

# M4. Llenguatges de marques

## – BA 1 – CSS AVANÇAT

INSTITUT PÚBLIC DE SABADELL

Marc Albareda

Eloi Vazquez

# INDEX

- INDEX
  - FLEXBOX
  - RESPONSIVE

# FLEXBOX

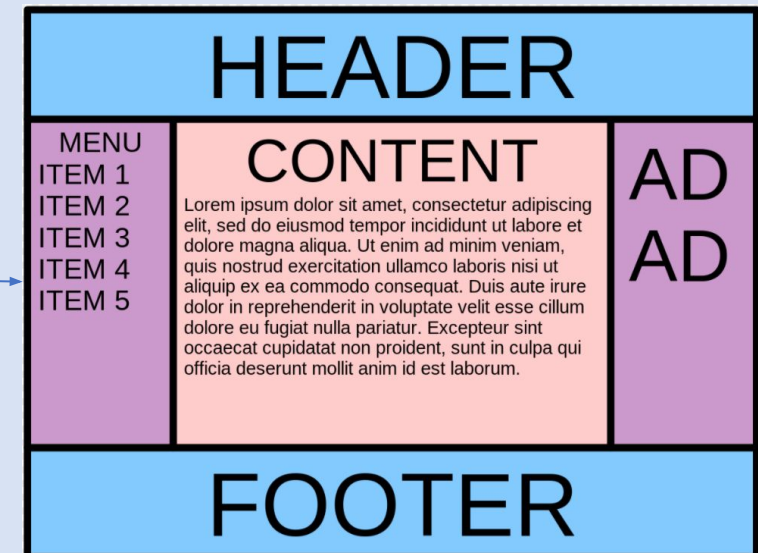
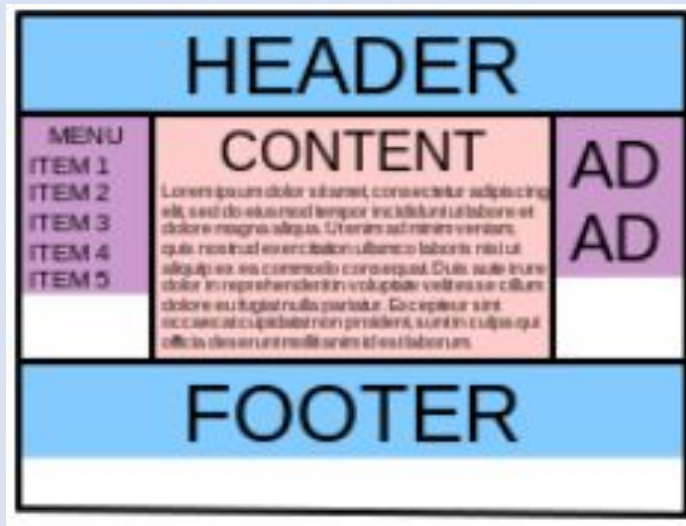
- Hi ha 4 estils de disseny:
  - Blocs per seccions a una pàgina web.
  - En línia pel text.
  - Taula per a la representació en 2 dimensions de les dades.
  - Posicionament per a la posició explícita d'un element.
- Flexbox facilita el disseny sense float o positioning.

# FLEXBOX

- Problema:
  - Sant Grial del disseny web que té múltiples columnes amb la mateixa alçada definides al CSS.
  - Les limitacions de CSS i HTML:
    - Per la conveniència de què les pàgines siguin significatives. Així els motors de cerca poden classificar millor les pàgines.
    - Deficiències dels diversos navegadors.
  - La solució va arribar amb Flexbox i Grid.

