

# CSS Layout - Flexbox



**SoftUni Team**  
Technical Trainers



**SoftUni**



**Software University**

<https://softuni.bg>

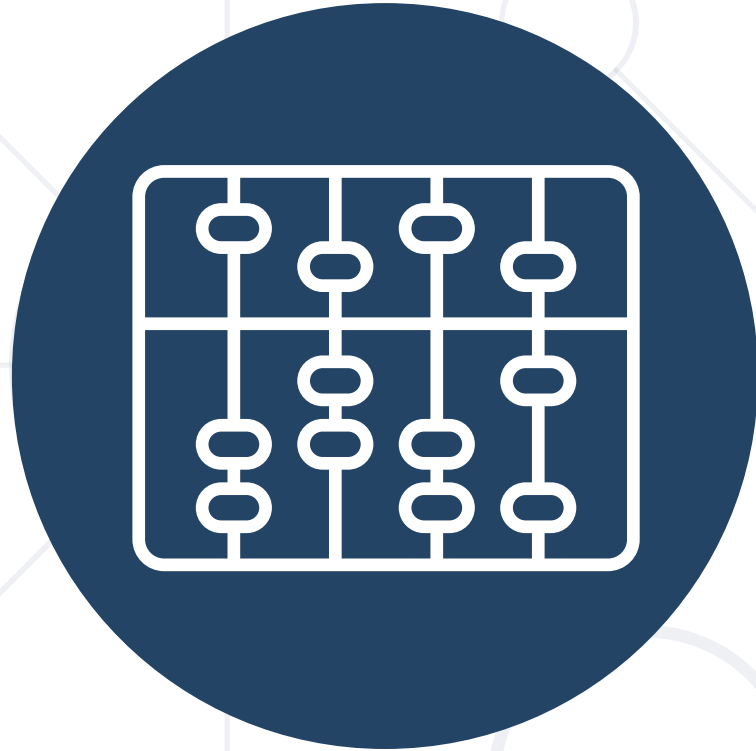
sli.do

**#html-css**

# Table of Contents

1. Flexbox
2. Properties for the Container
3. Properties for the Children

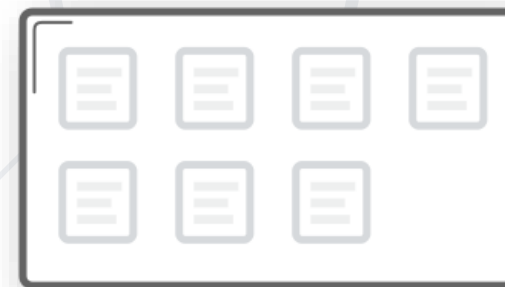




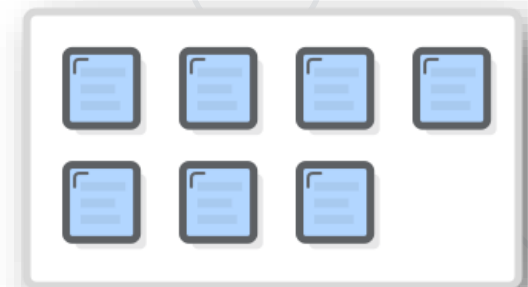
**Flexbox**

# What is Flexbox?

- The **Flexible Box Module**, usually referred to as **flexbox**, was designed as a one-dimensional layout model, and as a method that could offer space distribution between items in an **interface** and **powerful** alignment capabilities
- [https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS\\_layout/Flexbox](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox)



“FLEX CONTAINER”



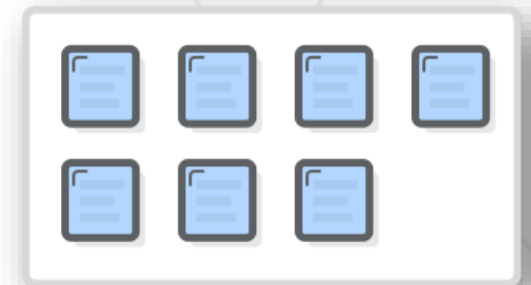
“FLEX ITEMS”

# Why Flexbox?

- For a long time, the only **reliable cross browser-compatible** tools available for creating CSS layouts were things like **floats** and **positioning**
  - These are fine and they work, but in some ways, they are also rather limiting and frustrating
- **Flexbox** can be look at as a logical system to arrange **boxes in 2 dimensions**
- There are many places where an understanding of the ideas of Flexbox will be very useful

