

## UD3.3. CSS AMB FLEX

### Tabla de contenido

<b>UD3.3. CSS AMB FLEX .....</b>	<b>1</b>
<b>1. CONCEPTES BÀSICS DE DISPOSICIÓ FLEX.....</b>	<b>2</b>
1.1. Per què FLEX? .....	2
1.2. Presentació d'un Exemple Senzill.....	2
1.3. Especificar quins elements col·locar com a caixes flexibles .....	3
1.4. El Model Flexible .....	4
1.5. Columnes o files?.....	5
1.6. Delimitar .....	5
1.7. Propietats Abreujades de Flex-flow .....	7
1.8. Dimensionament flexible d'elements flexibles .....	7
<b>2. ALINEACIÓ HORITZONTAL I VERTICAL.....</b>	<b>9</b>
<b>3. ORDENAR ELS ELEMENTS FLEXIBLES.....</b>	<b>11</b>
3.1. Caixes Flexibles Niades .....	12

## 1. CONCEPTES BÀSICS DE DISPOSICIÓ FLEX

### 1.1. Per què FLEX?

Durant molt de temps, les úniques eines fiables amb compatibilitat creuada entre navegadors disponibles per a crear dissenys CSS van ser coses com la flotació i el posicionament. Estan bé i funcionen, però d'alguna manera també limiten bastant i frustren.

Amb tals eines resulta difícil, si no impossible, aconseguir obtenir en qualsevol forma convenient i flexible un disseny de pàgina senzill amb uns requisits com els següents:

- Centrar verticalment un bloc de contingut dins del seu element pare.
- Fer que tots els elements secundaris d'un contenidor ocupen una quantitat igual de l'ample/alt disponible, independentment de l'ample/alt que hi haja disponible.
- Fer que totes les columnes en una compaginació en columnes múltiples adopten la mateixa altura fins i tot si contenen quantitats diferents de contingut.

Com veuràs en les seccions següents, els elements flexbox faciliten molt algunes tasques de compaginació. Aprofundirem en això!

### 1.2. Presentació d'un Exemple Senzill

Te guiaré per una sèrie d'exercicis per a ajudar-te a comprendre com funcionen els elements flexbox. Per a començar, has de fer una còpia local de la carpeta del AULES de codi per aquesta teoria. El primer arxiu que emprarem serà el contingut de la carpeta "PRIMER". Carrega-ho en un navegador modern (com Firefox o Chrome) i dona una ullada al codi en el teu editor de codi.

Veuràs que hi ha un element `<header>` amb un encapçalat de nivell superior en ell, i un element `<section>` que conté tres elements `<article>`. Els usarem per a crear una compaginació bastant habitual de tres columnes.

**Sample flexbox example**

**First article**  
Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

**Second article**  
Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

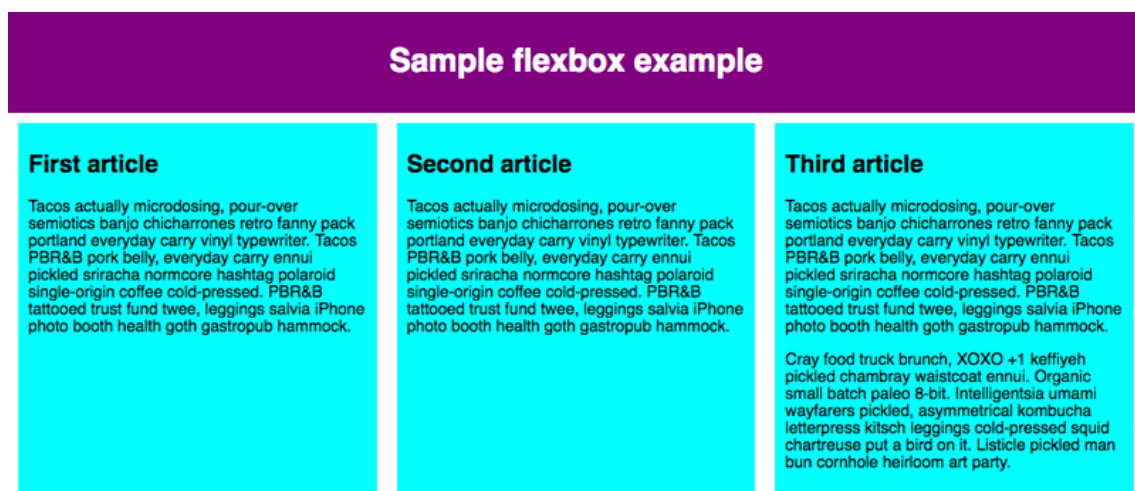
**Third article**  
Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.  
Cray food truck brunch, XOXO +1 keffiyeh pickled chambray waistcoat ennui. Organic small batch paleo 8-bit. Intelligentsia umami wayfarers pickled, asymmetrical kombucha letterpress kitsch leggings cold-pressed squid charreuse put a bird on it. Listicle pickled man bun cornhole heirloom art party.

