notes F 1.18

Website development

★☆☆☆

Beginner level

Materials prepared by the department
of methodological development department

A website for an Art Studio. Part 2

flex-shrink, flex-basis, flex, order (2 lessons)

Getting more acquainted with flex-grow

flex-grow can not only specify how much space an element can take inside its parent element, but also set **proportions of the elements relative to each other.**

So, if the **flex-grow** property with the value of 1 is set to all flex container elements, they will have the same size. And if one element has **flex-grow: 1** and another one has the **flex-grow:2**, then the second element will be two times bigger than the first one (provided that there's some empty space for growth in the container).

flex-basis

flex-basis is a property that affects the size of the flexbox elements. It is set to the elements, and is responsible for the **size along the main axis**.

flex-basis can take values in different measurement units, including **px** and **%**:

```
.item {
    flex-basis: 350px; /* the size of the element along the main axis*/
}
```

order

We can use the **order** property when we need to change the order of flex container elements without having to change their order in the HTML code.

We just need to set a unique class to each element and specify the **order** property with the ordinal number of the element.

The default value of the **order** property for each element is **0**. That's why if we need to change the order of only one element, we can specify a negative value to it (order: -1) – then it will be displayed as **the first element**. Or any number that is **greater than 0** – then it will shift **to the end**. Of course, if you change the order property for several elements, you need to make sure that the specified number corresponds to the desired position.

flex-shrink

If you need to increase a flex-element, you can use the flex-grow property. And if, on the contrary, you want to decrease an element, you can use another property – **flex-shrink**.

flex-shrink works similarly to **flex-grow**: you need to pass a number to it that will determine how much the element will shrink relative to the rest of the elements.

```
.item1{
    width: 100px;
    flex-shrink: 1;
}
```

```
.item2{
    width: 100px;
    flex-shrink: 2;
}
```

In the given example, if the width of the container with .item1 and .item2 is less than 200px, then both elements should shrink **on a 2:1 ratio**. For example, if the container will be decreased to **100px**, the first element will be **66.6px** wide and the second one will be **33.3px** wide.

flex

flex is a shorthand property that allows setting the values to the **flex-grow**, **flex-shrink** and **flex-basis** properties in one line:

```
.item {
    flex: 1, 2, 300px;
}
```

This element will have a size of **300px (flex-basis)** along the main axis, and also the properties **flex-grow: 1** and **flex-shrink: 2** will be set to it.