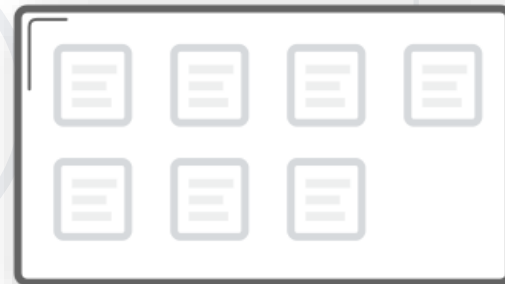


“FLEX CONTAINER”



“FLEX ITEMS”

- For elements to be laid out as **flexible boxes** we set a special value of display on the parent element of the elements you want to affect
- This causes the element to become a **flex container** and its children to become **flex items**
- When elements are laid out as flex items, they are laid out along two axes:

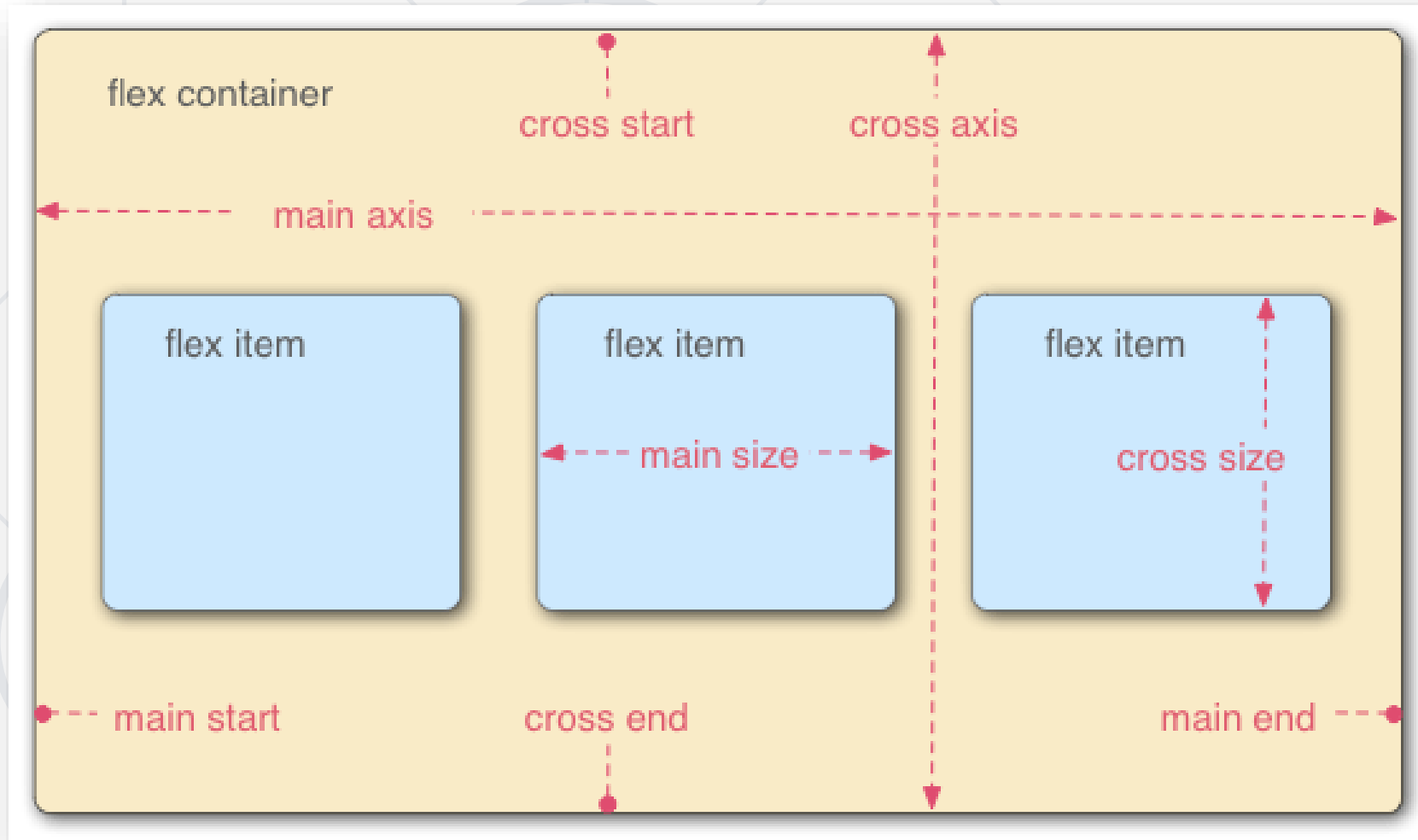


“FLEX CONTAINER”