# FLEXBOX

- Problema:



# FLEXBOX

- Problema:
  - Més informació a:
    - [https://en.wikipedia.org/wiki/Holy\\_grail\\_\(web\\_design\)#:~:text=The%20holy%20grail%20is%20a,available%20technologies%20all%20had%20drawbacks](https://en.wikipedia.org/wiki/Holy_grail_(web_design)#:~:text=The%20holy%20grail%20is%20a,available%20technologies%20all%20had%20drawbacks)
  - Una solució, la podeu trobar a:
    - [https://codepen.io/vmasto\\_1470672674/pen/YEVPdm](https://codepen.io/vmasto_1470672674/pen/YEVPdm)
  - Video fent la programació amb position Holy Grail:
    - [https://www.youtube.com/watch?v=haAw7\\_HBc24&feature=emb\\_logo](https://www.youtube.com/watch?v=haAw7_HBc24&feature=emb_logo)
  - Video fent la programació amb flexbox del Holy Grail:
    - [https://www.youtube.com/watch?v=szkaT7ccgkl&feature=emb\\_logo](https://www.youtube.com/watch?v=szkaT7ccgkl&feature=emb_logo)

# FLEXBOX

- Per començar a utilitzar el model Flexbox, primer s'ha de definir un contenidor flex.
- L'element de sota representa un contenidor flexible (l'àrea blava) amb tres elements flexibles.



# FLEXBOX

- Flexbox: Contenidor
  - El contenidor es converteix en flexible quan s'assigna el valor flex a l'element display.

```
.flex-container {  
  display: flex;  
}
```



# FLEXBOX

- Les propietats de flexbox són:
  - flex-direction
  - flex-wrap
  - flex-flow
  - justify-content
  - align-items
  - align-content
  - gap (genera un espai en blanc entre 2 items)

# FLEXBOX

- Propietat flex-direction:
  - Defineix la direcció de com apila els elements flexibles del contenidor.
  - Els valors són:
    - column
    - column-reverse
    - row
    - row-reverse

# FLEXBOX

- Proprietat flex-direction:
  - Exemple

```
<!DOCTYPE html>
<html>

  <head>
  </head>

  <body>
    <h1>The flex-direction Property</h1>

    <h2>Row Reverse</h2>
    <div class="flex-container-row-reverse">
      <div>1</div>
      <div>2</div>
      <div>3</div>
    </div>

    <h2>Row</h2>
    <div class="flex-container-row">
      <div>1</div>
      <div>2</div>
      <div>3</div>
    </div>
```

```
    <h2>Column</h2>
    <div class="flex-container-column">
      <div>1</div>
      <div>2</div>
      <div>3</div>
    </div>
    <h2>Column</h2>
    <div class="flex-container-column-reverse">
      <div>1</div>
      <div>2</div>
      <div>3</div>
    </div>
  </body>
</html>
```

```
.flex-container-row-reverse {
  display: flex;
  flex-direction: row-reverse;
  background-color: DodgerBlue;
}

.flex-container-row-reverse > div {
  background-color: #f1f1f1;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}

.flex-container-row {
  display: flex;
  flex-direction: row;
  background-color: DodgerBlue;
}

.flex-container-row > div {
  background-color: #f1f1f1;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
```

# FLEXBOX

- Propietat flex-direction:
  - Exemple (Continuació)

```
▼ .flex-container-column {  
  display: flex;  
  flex-direction: column;  
  background-color: DodgerBlue;  
}  
  
▼ .flex-container-column > div {  
  background-color: #f1f1f1;  
  width: 100px;  
  margin: 10px;  
  text-align: center;  
  line-height: 75px;  
  font-size: 30px;  
}  
  
▼ .flex-container-column-reverse > div {  
  background-color: #f1f1f1;  
  width: 100px;  
  margin: 10px;  
  text-align: center;  
  line-height: 75px;  
  font-size: 30px;  
}  
  
▼ .flex-container-column-reverse {  
  display: flex;  
  flex-direction: column-reverse;  
  background-color: DodgerBlue;  
}
```

## The flex-direction Property

### Row Reverse



### Row



### Column

# FLEXBOX

- Propietat flex-wrap:
  - Especifica si els elements flex s'han d'ajustar o no.
  - Valors:
    - wrap
    - nowrap

# FLEXBOX

- L'exemple és el següent:

```
.flex-container-wrap {  
  display: flex;  
  flex-wrap: wrap;  
  background-color: DodgerBlue;  
}
```

```
.flex-container-wrap>div {  
  background-color: #f1f1f1;  
  width: 100px;  
  margin: 10px;  
  text-align: center;  
  line-height: 75px;  
  font-size: 30px;  
}
```

