

4. INTRODUCCIÓ A CSS *posicionament*

EXERCICIS

19. HTML diferencia dos tipus d'elements bàsics:

- Elements de bloc, que comencen en una nova línia i ocupen tot l'espai disponible fins al final de la línia. Els principals elements de bloc són: *blockquote, center, div, dl, form, h1, h2, h3, h4, h5, h6, hr, ol, p*, pre, table, ul*
- Elements de línia, que només ocupen l'espai necessari per a mostrar els seus continguts. Els principals elements de línia són: *a, em, font, i, img, input, label, select, span, strong, sub, sup, textarea, u*.
- Els elements de bloc només poden aparèixer dins d'altres elements de bloc, en canvi els elements de línia poden aparèixer dins d'elements de bloc i de línia. Per comprovar tot això, dissenya una pàgina com aquesta:

Parrafo 1: Los parrafos son elementos de bloque

Un enlace puede aparecer al lado de otro enlace

Parrafo 2: Un parrafo puede contener un enlace dentro del parrafo

Por defecto,

un Header 3

no puede aparecer al lado de

otro Header 3

20. Mitjançant CSS es poden establir 4 maneres bàsiques de posicionar els elements a la pàgina. La primera és el posicionament estàtic, que és el model que utilitzen per defecte els navegadors (com es va veure en l'exercici anterior), i on la distància entre els elements de bloc ve determinada pels marges verticals, i la distància entre els elements de línia pels marges laterals. Redefineix els estils de l'exercici anterior, afegint marges de 10 píxels per a tots els elements, i confirma que queda de la següent manera:

Parrafo 1: Los parrafos son elementos de bloque

Un enlace puede aparecer al lado de otro enlace

Parrafo 2: Un parrafo puede contener un enlace dentro del parrafo

Por defecto,

un Header 3

no puede aparecer al lado de

otro Header 3

21. El posicionament relatiu consisteix en posicionar una caixa segons el posicionament estàtic o normal, i després desplaçar-la respecte de la seva posició original. El desplaçament de la caixa es controla amb les propietats *top*, *right*, *bottom* i *left*. Per comprovar-ho, dissenya una pàgina com la de l'exemple, on inicialment es mostra els tres enllaços i el paràgraf mitjançant posicionament estàtic (per defecte), i a continuació, es mostra com l'enllaç 2 s'ha posicionat de manera relativa, col·locant-lo a 15 píxels de seva posició superior (*top*) original, i a 45 píxels de la seva posició esquerra (*left*) original:

Enlace 1 Enlace 2 Enlace 3

Parrafo 1: Así quedaria mediante posicionamiento estático

Enlace 1 Enlace 2 Enlace 3

Parrafo 1: El enlace 2 tiene un posicionamiento relativo

22. També és important tenir en compte que el posicionament relatiu no influeix sobre la resta d'elements, així per exemple, si per a l'exercici anterior es decideix posicionar de manera relativa l'enllaç 1, col·locant-75 píxels per sota de la seva posició original, això no influeix sobre la resta d'elements (enllaços i paràgrafs), que segueixen ocupant la seva posició normal o estàtica (prova-ho):

Enlace 2 Enlace 3

Parrafo 1: Así quedaria mediante posicionamiento estático

Enlace 1

I amb aquest posicionament ja pots dissenyar la següent pàgina, on apareix un *Header* al costat d'un altre, encara que els dos siguin elements de bloc:

Un Header 3 Otro Header 3

23. Mitjançant el posicionament absolut, la posició d'un element s'estableix de forma absoluta respecte del seu element contenidor (el que el conté), i la resta d'elements de la pàgina ignoren la seva posició, ocupant el seu lloc. Per provar-ho, canvia el posicionament de l'enllaç 1 de l'exercici anterior, de relatiu a absolut, sense canviar cap altra propietat, i comprova que s'obté el següent resultat, on els enllaços 2 i 3 han ocupat l'espai deixat per l'enllaç 1:

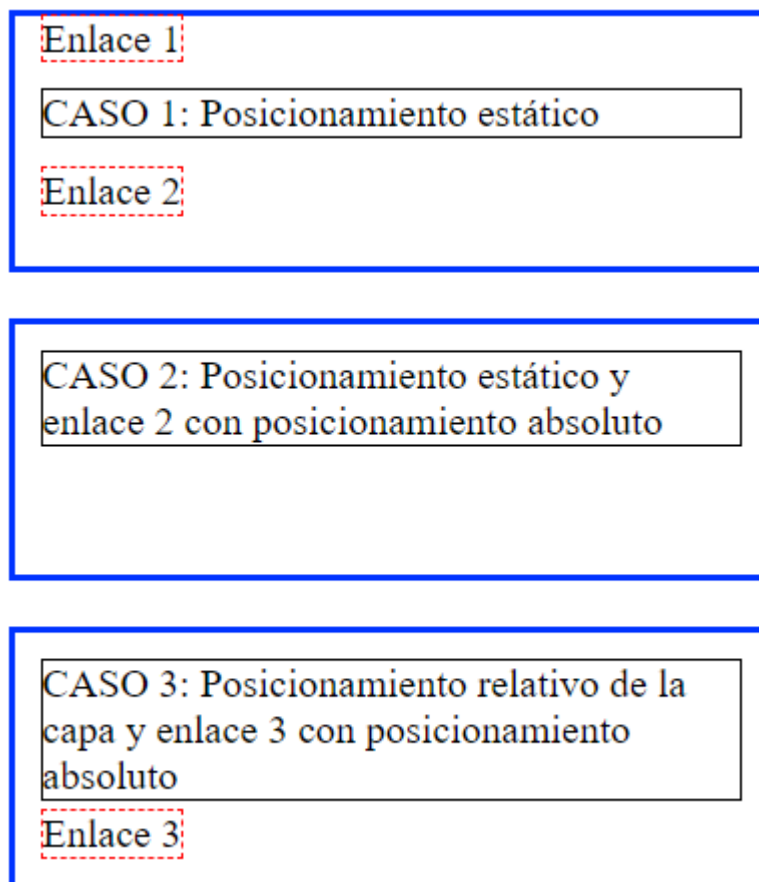
Enlace 2 Enlace 3

Parrafo 1: Así quedaria mediante posicionamiento estático

Enlace 1

24. En el posicionament absolut, per establir quin és l'origen de coordenades d'un element, el navegador es queda amb el primer element contenidor començant pel més proper, tal que estigui posicionat de qualsevol forma diferent a la estàtica. Anem a comprovar-ho mitjançant un exemple, on primer es mostra una capa (sense indicar res respecte al seu posicionament, per tant amb posicionament estàtic), que conté un enllaç 1 i tot seguit un paràgraf. Posteriorment, es torna a mostrar la capa, però aquesta vegada l'enllaç

2 es posiciona de manera absoluta amb propietats *top = 60px*. Què succeeix? Finalment, a la tercera capa, canvia les seves propietats, afegint posicionament relatiu, sense indicar cap valor. I afegint dins aquesta tercera capa, també un enllaç 3 posicionat de manera absoluta amb propietats *top = 60px*. Què succeeix? Comprova que la pàgina queda així pels 3 casos:



CONCLUSIÓ: Si es vol posicionar un element de forma absoluta respecte del seu element contenidor, només cal posicionar l'element contenidor de manera relativa, fins i tot sense necessitat de desplaçar-ho de la seva posició estàtica o normal.