“FLEX ITEMS”

# The Flex Model





- Complexity and the value of reference
  - The entire specification with all possible properties that are related to flexbox can be **overwhelming**, especially in the beginning
  - Don't try to learn and remember everything
  - Try to **understand** the basic concepts – the properties will be learned through exercise and repetition
  - <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

- **Container and Item properties**
  - We can **separate the collection** of properties related to Flexbox in to two main categories
  - Properties that are applied to the **Flex container** and properties applied to the **Flex items**



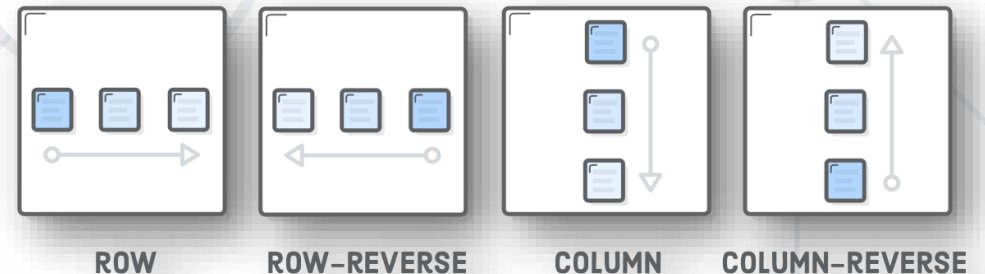
- **display: flex;**
  - We set a value of 'flex' to the **display** property on the **parent element** of the elements you want to affect

```
.box {  
  display: flex;  
}
```

# Flexbox Properties – Flex Container

- Columns or rows? - **flex-direction**
  - Flexbox provides a property called **flex-direction** that specifies which direction the main axis runs (which direction the flexbox children are laid out in)
  - By default, this is set to row, which causes them to be laid out in a row in the direction your browser's default language works in

```
.box {  
  flex-direction: row;  
}
```



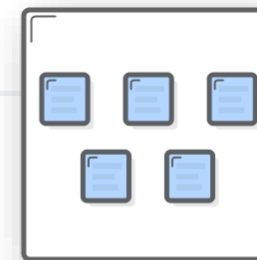
# Flexbox Properties – Flex Container

- Wrapping - **flex-wrap**
  - One issue that arises when you have a fixed **width** or **height** in your layout is that eventually your **flexbox children** will overflow their container, breaking the layout

```
.box {  
  flex-wrap: wrap;  
}
```