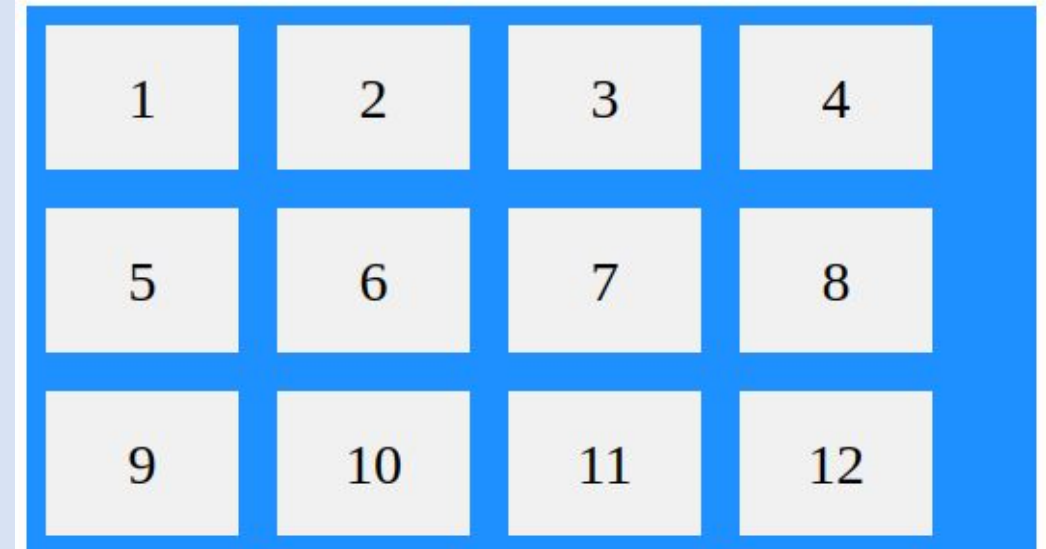
```
.flex-container-nowrap {  
  display: flex;  
  flex-wrap: nowrap;  
  background-color: DodgerBlue;  
}
```

```
.flex-container-nowrap>div {  
  background-color: #f1f1f1;  
  width: 100px;  
  margin: 10px;  
  text-align: center;  
  line-height: 75px;  
  font-size: 30px;  
}
```

## The flex-wrap Property

Intenta redimensionar el navegador.

### Wrap



### No Wrap



# FLEXBOX

- Propietat flex-wrap:

- flex-flow

- És una propietat abreujada que conte de forma abreujada les propietats flex-direction i flex-wrap.

- Per exemple:

```
.flex-container {  
  display: flex;  
  flex-flow: row wrap;  
}
```