### 1.3. Especificar quins elements col·locar com a caixes flexibles

Per a començar, seleccionarem quins elements es presentaran com a caixes flexibles. Per a això, establim un valor especial de `display` en l'element pare dels elements que desitges editar. En este cas, volem compaginar els elements `<article>`, per la qual cosa l'establim en `<section>` (que es converteix en un contenidor flexible):

```
section {  
    display: flex;  
}
```

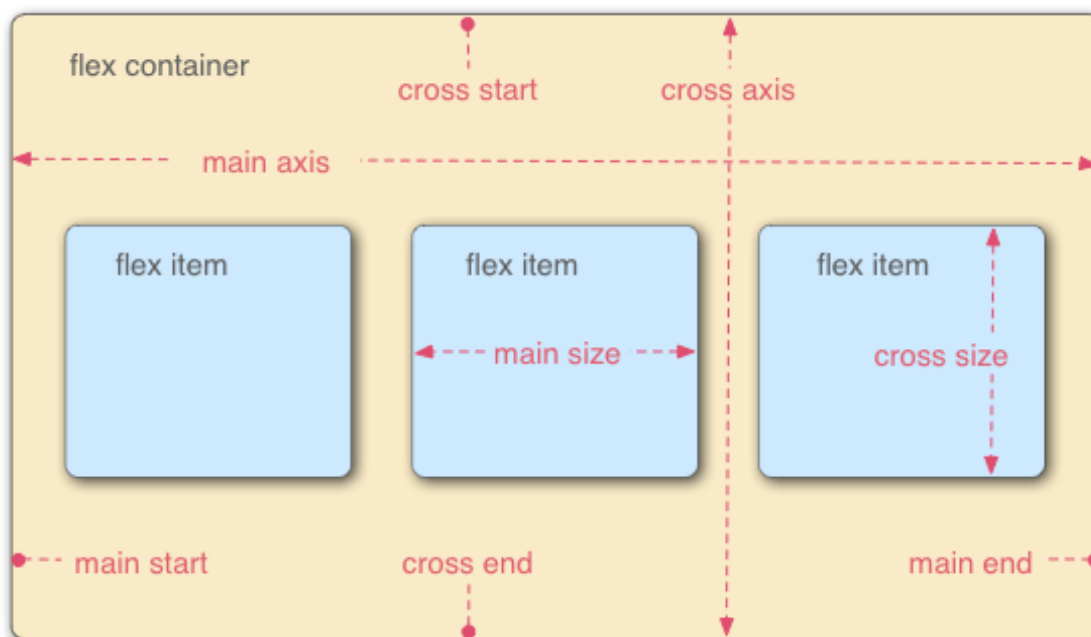
Això fa que l'element `<section>` es convertisca en **contenidor flex**, i els seus fills en **elements flexibles**. El resultat d'això hauria de ser una cosa així:



Així, esta declaració única ens dona tot el que necessitem. Increïble, veritat? Tenim la nostra compaginació en columnes múltiples amb columnes d'igual grandària, i totes les columnes tenen la mateixa altura. Això es deu al fet que els valors per defecte que s'han assignat als elements flexibles (els elements secundaris del contenidor flexible) estan pensats per a resoldre problemes comuns com este. Veurem més sobre el tema més endavant.

Perquè quede clar, reiterem el que està succeint ací. L'element al qual li hem donat un valor de `display` de `flex` actua com un element a nivell de bloc en termes de com interactua amb la resta de la pàgina, però els seus elements secundaris es presenten com a elements flexibles. La següent secció explicarà amb més detall què significa això. Tingues en compte també que pot usar un valor de `display` d'`inline-flex` si desitja dissenyar els elements secundaris d'un element com a elements flexibles, però fer que eixe element es comporte com un element en línia.