**NO WRAPPING**  
FLEX-WRAP: NOWRAP;

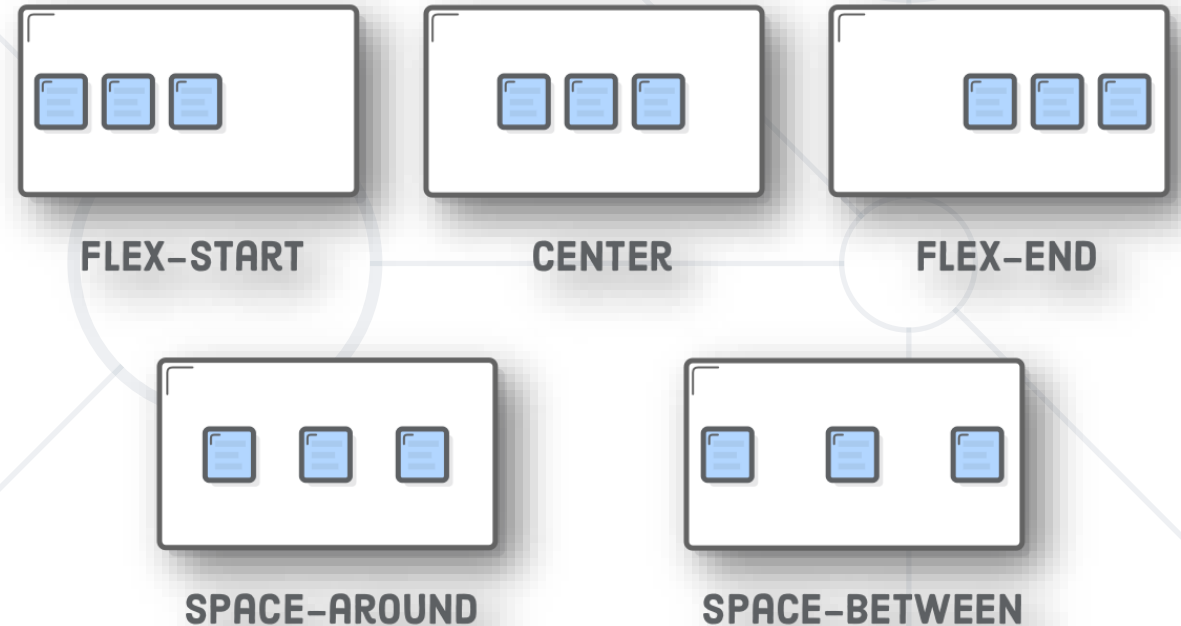


**WITH WRAPPING**  
FLEX-WRAP: WRAP;

# Flexbox Properties – Flex Container

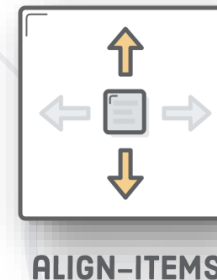
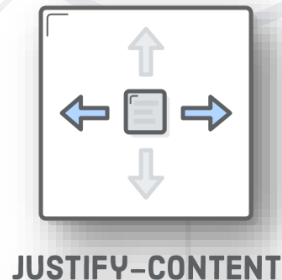
- Horizontal alignment- **justify-content**
  - Defines how the browser distributes **space between** and **around content** items along the **main-axis** of a flex container, and the inline axis of a grid container

```
.box {  
  justify-content: center;  
}
```

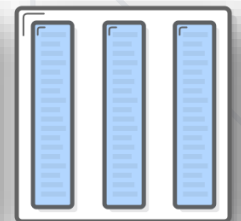
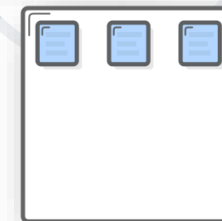


# Flexbox Properties – Flex Container

- Horizontal alignment- **align-items**
  - Sets the **align-self** value on all direct children as a group
  - In **Flexbox**, it controls the alignment of items on the **Cross Axis**



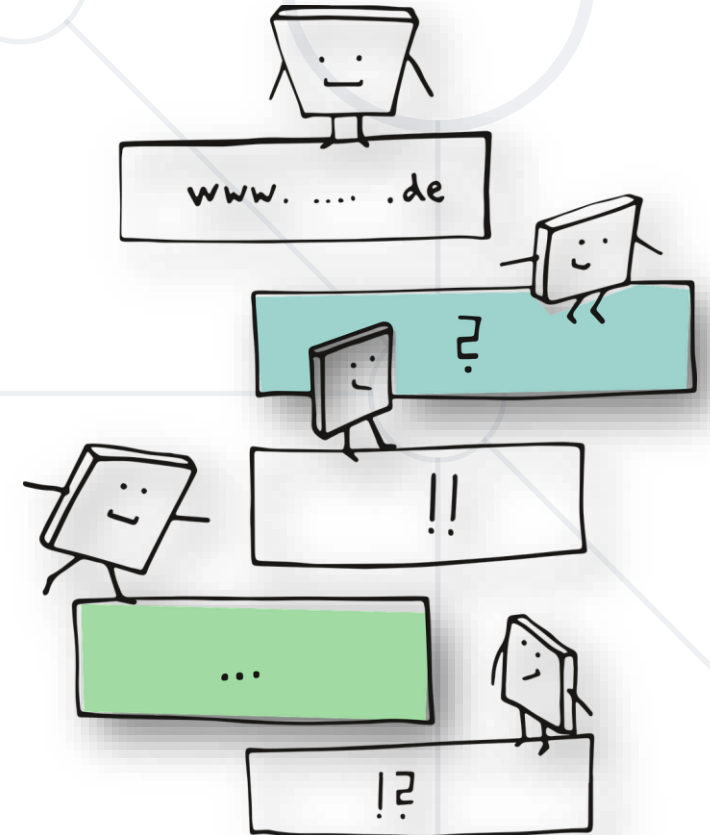
```
.box {  
  align-items: stretch;  
}
```



