

# Flexbox

## Layout and structure

Shadi Lahham - Programmazione web - Frontend - HTML e CSS

in collaborazione con:



per una crescita intelligente,  
sostenibile ed inclusiva

[www.regione.piemonte.it/europa2020](http://www.regione.piemonte.it/europa2020)

INIZIATIVA CO-FINANZIATA CON FSE

# Using Flex



# What is Flexbox

---

- A web layout model
- Consists of responsive elements (Flex boxes) within a container
- Elements can be ordered and arranged in various ways (spacing, alignment)
- Elements can grow and shrink based on container size
- Content can flow in various directions, e.g. left, right, up, down

## Float vs Flexbox

---

- Float
  - Complex
  - Requires the use of float, margin, width and clear to work properly
  - Works on all browsers, even very old ones
- Flexbox
  - Easy and intuitive
  - Can do a lot more than float
  - Works on all modern browsers

## Flexbox in a nutshell

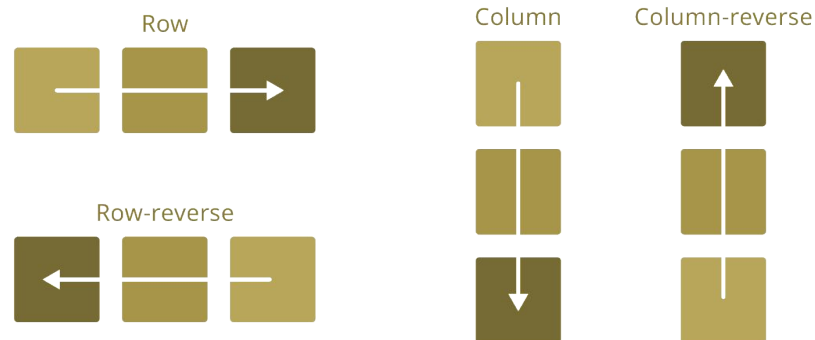
---

- Container and elements
  - Flex boxes appear within a container
  - Flex determines the layout which is easily responsive
- Direction and wrap
  - In what sequence elements appear
  - What happens when space runs out
- Justify and align
  - Spacing between elements on the main axis
  - Alignment of elements with respect to each other
- Grow, shrink, basis
  - How elements grow or shrink based on the changing space of the container

# Direction and wrap

In what sequence the elements appear and what happens when space runs out

The order of the items can also be customized with the order property



[flex-direction](#)

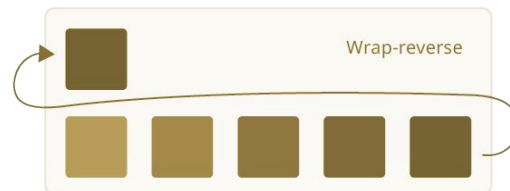
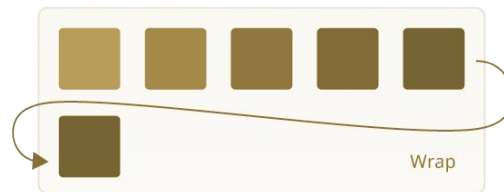
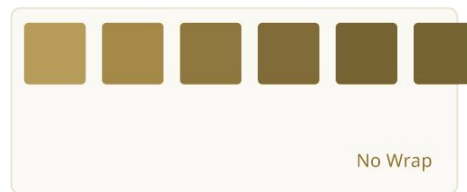
[flex-wrap](#)

[flex-flow](#)

[the order property](#)

# Direction and wrap

In what sequence the elements appear and what happens when space runs out



[flex-direction](#)

[flex-wrap](#)

[flex-flow](#)

# Your turn





# Justify and align

Spacing between elements on the main axis and  
alignment of elements with respect to each other

