# Flexible Box Module

- Vocational Education and Training
- Web applications development.
- Second year.
- Subject – Web interfaces development.
- Unit 2 – CSS
- Professor: Vicent Tortosa.
- Official specification:
  - https://www.w3.org/TR/css-flexbox-1/

# Introduction

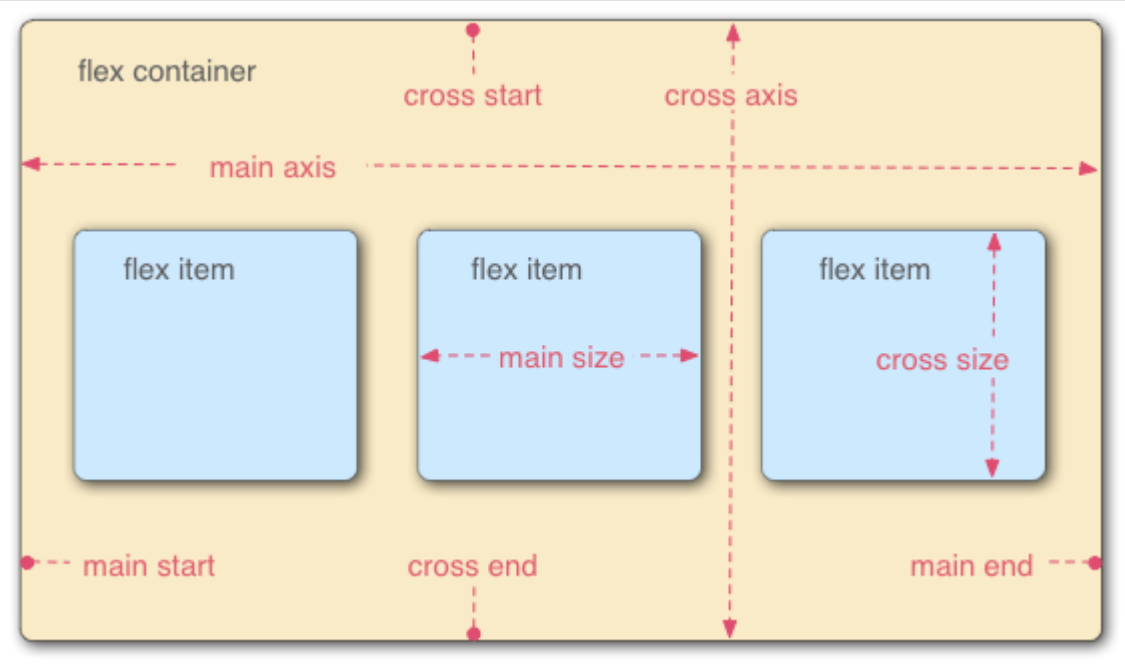
- Predictable behavior of the elements of a page when it must be accommodated in different devices.
- The child elements of a flexible box can be placed in any direction, horizontal or vertical, and can have flexible dimensions that adapt to free space or shrink.
- The display order is independent of the source code order.
- The CSS order does not affect the Screen Readers locution.

# Concept

- A flexible container expands or compresses its elements depending on the available box space.
- There is no set direction, horizontal or vertical as in a block model.
- The flexible box model is more appropriate for small scale models.
- For large scale we will use the grid model, which we will study later.
  - https://www.w3.org/TR/css-grid-1/

# Terminology

- The image shows a flexible container that has a `flex-direction` type `row` and according to the preset write mode is left-to-right.




- Direction `row` indicates that the `main axis` will be the horizontal axis and the `cross axis` the vertical axis. This would be the other way around if the direction had been set `column`.

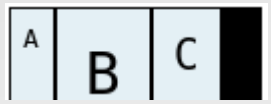
# Terminology

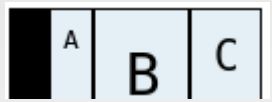
- `flex container`: element that contains flexible items, `display: flex` o `inline-flex`
- `flex item`: all boxes inside the flex cointainer become a flexible item.
- 2 axis: `main axis` where the flexible items appear one next to another, `cross axis`, runs perpendicular to the main axis.