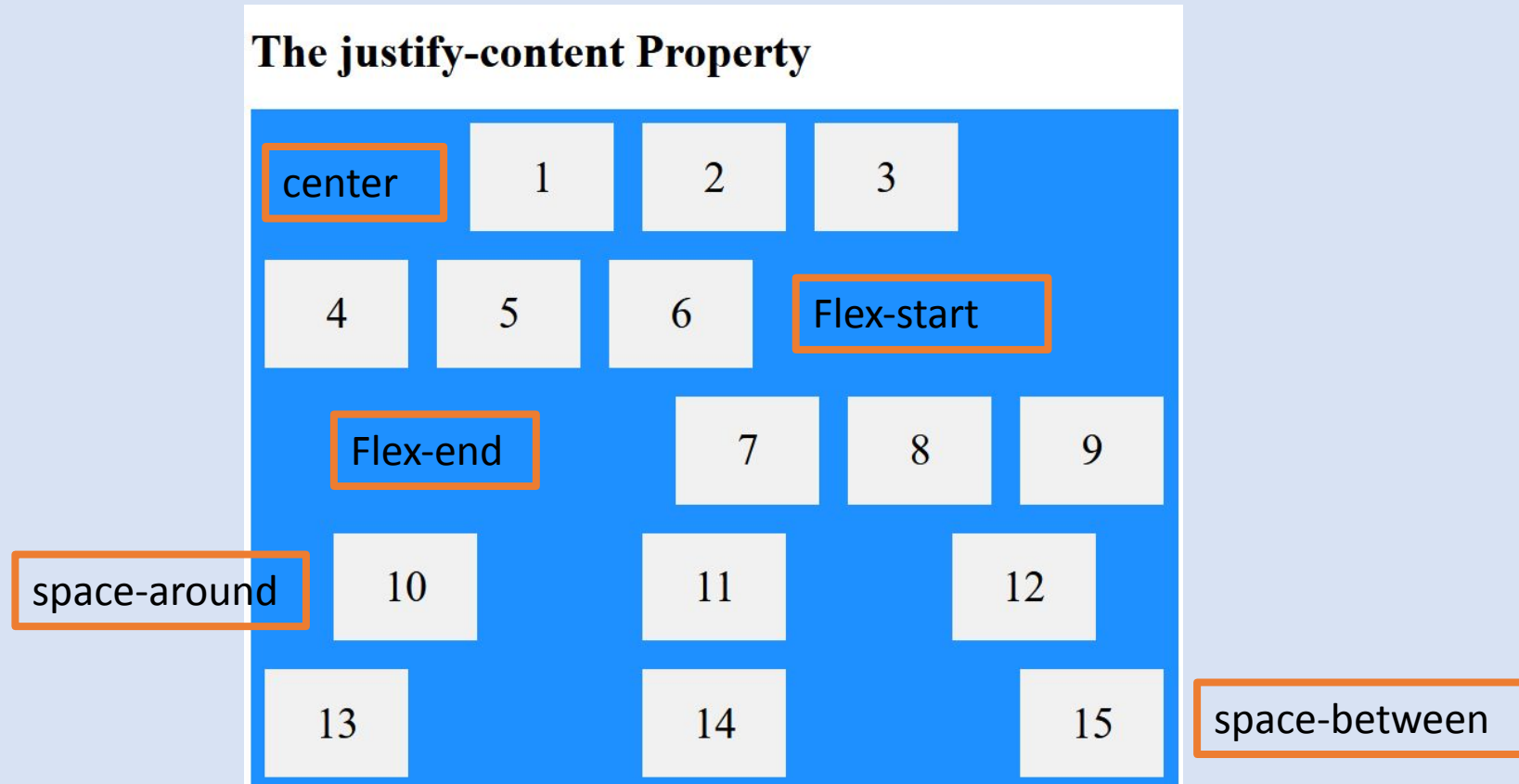
# FLEXBOX

- Propietat justify-content:
  - S'utilitza per alinear els elements flex.
  - Els valors de la propietat són:
    - center
    - flex-start
    - flex-end
    - space-around
    - space-between



# FLEXBOX

- Propietat justify-content:
  - L'exemple és el següent on es poden veure tots els valors de les propietats:



# FLEXBOX

- Propietat align-items:
  - S'utilitza per alinear els elements flex.
  - Els valors de la propietat són:
    - center □ alinea els elements flex al mig del contenidor.
    - flex-start □ alinea els elements flex a la part superior del contenidor.
    - flex-end □ alinea els elements flex a la part inferior del contenidor.
    - stretch □ estén els elements flex per omplir el contenidor (això és per defecte).
    - Baseline □ alinea els elements flex tal com s'alinien les seves línies de base.

# FLEXBOX

- Propietat align-items:
  - L'exemple és el següent on es poden veure tots els valors de les propietats:

**The align-items Property**

1	2	3	center	
4	5	6	flex-start	
7	8	9	flex-end	
10	11	12	stretch	
13	14	15	16	baseline

# FLEXBOX

- Propietat align-content:
  - S'utilitza per alinear les línies de flex.
  - Els valors de la propietat són:
    - space-between □ mostra les línies flex amb un espai igual entre elles.
    - space-around □ mostra les línies flex amb espai abans, entre i després d'elles.
    - stretch □ estén les línies flex per ocupar l'espai restant (això és per defecte).
    - center □ mostra les línies flex al mig del contenidor.
    - flex-start □ mostra les línies flex a l'inici del contenidor.
    - flex-end □ mostra les línies flex al final del contenidor.

