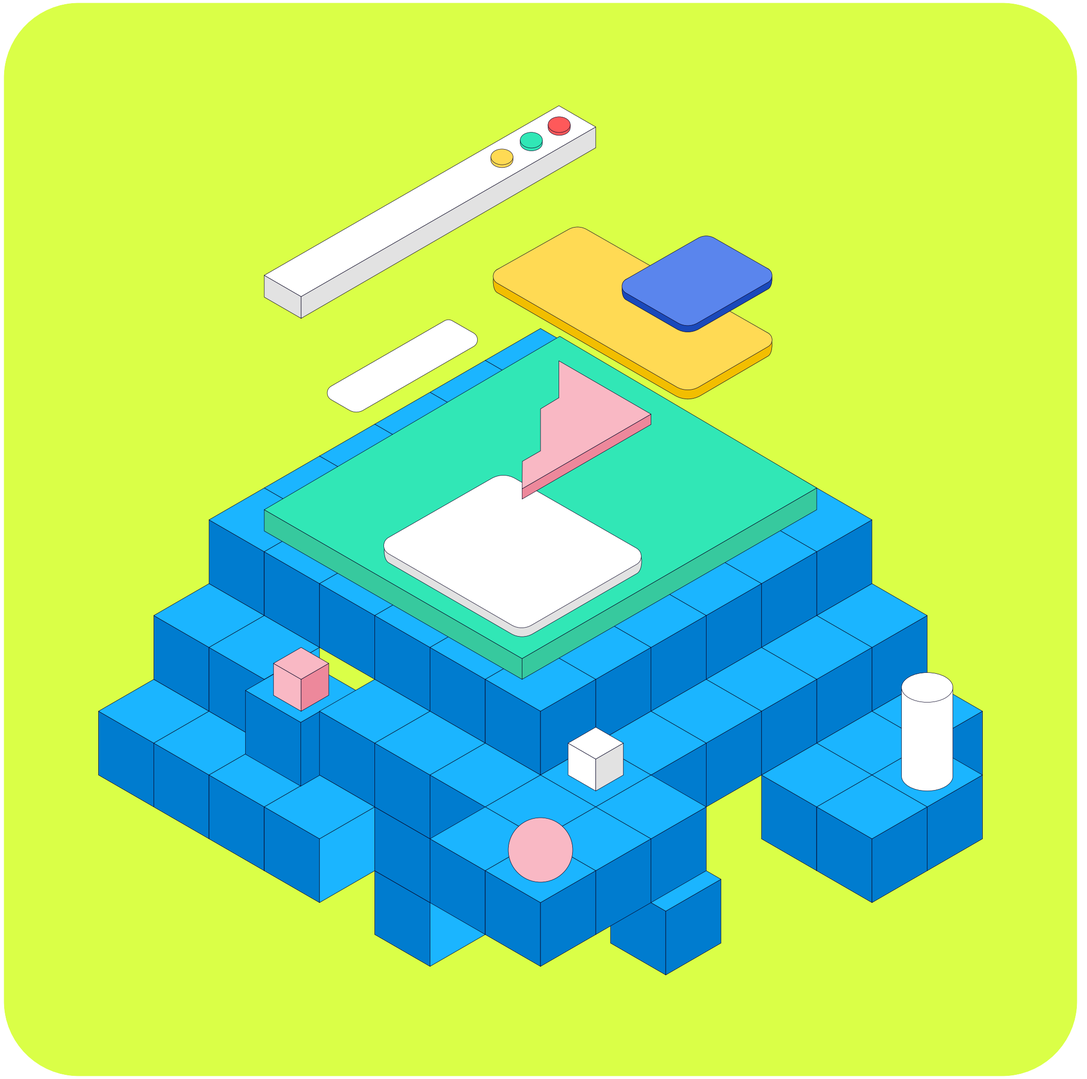
notes F 1.17

Website development



★☆☆☆

Beginner level

Materials prepared by the department
of methodological development department

A website for an Art Studio. Part 2

flex-wrap, flex-grow, align-content, align-self

flex-wrap

flex-wrap is a CSS property that allows to set the wrapping of the elements in a flex-container. The default value is **nowrap**, the elements don't wrap to the next line, they shrink to fit into one line.

flex-wrap has 2 more value:

- **wrap** –wrapping is allowed, the direction is the same as flex-direction.
- **wrap-reverse** – wrapping is allowed, the direction is opposite to flex-direction

Starting to get acquainted with flex-grow

Previously we worked with the properties of the flex-container and now we will get acquainted with one of the **flex element** properties.

flex-grow allows to set the size of the elements more flexibly, in particular it allows setting different sizes to flex-elements.

For example, if the **width** of the element is set to **50%**, it will span 50% of its parent element's width. After adding **flex-grow: 1**, if there's some space in the container, the element will occupy the whole available space.

align-self

align-self can be used when we need to align separate flexbox blocks along the cross axis.

align-self is set to elements and not to the container and can take the same values as align-items:

- **flex-start** - the element is placed at the start of the cross axis
- **flex-end** - the element is placed at the end of the cross axis
- **center** – the element is centered in the cross axis
- **baseline** - the element is aligned along the baseline of the text
- **stretch** - the element stretches to the height of the cross axis

align-content

align-content works just like justify-content, but it aligns rows of elements along the cross axis. It's important to note that this property aligns rows, not separate elements (the difference will be noticeable if a flex-container has more than 1 rows of elements).

align-content is set to a flex-container and takes the same values as justify-content:

- **flex-start** - the rows are stuck to the start of the cross axis
- **flex-end** the rows are stuck to the end of the cross axis
- **center** – the rows are in the center of the cross axis
- **space-between** – the space between the rows is even, the first row and the last row are stuck to the borders.

align-content

- **space-around** – the space between the rows is even there's a half of this space between the first row / the last row and a border
- **space-evenly** – the space between the rows and the space between the first row / the last row and a border is even
- **stretch** – the rows evenly stretch to the whole available space