## 1.4.El Model Flexible



Quan els elements es presenten com a caixes flexibles, es distribueixen respecte a dos eixos:

- L'eix **principal (main axis)** és l'eix que corre en la direcció en què es col·loquen els elements flexibles (per exemple, segons es disposen les files en una pàgina o cap avall segons es disposen les columnes en una pàgina). L'inici i el final d'este eix es denominen **inici principal (main start)** i **final principal (main end)**.
- L'eix **transversal (cross axis)** és l'eix que corre perpendicular a la direcció en la qual es col·loquen els elements flexibles. L'inici i el final d'este eix es denominen **inici transversal (cross start)** i **extrem transversal (cross end)**.
- L'element pare que té establert `display: flex` (l'element [<section>](#) en el nostre exemple) es diu **contenidor flexible**.
- Els elements que es presenten com a caixes flexibles dins del contenidor flexible es denominen **elements flexibles** (són els elements [<article>](#) del nostre exemple).

Tingues present esta terminologia en avançar per les seccions posteriors. Si en algun moment et confons amb l'ús d'estos conceptes, sempre pots tornar arrere a consultar-los.

## 1.5. Columnes o files?

Els elements flexbox proporcionen una propietat anomenada [flex-direction](#) que especifica en quina adreça corre l'eix principal (en quina adreça estan disposats els elements fill d'un element flexbox); per defecte, està establert en el valor `row`, per la qual cosa es presenta en una fila en l'adreça en què s'escriu l'idioma predeterminat del teu navegador (d'esquerra a dreta, en el cas d'un navegador en espanyol).

Afig la declaració següent a la teua regla [<section>](#):

```
flex-direction: column;
```

Observa que això torna a col·locar els elements en una disposició en columna, igual que abans d'afegir qualsevol CSS. Abans de continuar, elimina esta declaració del teu exemple.

## 1.6. Delimitar

Un problema que sorgeix quan tens una quantitat fixa d'ample o alt en el teu disseny és que els fills d'un element flexbox eventualment desborden el contenidor i trenquen el disseny. Pots crear molts articles com en el exemple següent:

Sample flexbox example							
First article	Second article	Third article	Fourth article	Fifth article	Sixth article	Seventh article	Eighth article
Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth	Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth	Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth	Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth	Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth	Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth	Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth	Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth

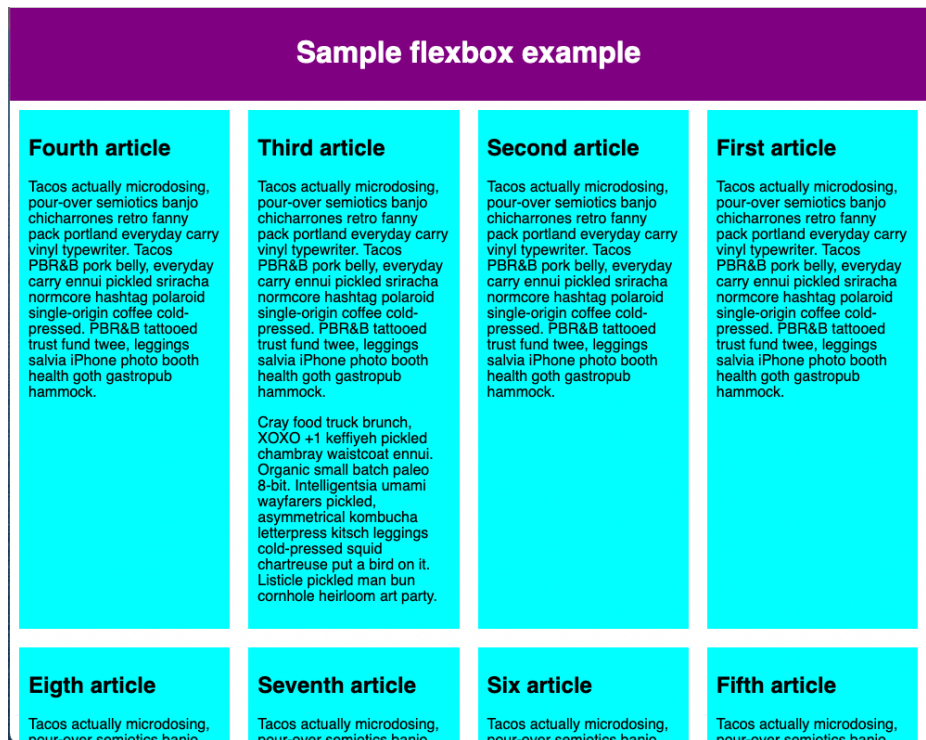
Ací veiem que els elements fill s'ixen del seu contenidor. Una manera de solucionar això és afegir la declaració següent a la teua regla `<section>`:

```
flex-wrap: wrap;
```