Flex-start



Flex-end



Center



Space-between



Space-around



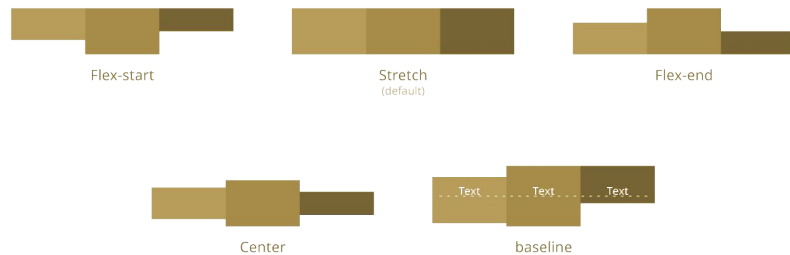
[justify-content](#)

[align-items](#)

[Aligning Items in a Flex Container](#)

# Justify and align

Spacing between elements on the main axis and  
alignment of elements with respect to each other



[justify-content](#)

[align-items](#)

[Aligning Items in a Flex Container](#)

## Grow, shrink, basis

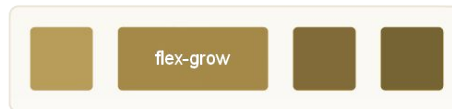
How elements grow or shrink based on the changing space of the container

flex-grow determines if the element can take up additional space or not

flex-shrink similar to flex-grow but determines how much an element is allowed to shrink

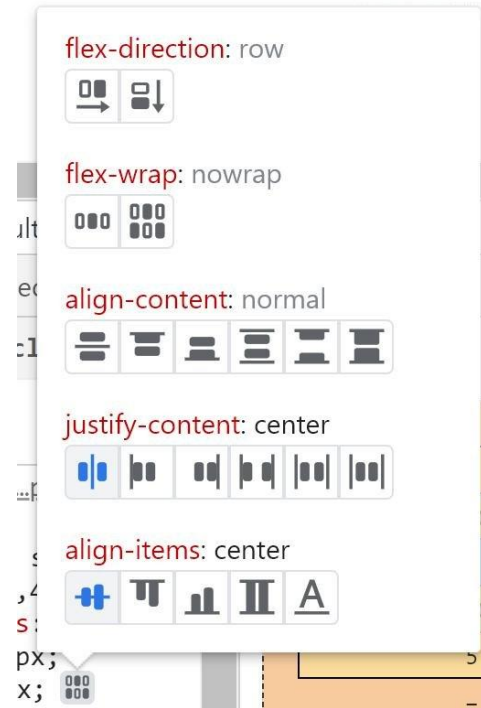
flex-basis sets the default size of a node along the main axis

flex shorthand for grow,shrink,basis  
e.g. `flex: 1 0 200px;`



# Chrome dev tools

Chrome devtools support flexbox and have a friendly UI for editing flex properties



[Flexbox debugging In DevTools](#)

# 1.Presentation

Study all the links in references and use them to create a short presentation

- Presentation should be 10-15 slides
- Explain in detail how flexbox works
- Include your own code in the presentation
- Show at least 10 different code examples
- Include references to sites that have interesting usages of flex
- Include your `html` and `css` files in the delivery

# Bonus



## 2.Flex games

---

Complete all levels of the following flex games for learning CSS flexbox

[Flexbox Froggy](#)

[Flexbox Defense](#)

Submit an .md file with the answer for each level

### 3.To flex or not to flex

---

Read the included article from The Blog of Karen Menezes

It's an old article but still contains some useful hints on when to use Flex and when to use basic CSS properties.



## References

---

[A Complete Guide to Flexbox](#)

[Basic concepts of flexbox](#)

[CSS Flexbox Container](#)

[Flexbox](#)

[Even more about how Flexbox works \(animated examples\)](#)

[Build with Flexbox \(nice playground\)](#)

## References

---

[Solved by Flexbox — Cleaner, hack-free CSS](#)

[Flexbox Patterns](#)

[Flexbox debugging In DevTools](#)