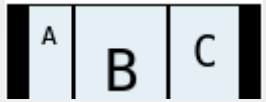
# Axes properties

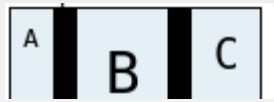
- `flex-direction` sets the main axis and says how the `flex items` will be ordered.
- Values:
  - `row`
  - `row-reverse`: the start and end lines are switched.
  - `column`
  - `column-reverse`: the start and end lines are switched.
- `https://developer.mozilla.org/en-US/docs/Web/CSS/flex-direction`

# Axes properties

- `justify-content` is used to align `flex item` on the main axis.
- Values:
  - `flex-start` 

  - `flex-end` 

  - `center` 

  - `space-between` 

  - `space-around` 


https://developer.mozilla.org/en-US/docs/Web/CSS/justify-content

# Axes properties

- `align-items` will align the items on the cross axis.
- Values: `stretch` by default
  - `flex-start:` the box starts in the start axis edge.
  - `flex-end:` the box starts in the end axis edge.
  - `center:` the box appears centered in the main axis.
  - `baseline:` all base items are aligned at the same line.
  - `stretch` o `normal:` takes all `flexbox` height.
- https://developer.mozilla.org/en-US/docs/Web/CSS/align-items

# Axes properties

- `align-self` is set in the flex item. It overrides the `align-item` value.
- Values:
  - `flex-start:` the box starts in the start axis edge.
  - `flex-end:` the box starts in the end axis edge.
  - `center:` the box appears centered in the main axis.
  - `baseline:` all base items are aligned at the same line.
  - `stretch` o `normal:` takes all `flexbox` height.

# Multi-line

- `flex-wrap,` specifies whether flex items are forced into a single line or can be wrapped onto multiple lines.
  - `nowrap`: by default. Items are laid out in a single line which may cause the flex container to overflow.
  - `wrap`: The flex items break into multiple lines..
  - `wrap-reverse`: Behaves the same as wrap but cross-start and cross-end are permuted.
- https://developer.mozilla.org/en-US/docs/Web/CSS/flex-wrap

# Direction

- `order:` Represents the ordinal number to be used by the flex item. May be positive, negative, or 0.

  https://developer.mozilla.org/en-US/docs/Web/CSS/order

- `flex-flow:` combines `flex-direction` and `flex-wrap.`

  https://developer.mozilla.org/en-US/docs/Web/CSS/flex-flow

# Dimensions

- `min-height, max-height:` sets the minimum and maximum height.

  https://developer.mozilla.org/en-US/docs/Web/CSS/min-height

- `min-width, max-width:` sets the minimum and maximum width.

  https://developer.mozilla.org/en-US/docs/Web/CSS/min-width

# Dimensions

- `flex:` combines `flex-grow, flex-shrink` and `flex-basis`

  https://developer.mozilla.org/en-US/docs/Web/CSS/flex
- Very important to read the documentation. Flex is a very important property. Depending on how many attibutes flex has and if they have units, the flex box behaves in one way or another.

# Dimensions

- `flex-basis:` specifices the inital **main size** of a flex item. If we have the three parameters, this is the first one to be read. If you give no value to it or there are free space in the flex box, Firefox will read the flex-grow value to increase its size.

  https://developer.mozilla.org/en-US/docs/Web/CSS/flex-basis

# Dimensions

- `flex-grow:` if there is free space in a container or there is not a flex basis parameter, flex-grow specifies the flex grow factor of a flex item.

  https://developer.mozilla.org/es/docs/Web/CSS/flex-grow

# Dimensions

- `flex-shrink:` when the default size of the flex items is larger than the flex container flex itemswill shrink to fill the container according to the `flex-shrink` number.

  https://developer.mozilla.org/en-US/docs/Web/CSS/flex-shrink

# Properties not affecting **`flex box`**

- `column-*` from Multicol Module.
- `float` and `clear` have not effect in the flex items.
- `vertical-align` has no effect in the vertical items alignment.

# Examples

- Check out the examples I give to you, read the html and CSS code to understand the flex box behaviour:
- 4_Ejemplo_basico_Flex.html
- 5_Ejemplo_de_Disenyo_del_Santo_Grial.html
- 6_flexbox_paises
- 7_Ejemplo_Flex_Box_Multiple.html
- See that I use `media queries` where I use responsive design. Do not struggle with it, we'll see it soon.

# Webography and examples

- https://www.w3.org/TR/css-flexbox-1/
- https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox
- https://www.emenia.es/flexbox-la-caja-flexible-css3/
- https://css-tricks.com/snippets/css/a-guide-to-flexbox/
- https://css-tricks.com/snippets/css/complete-guide-grid/
- Complex layout with `flexible boxes`:
  - https://codepen.io/moyicat/pen/qaWPba

# Creative Commons License
# Attribution-ShareAlike – CC BY-SA

**You are free to:**

**Share** — copy and redistribute the material in any medium or format

**Adapt** — remix, transform, and build upon the material

for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

**Under the following terms:**

**Attribution** — You must give **appropriate credit**, provide a link to the license, and **indicate if changes were made**. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

**ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the **same license** as the original.

**No additional restrictions** — You may not apply legal terms or **technological measures** that legally restrict others from doing anything the license permits.

http://creativecommons.org/licenses/by-sa/4.0/