Afig també la declaració següent a la teua regla `<article>`:

```
flex: 200px;
```

Prova-ho; observa que en haver-ho inclòs l'aspecte de la compaginació resulta molt més agradable:



Ara hi ha diverses files i en cada fila caben tants elements fill d'un element flexbox com siguin necessaris, i qualsevol desbordament fa saltar l'element cap a la línia següent. La declaració `flex: 200px` que hem establert en els articles significa que cadascun tindrà almenys 200 px d'ample; discutirem esta propietat amb més detall més endavant. Observa també que els últims elements fill de l'última fila s'engrandeixen fins a emplenar tota la fila.

Però ací podem fer molt més. En primer lloc, canvia el valor de la teua propietat `flex-direction` a `row-reverse`; ara veuràs que encara tens la teua compaginació en diverses files, però comença des de la cantonada oposada de la finestra del navegador i flueix a l'inrevés.

## 1.7. Propietats Abreujades de Flex-flow

En este punt val la pena assenyalar que hi ha una propietat abreujada per a [flex-direction](#) i [flex-wrap](#): [flex-flow](#). Així, per exemple, pots reemplaçar:

```
flex-direction: row;  
flex-wrap: wrap;
```

amb

```
flex-flow: row wrap;
```

## 1.8. Dimensionament flexible d'elements flexibles

Tornem ara al nostre primer exemple i vegem com podem controlar quina proporció d'espai ocupen els elements flexibles.

Primer, afegim la regla següent al final de la teua CSS:

```
article {  
  flex: 1;  
}
```

Este és un valor de proporció sense unitats que especifica la quantitat d'espai disponible sobre l'eix principal que ocupa cada element flexible. En este cas, donem a cada element `<article>` un valor de 1, cosa que significa que tots ocuparan una quantitat igual de l'espai lliure restant després que s'hagen establert elements com l'àrea de farciment i el marge. És una proporció, cosa que significa que donar a cada element flexible un valor de 400000 tindria exactament el mateix efecte.

Ara afegim la regla següent davall de l'anterior:

```
article:nth-of-type(3) {  
  flex: 2;  
}
```

En actualitzar veuràs que el tercer `<article>` ocupa ara el doble de l'ample disponible que els altres dos; ara hi ha quatre unitats de proporció disponibles en total. Els primers dos elements flexibles tenen una cadascun, per la qual cosa ocupen 1/4 de l'espai disponible cadascun. El tercer té dues unitats, per la qual cosa ocupa 2/4 de l'espai disponible (o 1/2).



També pots especificar un valor de grandària mínima dins del valor flexible. Actualitza les regles per al teu article de la manera següent:

```
article {  
  flex: 1 200px;  
}  
  
article:nth-of-type(3) {  
  flex: 2 200px;  
}
```

Això estableix bàsicament que «a cada element flexible se li dona primer 200px de l'espai disponible. Després d'això, la resta de l'espai disponible es reparteix d'acord amb les unitats de proporció». Actualitza i observa de quin mode es reparteix ara l'espai.

### Sample flexbox example

#### First article

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

#### Second article

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

#### Third article

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

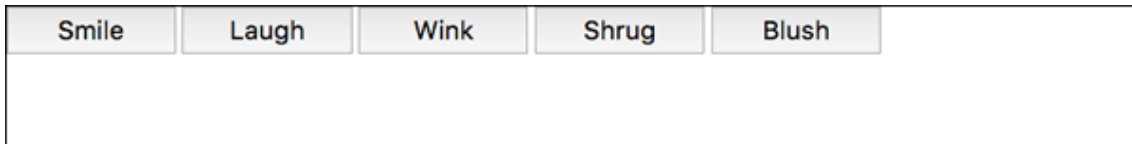
Cray food truck brunch, XOXO +1 keffiyeh pickled chambray waistcoat ennui. Organic small batch paleo 8-bit. Intelligentsia umami wayfarers pickled, asymmetrical kombucha letterpress kitsch leggings cold-pressed squid chartreuse put a bird on it. Listicle pickled man bun cornhole heirloom art party.