# Flexbox Properties – Flex Container

- Spacing - **gap**
  - Sets the gaps (gutters) between **rows** and **columns**

```
.box {  
  gap: 2em;  
}
```





# Flexbox Properties – Flex Item

- Ordering flex items - **order**
  - Sets the order to lay out an item in a **flex container**
  - Items in a container are sorted by **ascending** order value and then by their source code order

```
.box {  
  order: 1;  
}
```



**FLEX-DIRECTION**  
(WHOLE CONTAINER)

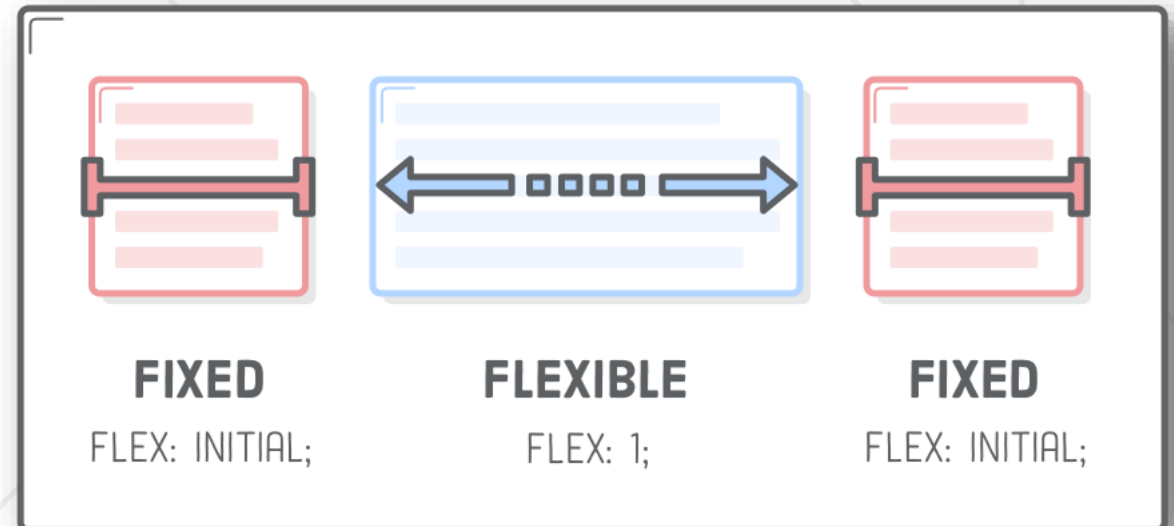


**ORDER**  
(INDIVIDUAL ITEMS)

# Flexbox Properties – Flex Item

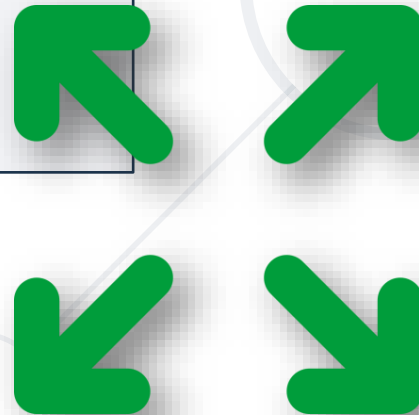
- Flexible sizing of flex items – **flex-grow**
  - Sets the **flex grow** factor, which specifies how much of the flex container's remaining space should be **assigned** to the **flex item's** main size

```
.box {  
  flex-grow: 1;  
}
```



- Flexible sizing of flex items – **flex-shrink**
  - Sets the **flex shrink** factor of a flex item
  - If the size of all flex items is **larger** than the **flex container**, items shrink to fit according to flex-shrink

```
.box {  
  flex-shrink: 1;  
}
```



# Flexbox Properties – Flex Item

- Flexible sizing of flex items – **flex-basis**
  - Sets the initial **main size** of a flex item
  - It sets the size of the content box unless otherwise set with **box-sizing**

```
.box {  
  flex-basis: 20em;  
}
```