# FLEXBOX

- Propietat align-content:
  - L'exemple és el següent on es poden veure tots els valors de les propietats:

## The align-content Property

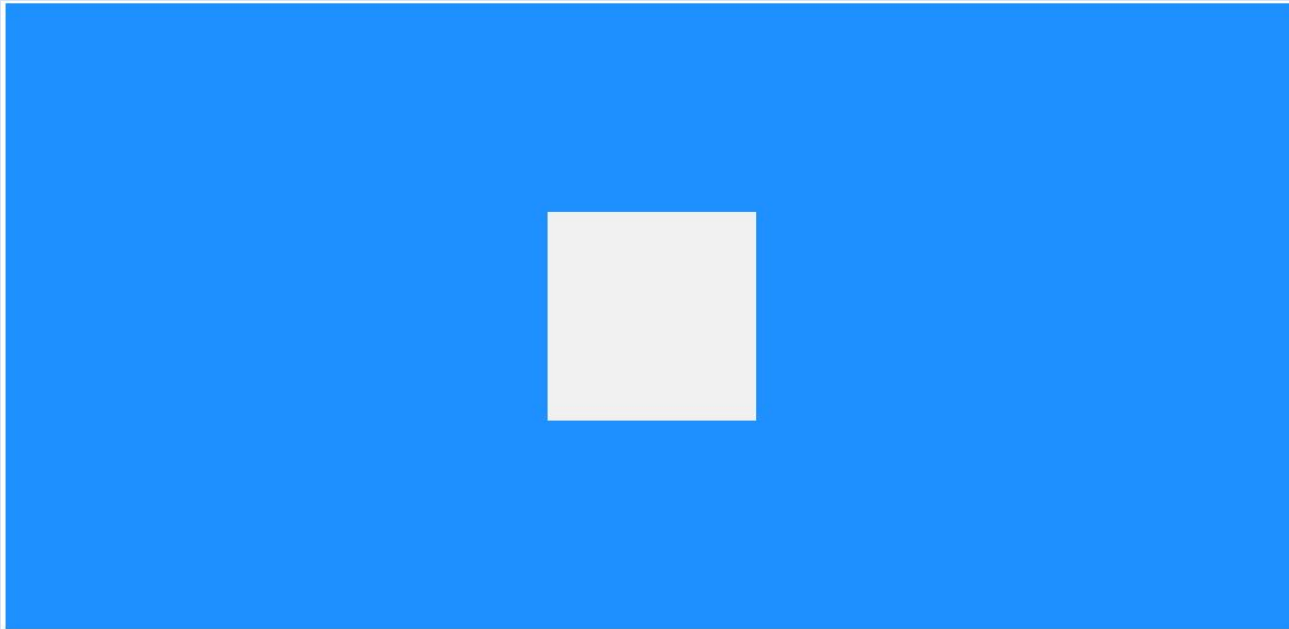
1	2	3	4	5	6
space-between					
7	8	9	10	11	12
1	2	3	4	5	6
space-around					
7	8	9	10	11	12
1	2	3	4	5	6
stretch					
7	8	9	10	11	12

center

1	2	3	4	5	6
7	8	9	10	11	12
flex-end					
1	2	3	4	5	6
7	8	9	10	11	12
1	2	3	4	5	6
7	8	9	10	11	12
Flex-start					

# FLEXBOX

- Problema: Centrat perfecte
  - Solució:
    - Estableix les propietats justify-content i align-items per centrar i l'element flex estarà perfectament centrat.



```
.flex-container {  
  display: flex;  
  height: 300px;  
  justify-content: center;  
  align-items: center;  
}
```

# FLEXBOX

- Descendents de Flex: Items
  - Els elements descendents de flex esdevenen items de flex.
  - Al següent exemple tenim un contenidor flex gris i 4 items de color blau:



# FLEXBOX

- Items flex:
  - Les propietats dels ítems són:
    - order □ Especifica l'ordre dels ítems
    - flex-grow □ Especifica quan creixerà un element flex en relació amb la resta d'elements flex.
    - flex-shrink □ Especifica quan es reduirà un element flex en relació amb la resta d'elements flex.
    - flex-basis □ Especifica la longitud inicial d'un element flex.
    - flex □ És una propietat abreujada de les propietats flex-grow, flex-shrink i flex-base.
    - align-self □ especifica l'alineació de l'element seleccionat dins del contenidor flex. Substitueix l'alineació predeterminada establerta per la propietat align-items del contenidor.