El valor real de l'element flexbox es pot veure en la seua flexibilitat/adaptabilitat: si canvies la grandària de la finestra del navegador o afeges un altre element `<article>`, el disseny continua funcionant bé.



## 2. ALINEACIÓ HORITZONTAL I VERTICAL

També pots usar les funcions dels elements flexbox per a alinear elements flexibles sobre l'eix principal o transversal. Explorem este aspecte a partir d'un exemple nou dins de la carpeta "SEGON" que convertirem en una barra d'eines/botons ordenada i flexible. En este moment pots veure una barra de menú horitzontal, amb alguns botons pegats en línia a la cantonada superior esquerra.



Primer, fes una còpia local d'este exemple.

Ara, afeg a la part inferior del CSS de l'exemple el següent:

```
div {  
  display: flex;  
  align-items: center;  
  justify-content: space-around;  
}
```

Actualitza la pàgina i observa que els botons ara estan convenientment centrats, horitzontal i verticalment. Ho hem fet a partir de dues propietats noves.

align-items controla on se situen els elements flexibles en l'eix transversal.

- Per defecte, el valor és `stretch`, que eixampla tots els elements flexibles per a emplenar l'element primari en la direcció de l'eix transversal. Si l'element pare no té un ample fix en la direcció de l'eix transversal, tots els elements flexibles són tan llargs com els elements flexibles més llargs. Així és com el nostre primer exemple va obtenir columnes d'igual altura per defecte.
- El valor `center` que utilitzem en el nostre codi anterior manté les dimensions intrínseques dels elements però els centra sobre l'eix transversal. És per això que els botons del nostre exemple ara estan centrats verticalment.
- També pots tindre valors com `flex-start` i `flex-end`, que alineen tots els elements a l'inici i al final de l'eix transversal, respectivament.

Pots anul·lar el comportament align-items per a elements flexibles individuals aplicant-los la propietat align-self. Per exemple, afeg al teu CSS el següent:

```
button:first-child {  
  align-self: flex-end;  
}
```

Observa quin efecte té això i retira-ho de nou en acabar.

justify-content controla on se situen els elements flexibles sobre l'eix principal.

- El valor per defecte és `flex-start`, que assentisca tots els elements al començament de l'eix principal.
- Pots usar `flex-end` perquè s'assenteixen al final.
- `center` també és un valor de `justify-content` (per a alinear contingut), que assentisca els elements flexibles sobre el centre de l'eix principal.
- El valor `space-around` que hem usat abans és útil perquè distribueix tots els elements de manera uniforme sobre l'eix principal i deixa un poc d'espai en cada extrem.
- Hi ha un altre valor, `space-between`, que és molt similar a `space-around`, però no deixa espai en els extrems.

T'anime a jugar amb estos valors per a veure com funcionen abans de continuar.

### 3. ORDENAR ELS ELEMENTS FLEXIBLES

Els elements flexbox també tenen una funció per a canviar l'ordre de disposició dels elements flexibles, sense que això afecte la seua ordre en el codi font. Això és una cosa que resulta impossible de fer amb els mètodes de compaginació tradicionals.

El codi per a això és simple: afegir el següent CSS al codi d'exemple de la barra de controls:

```
button:first-child {  
  order: 1;  
}
```

Actualitza, i observa que el botó «Smile» s'ha mogut al final de l'eix principal. Observem com funciona això amb un poc més de detall:

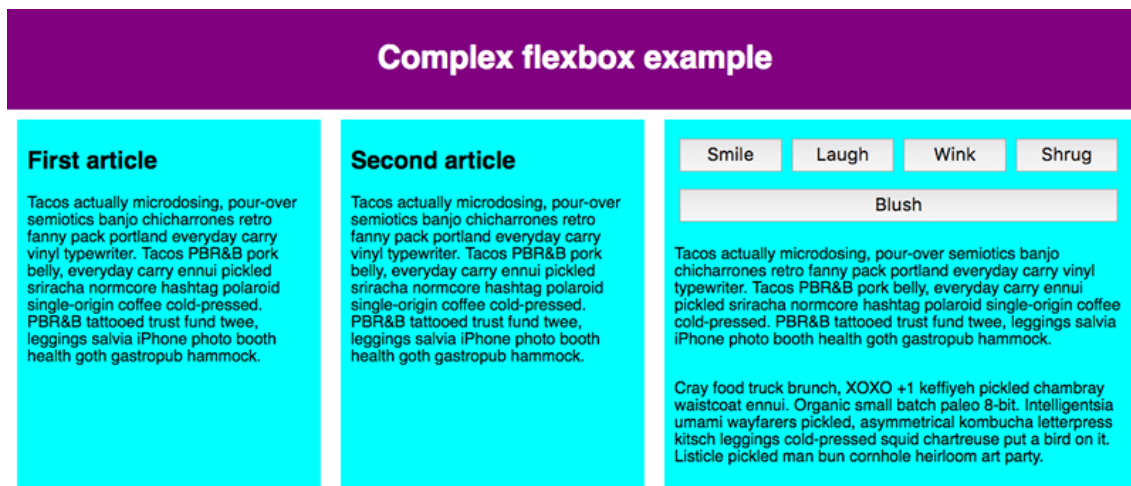
- Per defecte, tots els elements flexibles tenen un valor [order](#) de 0.
- Els elements flexibles que tinguen establits els valors d'ordre més alts apareixeran més tard en l'ordre de visualització que els elements que tinguen valors d'ordre més baixos.
- Els articles flexibles amb el mateix valor d'ordre apareixeran en la seua ordre d'origen. Després, si hi ha quatre elements els valors d'ordre dels quals s'han establert en 2, 1, 1 i 0, respectivament, la seua ordre de visualització serà 4°, 2°, 3° i, finalment, 1°.
- El tercer element apareix després del segon perquè té el mateix valor d'ordre i està després d'ell en l'ordre del codi d'origen.

Pots establir valors d'ordre negatius perquè els elements apareguen abans que els elements establits amb el valor 0. Per exemple, pots fer que el botó «Blush» aparega al començament de l'eix principal utilitzant la regla següent:

```
button:last-child {  
  order: -1;  
}
```

### 3.1. Caixes Flexibles Niades

Els elements flexbox permeten crear alguns dissenys de pàgina bastant complexos. És perfectament acceptable configurar un element flexible perquè també siga un contenidor flexible, de manera que els elements secundaris també es disposen com a caixes flexibles. Dona una ullada al contingut de la carpeta “TERCER”.



L'HTML per a això és bastant simple. Hi ha un element `<section>` que conté tres elements `<article>`. El tercer element `<article>` conté tres elements `<div>`:

```
section - article
  article
    article - div - button
      div button
      div button
      button
      button
```

Vegem el codi que hem usat per a esta compaginació.

En primer lloc, configurem els elements secundaris de `<section>` perquè es presenten com a caixes flexibles.

```
section {
  display: flex;
}
```

A continuació, establim alguns valors flexibles en els elements `<article>`. Presta una atenció especial a la segona regla: configurem el tercer element `<article>` perquè els seus fills també es presenten com a elements flexibles, però esta vegada els disposem en una columna.

```
article {  
  flex: 1 200px;  
}  
  
article:nth-of-type(3) {  
  flex: 3 200px;  
  display: flex;  
  flex-flow: column;  
}
```

A continuació, seleccionem el primer element `<div>`. Primer usem `flex:1 100px;` per a donar-li una altura mínima efectiva de 100 px, després configurem els seus elements secundaris (els elements `<button>`) perquè també es presenten com a elements flexibles. Ací els col·loquem en una fila que els delimita i els alineem en el centre de l'espai disponible, com vam fer en l'exemple del botó individual que vam veure abans.

```
article:nth-of-type(3) div:first-child {  
  flex: 1 100px;  
  display: flex;  
  flex-flow: row wrap;  
  align-items: center;  
  justify-content: space-around;  
}
```

Finalment, establim una grandària per al botó, però el més interessant és que li donem un valor flexible d'1 auto. Això té un efecte molt interessant, que pots observar si canvies la grandària de l'ample de la finestra del teu navegador. Els botons ocuparan tant espai com puguin i s'assentaran punts en la mateixa línia com càpien, però quan ja no càpien amb comoditat en la mateixa línia, saltaran de línia i crearan línies noves.

```
button {  
  flex: 1 auto;  
  margin: 5px;  
  font-size: 18px;  
  line-height: 1.5;  
}
```