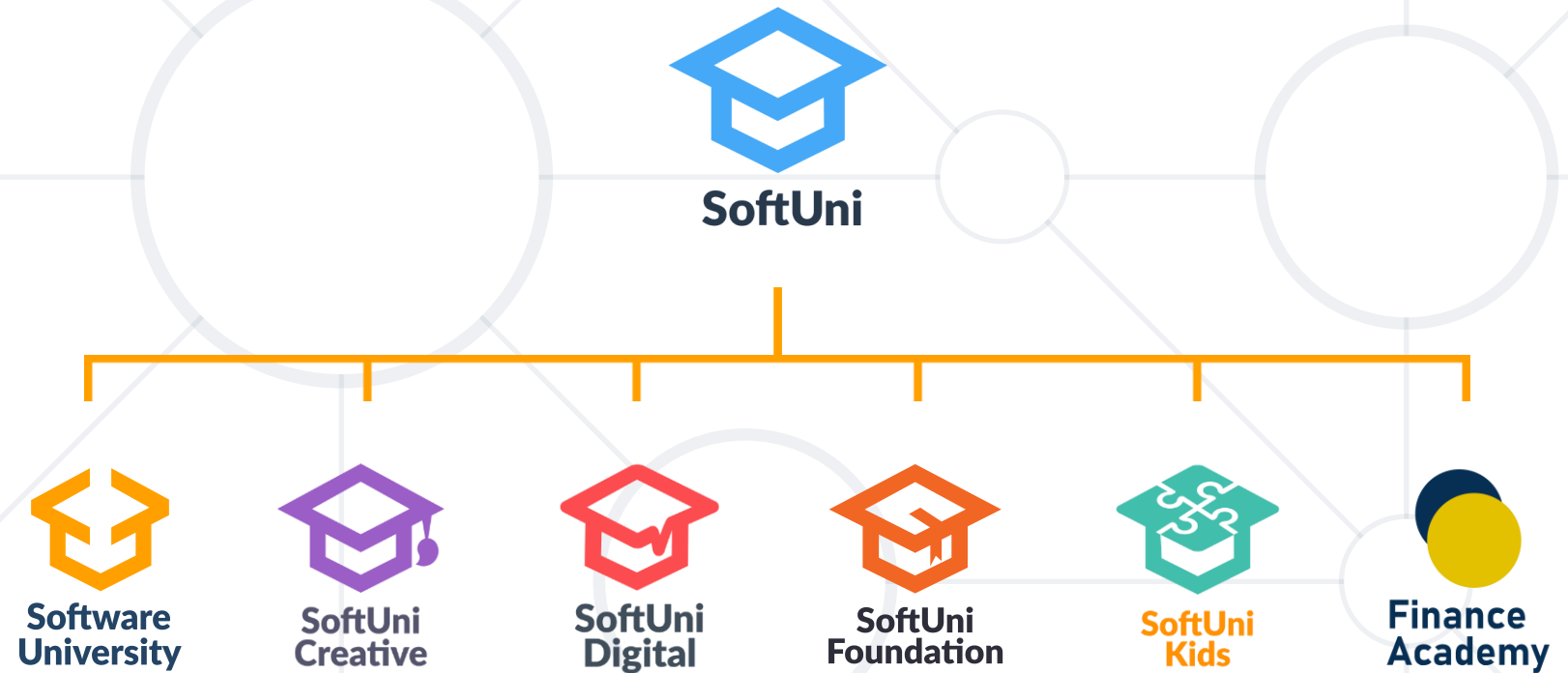


- [https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS\\_layout/Flexbox](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox)
- [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Flexible\\_Box\\_Layout/Basic\\_Concepts\\_of\\_Flexbox](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox)
- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- <https://www.youtube.com/watch?v=hs3piaN4b5I>
- <https://www.flexboxgame.com/>

- **Flexbox**
- **Properties for the Container**
- **Properties for the Children**



# Questions?



# SoftUni Diamond Partners





- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – <https://about.softuni.bg/>
- © Software University – <https://softuni.bg>



- Software University – High-Quality Education, Profession and Job for Software Developers

- [softuni.bg](http://softuni.bg), [about.softuni.bg](http://about.softuni.bg)

- Software University Foundation

- [softuni.foundation](http://softuni.foundation)

- Software University @ Facebook

- [facebook.com/SoftwareUniversity](https://facebook.com/SoftwareUniversity)

- Software University Forums

- [forum.softuni.bg](http://forum.softuni.bg)