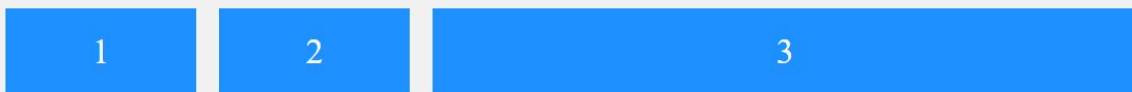
# FLEXBOX

- Items flex:
  - L'exemple és el següent on es poden veure tots els valors de les propietats:

## The order Property



## The flex-grow Property



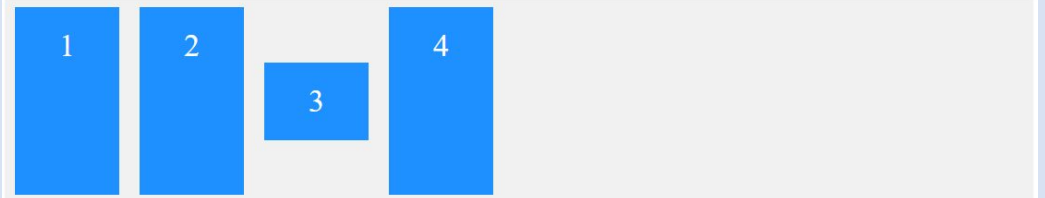
## The flex-shrink Property



## The flex-basis Property



## The align-self Property



## The flex Property



# FLEXBOX

- Enllaços d'ajuda:
  - <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
  - [https://www.w3schools.com/css/css3\\_flexbox.asp](https://www.w3schools.com/css/css3_flexbox.asp)

# FLEXBOX

- Responsive amb Flexbox:
  - Mitjançant media queries per crear diferents dissenys per diferents tamanys de pantalla i dispositius.
  - Almenys tenim 2 maneres:
    - Si volem crear un disseny de 2 columnes per molts tamanys de pantalla i una columna per pantalles petites, podem canviar flowdirection de row a column a partir d'un tamany de pantalla.
    - També, es pot canviar el percentatge de la propietat flex i posar la propietat flex-wrap i igual a wrap.

# FLEXBOX

- Responsive:
  - Exemple 1 de responsive:

## Responisve Flexbox

Opció 1

Superior a 800



## Responisve Flexbox

Opció 1

Inferior o igual a 800



```
/* Responsive layout - makes a one column-layout instead of two-column layout */  
@media (max-width: 800px) {  
  .flex-container {  
    flex-direction: column;  
  }  
}
```

# FLEXBOX

- Responsive:
  - Exemple 2 de responsive:

## Responsive Flexbox

Opció 2

Superior a 800

1

2

## Responsive Flexbox

Opció 2

Inferior o igual a 800

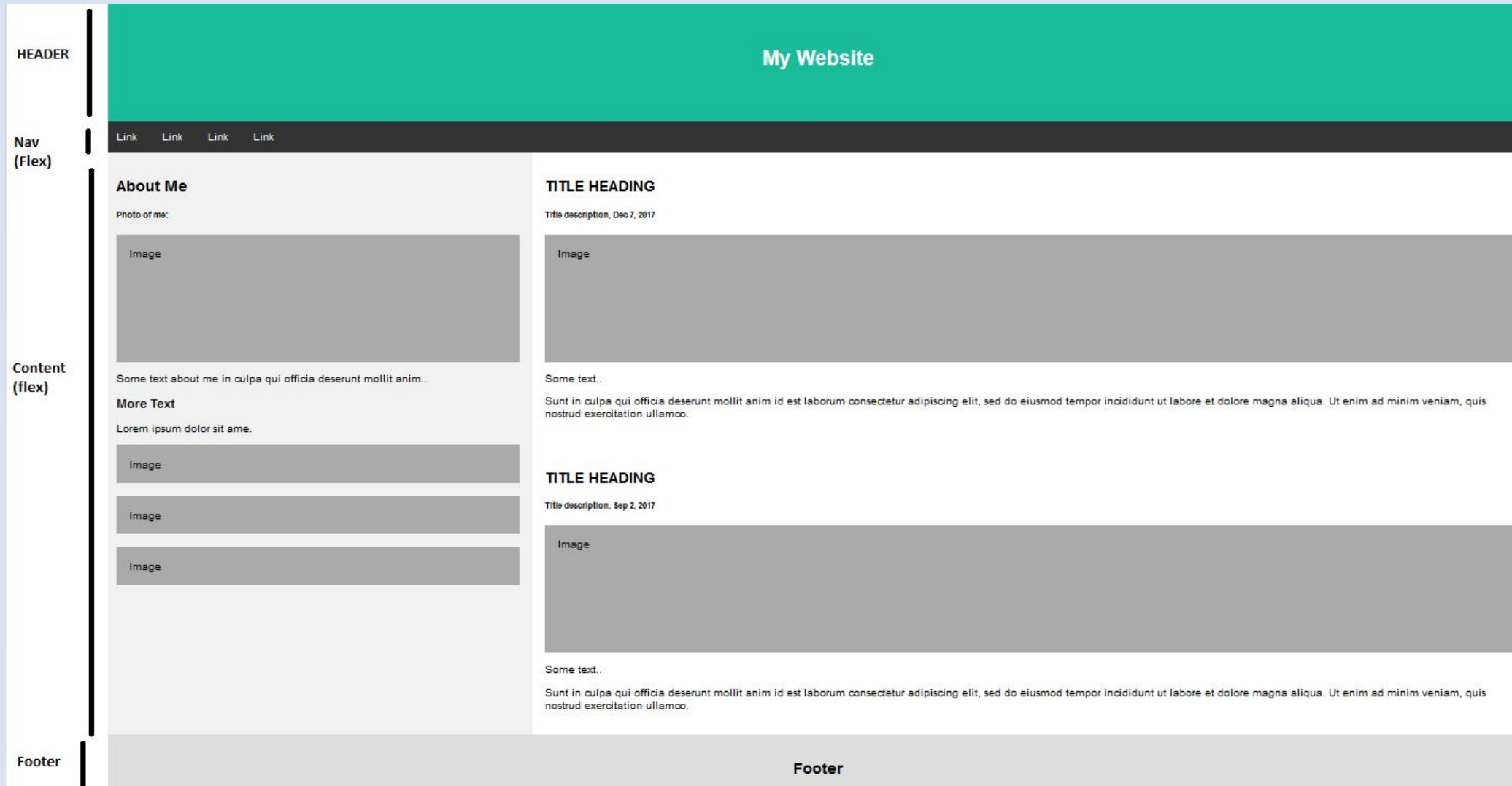
1

2

```
/* Responsive layout - makes a one column-layout instead of a two-column layout */  
@media (max-width: 800px) {  
  
  .flex-item-right,  
  .flex-item-left {  
    flex: 100%;  
  }  
}
```

# LAYOUT AMB FLEXBOX

Com  
fer  
un  
bon  
layout?



# LAYOUT AMB FLEXBOX (Nav)

```
8
9 <!-- Navigation Bar -->
10 <div class="navbar">
11   <a href="#">Link</a>
12   <a href="#">Link</a>
13   <a href="#">Link</a>
14   <a href="#">Link</a>
15 </div>
16
```

```
18
19 /* Style the top navigation bar */
20 .navbar {
21   display: flex; /*es flex, com que no especifiquem es per row */
22   background-color: #333;
23 }
24
25 /* links dintre del navbar */
26 .navbar a {
27   color: white;
28   padding: 14px 20px; /*esta delimitat per el text intern, no te un %"
29   text-decoration: none;
30   text-align: center;
31   /*fem una animació al boto*/
32   transition: background 1s, color 1s;
33 }
34
35 /* al fer hover canviem el botó amb la transició feta */
36 .navbar a:hover {
37   background-color: #ddd;
38   color: black;
39 }
40
```