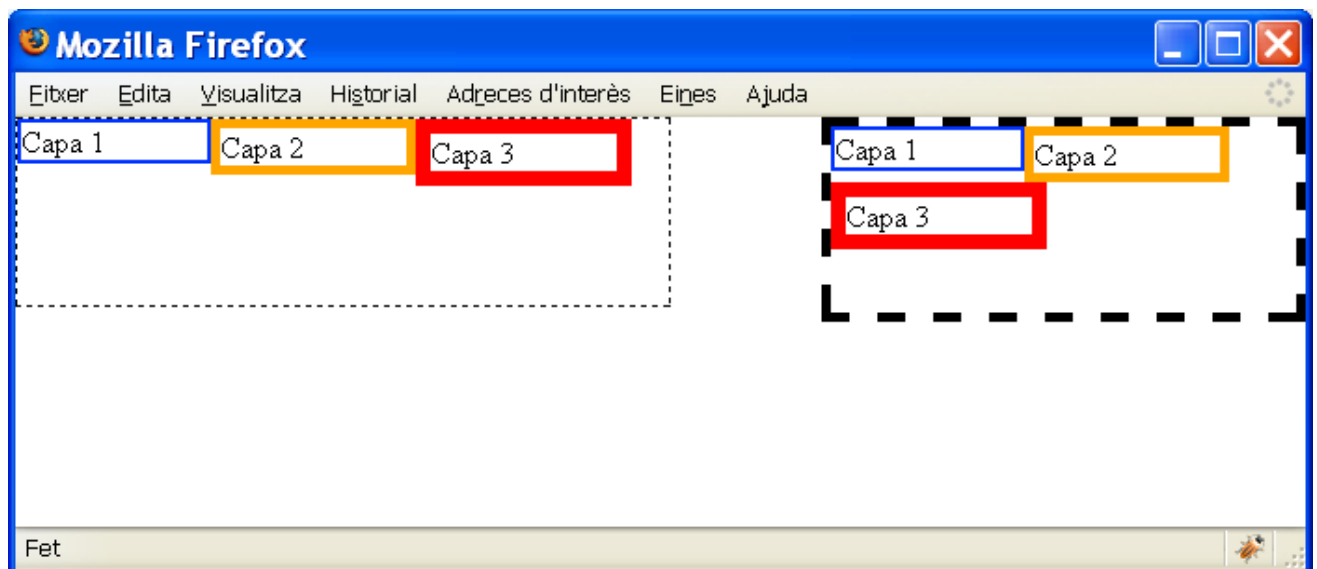
25. L'últim mètode per posicionar els elements, denominat posicionament flotant, consisteix a desplaçar un element fins a la zona més a l'esquerra o més a la dreta possible, respecte a la seva posició original. Els elements en línia fan lloc als elements flotants, i els elements de bloc adapten el seu contingut. Per entendre això, vegem el següent exemple, en el qual apareixen 3 capes amb vores de 2,5 i 8 píxels respectivament i colors blau, taronja i vermell. Totes les capes tenen un ample de 100 píxels. En el primer cas apareixen totes amb posicionament normal o estàtic, en el segon cas la capa 1 flota cap a la dreta, i en el tercer cas la capa 1 flota cap a l'esquerra (la capa 2 adapta el seu contingut). Prova-ho:



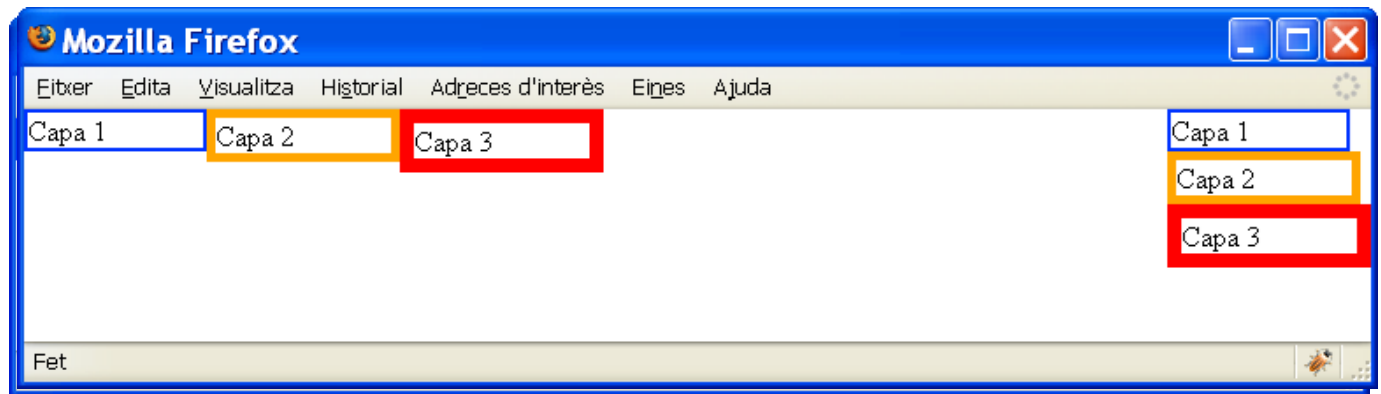
26. En el posicionament flotant, al posicionar un element, si no hi ha lloc a la línia actual, l'element baixa a la línia inferior, fins que trobi el lloc necessari per mostrar-se el més a la dreta o a l'esquerra possible en la nova línia. Per veure això, dissenya una pàgina com la que es mostra a continuació, on les 3 capes de l'exercici anterior, es posicionen de manera flotant cap a l'esquerra, i apareixen primer dins d'una capa de 350px d'ample per 100 d'alt, i després, dins d'una capa de 250px d'ample per 100 d'alt, on s'aprecia que com no hi caben les 3 capes en una mateixa línia, la tercera capa baixa a la següent línia:



27. Modifica l'exercici anterior, perquè ara apareguin en una pàgina web (sense marges), una capa a l'esquerra i l'altra a la dreta, com es mostra a continuació:



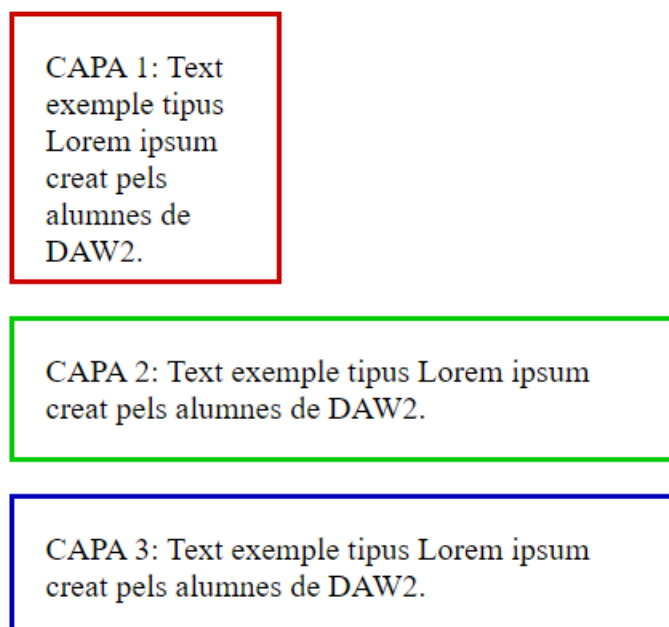
28. Torna a modificar l'exercici anterior per obtenir ara el següent disseny:



29. Quan s'utilitza el posicionament flotant, de vegades interessa fer net, per a això hi ha la propietat *clear* (esborrar), que permet modificar el comportament per defecte, per forçar un element a mostrar-se per sota de qualsevol element flotant. La propietat admet 3 valors:

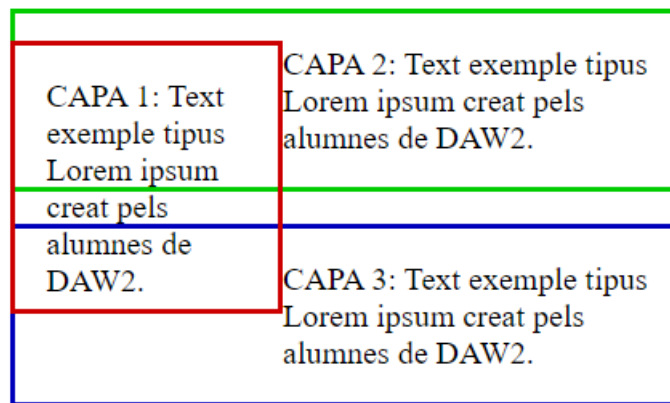
- *clear: left* (esborrar esquerra): L'element es desplaça de forma descendent fins que pugui col·locar-se en una línia en la qual no hi hagi cap element flotant a la banda esquerra.
 - *clear: right* (esborrar dreta): El mateix però per a elements flotants al costat dret.
 - *clear: both* (esborrar tots dos): Es col·loca en una línia en la qual no hi hagi elements flotants ni a la banda esquerra ni a dreta.
- Per posar a prova això, dissenya una pàgina amb 3 capes amb posicionament per defecte:

Posicionament NORMAL de les 3 capes



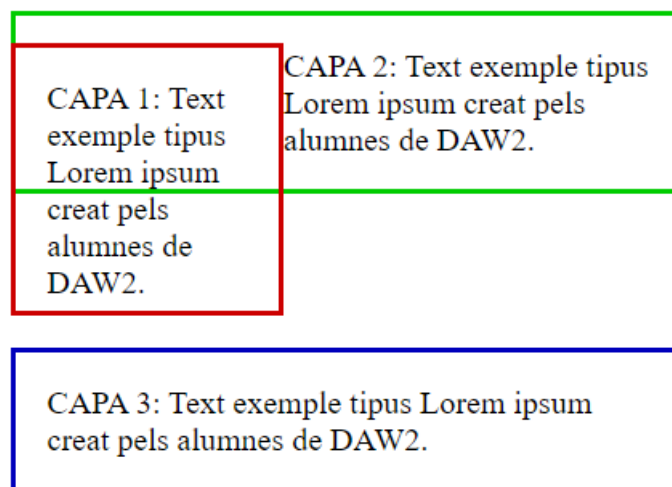
- A continuació, posiciona la CAPA 1 per tal que floti a l'esquerra:

Posicionament CAPA 1: float left



- Finalment, utilitza la propietat *clear:left* sobre la CAPA 3, per tal que aparegui a sota de la CAPA 1:

Posicionamiento CAPA 1: float left i CAPA 3:clear left



30. Un altre exemple interessant de posicionament flotant en què cal fer servir la propietat *clear*, és per dissenyar una estructura de paginació, que permet anar endavant i enrere .Per fer-ho, dissenya una pàgina que inclogui una capa contenidora de color de fons gris, que no especifica ni ample ni alt, aquesta capa conté dos elements posicionats a la banda esquerra i dreta respectivament (es pot utilitzar l'etiqueta ** per aplicar l'estil). Com veus, el text dels *span* sobresurt de la capa (sense *width* ni *height*) :

Problema de paginacion: div sin contenido, no cabe el texto

« Anterior

Siguiente »

- Hi ha dues possibles solucions: la primera consisteix a afegir un espai en blanc a la fi de la capa contenidora, d'aquesta manera, el contingut de la capa adapta la seva alçada per fer lloc als 2 elements flotants.

Problema de paginacion: SOLUCION 1: insercion de un caracter vacio al div

« Anterior

Siguiente »

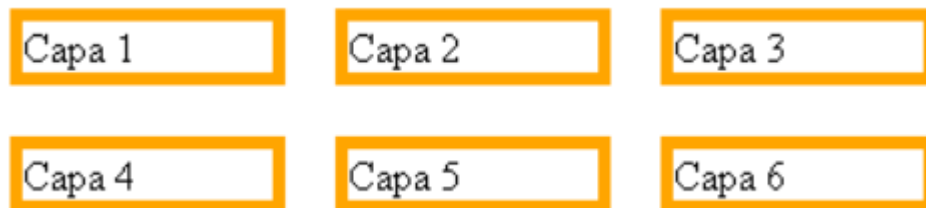
- La segona solució és més elegant, i consisteix en afegir al final de la capa, en comptes d'un espai en blanc, una altra capa amb la propietat *clear: both*, sense cap contingut, que obté el mateix resultat.

Problema de paginacion: SOLUCION 2: insercion de un div vació con clear both

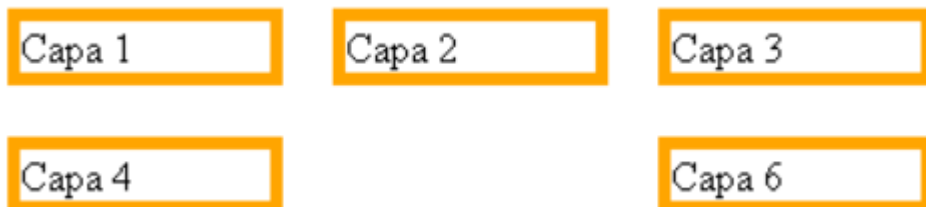
« Anterior

Siguiente »

31. La propietat CSS visibilitat (*visibility*), si és oculta, no mostra els continguts de l'element, tot i que si reserva l'espai ocupat per aquest element. Per exemple, dissenya la següent pàgina amb les 6 capes posicionades de la següent manera:

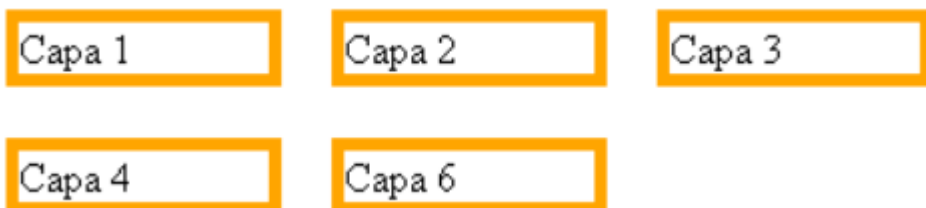


A continuació oculta amb la propietat anterior la capa 5, i comprova que s'obté el següent:

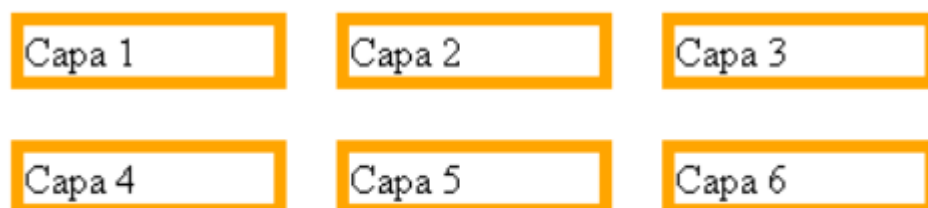


Hi ha una altra propietat CSS (*display*), on els principals valors utilitzats són 3:

- **display:none** Oculta l'element fent que desaparegui de la pàgina. La resta d'elements de la pàgina ocupen el seu espai. Per l'exemple anterior, s'oculta la capa 5:

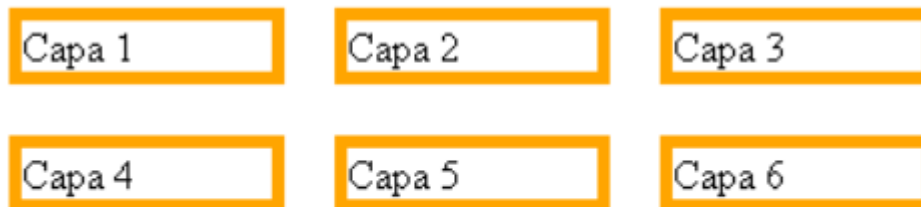


- **display:block** Mostra l'element com si fos de bloc, sense importar el tipus d'element. Per exemple, per a la pàgina inicial, afegeix 2 enllaços, que es mostrin un a costat de l'altre de la següent manera:



[Enlace 1](#) [Enlace 2](#)

A continuació, s'afegeix a l'enllaç 1 una classe CSS que ho mostra com si es tractés d'un element de bloc, i el disseny de la pàgina ara hauria de ser com aquest:



[Enlace 1](#)
[Enlace 2](#)

- **display:inline** Mostra l'element de línia, encara que sigui de tipus bloc. Per provar aquesta propietat, dissenya la següent pàgina en la qual apareix una nova capa (la 7) com si fos un element de línia, al costat d'un enllaç:



32. Normalment els continguts d'un element es poden mostrar en l'espai reservat per a aquest element. Però en ocasions el contingut d'un element no cap en l'espai reservat per a aquest element i es desborda. La propietat *overflow* (desbordament) permet controlar aquestes situacions, i admet 4 valors possibles:

- **overflow:visible** El contingut es mostra sobresortint de la zona reservada (és el comportament per defecte).
- **overflow:hidden** El contingut que sobresurt no es mostra.
- **overflow:scroll** Es mostren barres de desplaçament per a mostrar el contingut sobrant.
- **overflow:auto** Depèn del navegador, normalment coincideix amb el valor anterior *scroll*.

Dissenya una pàgina com la que es mostra a continuació, en la qual apareixen 4 capes amb un contingut sobrant, i sobre cadascuna de les 4 capes s'aplica un estil de desbordament:



33. [opcional] Aprofitant que ja sabem posicionar elements, en aquest exercici es pretén explicar les següents propietats pel que fa als marges:

- Quan coincideixen 2 o més marges verticals, es fusionen de forma automàtica i l'altura de el nou marge serà igual a l'altura del marge més alt dels que s'han fusionat.
- Si un element està contingut dins d'un altre element, els seus marges verticals també es fusionen i resulten en un nou marge amb l'altura del major marge dels que s'han fusionat (encara que en aquest cas es pot evitar afegint un petit farciment o vora a l' element contenidor).

Per entendre això, dissenya una pàgina web com la que es mostra en el gràfic de la següent pàgina i que també està disponible al curs Moodle, carpeta: material/CSS_33.png.

