    <span>This is only a span</span>
    <span>This is another span</span>
  </body>
</html>
```

- This span is inside an hyperlink.
- This span is inside a list item.

This span is inside a paragraph.

This is only a span This is another span

## Hautagailuen eta konbinatzaileen konbinazioak

Merezi du aipatzea hautatzaileen eta konbinatzaileen konbinazio ugari egin daitezkeela. Adibidez:

```
/* edozein <span> elementu hautatzen du <p> baten barruan dagoenean, eta azkenengo hau aldi berean <div> baten barruan badago */
div p span { ... }

/* edoizen <p> elementu hautatzen du zuzenean <ul> baten ondoren dagoenean, eta azkenengo hau aldi berean <h1> baten ondoren badago */
h1 + ul + p { ... }

/* custom-element id izenarekin identifikatzen den elementuaren barruan dauden esteka guztiak hautatzen du */
#custom-element a { ... }
```

## Iruzkinak

HTMLrekin gertatzen den bezala, komeni da CSS-ri buruzko iruzkinak egitea, denboran zehar kodea berrerrabiltzen denean nola funtzionatzen duen ulertzen laguntzeko, baita horrekin lan egingo duten besteei ulertzen laguntzeko ere.

CSS-ko iruzkinak `/*` ikurrarekin hasten dira eta `*/` ikurrarekin amaitzen dira.

```
p {
  /* Parrafo guztiei 16 pixeleko tamaina ezarri */
  font-size: 16px;
}
p.warning {
  /* warning klasedun parrafo guztiei kolore horia ezarri */
  color: red;
}
p.danger {
  /* danger klasedun parrafo guztiei kolore gorria ezarri */
  color: red;
}
```