# LAYOUT AMB FLEXBOX (Main)

```
<!-- The flexible grid (content) -->
<div class="row">
  <div class="side">
    <h2>About Me</h2>
    <h5>Photo of me:</h5>
    <div class="fakeimg" style="height:200px;">Image</div>
    <p>Some text about me in culpa qui officia deserunt mollit anim..</p>
    <h3>More Text</h3>
    <p>Lorem ipsum dolor sit ame.</p>
    <div class="fakeimg" style="height:60px;">Image</div><br>
    <div class="fakeimg" style="height:60px;">Image</div><br>
    <div class="fakeimg" style="height:60px;">Image</div>
  </div>
  <div class="main">
    <div class="article">
      <h2>TITLE HEADING</h2>
      <h5>Title description, Dec 7, 2017</h5>
      <div class="fakeimg" style="height:200px;">Image</div>
      <p>Some text..</p>
      <p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim ve
niam, quis nostrud exercitation ullamco.</p>
    </div>
    <div class="article">
      <h2>TITLE HEADING</h2>
      <h5>Title description, Sep 2, 2017</h5>
      <div class="fakeimg" style="height:200px;">Image</div>
      <p>Some text..</p>
      <p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim ve
niam, quis nostrud exercitation ullamco.</p>
    </div>
  </div>
</div>
```

```
/* El cos principal, es flex */
.row {
  display: flex;
  flex-wrap: none;
}

/*Dintre de la row hi ha dos columnes de mida diferent */
.side { /*columna de l'esquerra amb l'about me*/
  flex: 30%; /*sera sempre un 30%*/
  background-color: #f1f1f1;
  padding: 20px;

  flex-shrink: 0; /*no disminueixis mai*/
}

.main { /* columna principal amb els articles */
  flex: 70%; /* sera sempre un 70%, això significa que sempre sera el doble aprox de l'altra */
  background-color: white;
  padding: 20px;
}
```



# FLEXBOX

- Extra per programa amb Flex:
  - <https://codepip.com/games/grid-garden/>

# RESPONSIVE

- Disseny Responsive:
  - Utilitza CSS i HTML per redimensionar, amagar, reduir, ampliar o moure el contingut perquè quedi bé en qualsevol pantalla.
  - Fa que la pàgina web es vegi bé en tots els dispositius.
  - Només utilitza HTML i CSS.
  - No és un programa ni un JavaScript.
  - Adapta els continguts a qualsevol dispositiu: ordinadors de sobretaula, tablet i telèfons mòbils. □ Té un bon aspecte i és fàcil d'utilitzar, independentment del dispositiu.

# RESPONSIVE

- Viewport:

- És l'àrea visible de l'usuari d'una pàgina web.
- Depenen del dispositiu, la pàgina varia el seu tamany.
- Viewport s'estableix a l'etiqueta meta d'HTML. Per exemple:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- width=device-width □ defineix l'amplada de la pàgina a partir de l'amplada de la pantalla del dispositiu.
- initial-scale=1.0 □ estableix el nivell de zoom inicial quan el navegador carrega la pàgina per primera vegada.

# RESPONSIVE

- Viewport
  - Regles per a què el redimensionament sigui una bona experiència del dispositiu:
    - NO utilitzeu elements d'amplada fixa grans
    - NO deixeu que l'amplada de viewport renderitzi l'amplada de la pàgina
    - Usa media queries aplicar diferents estils per als diferents tipus de pantalles de dispositius. □ Utilitza tamanyes a partir de percentatges (o relatius).
- Us de media queries
  - Ho hem vist a la diapositives [27](#) - [28](#).

# RESPONSIVE

- Exemple de Viewport:



# RESPONSIVE

- Responsive:
  - Exemple 2 de responsive:

## Responsive Flexbox

Opció 2

Superior a 800

1

2

## Responsive Flexbox

Opció 2

Inferior o igual a 800

1

2

```
/* Responsive layout - makes a one column-layout instead of a two-column layout */  
@media (max-width: 800px) {  
  
  .flex-item-right,  
  .flex-item-left {  
    flex: 100%;  
  }  
}
```