notes F 1.15

Website development

★☆☆☆

Beginner level

Materials prepared by the department
of methodological development department

Flexbox: creating a side navbar for the website about ecology

object-fit

object-fit is a CSS property that allows to choose how the image should be resized to fit into the <img> element, if its size is different from the image's size.

For example to display an image with the dimensions of 500x350 inside an **<img>** element with the width of 300px and the height of 200px we will have to shrink it or crop it.

object-fit takes the following values:

- **fill** – the image is sized to fill the whole element, it can be shrunk or stretched.
- **contain** – the image is scaled to maintain its aspect ratio while fitting inside the <img> element (might not fill the whole element)
- **cover**- the image will fill the whole element while maintaining its aspect ratio (the image will be cropped if necessary).

object-fit

- **none** – no resizing
- **scale-down** – the content is sized as if none or contain were specified, whichever would result in a smaller concrete object size.

object-position

object-position is the second property for working with the content of the <img> tag.

If after using **object-fit** the visible part of the image is cropped and the wrong part of it is displayed, the **object-position** property can help to solve the problem.

This property is similar to the **background-position** property and also accepts the value of the element's side that should be displayed (**left**, **right**, **top**, **bottom**, **center**) or the value of the x-offset and the y-offset, for example, in pixels or in percent:

object-position: left bottom;

object-position: 20% 50%;

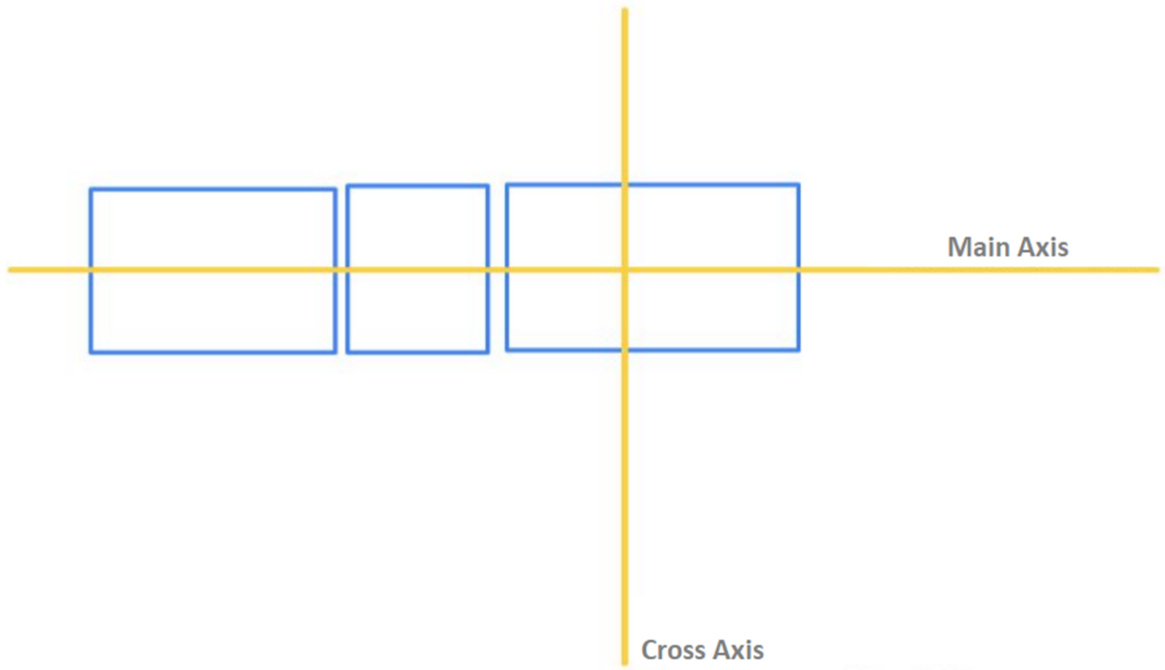
Flexbox

flexbox is a **value** of the display property that lays out the **child elements** of the chosen block (it's called **flex-container**) along two axes - the main axis (from left to right) and the cross axis (from top to bottom).

By default the horizontal axis is the main axis and the elements are laid out along it.

```
.menu {
    display: flex; /* the menu is a flex-container now */
}
```

- Flexbox, the main and the cross axes



flex-direction

flex-direction is a property that sets the direction of the main axis. It's specified for a flex-container and can take 4 values:

- **row** – the main axis is oriented from left to right
- **column** – the main axis is oriented from top to bottom
- **row-reverse** – the main axis is oriented from right to left
- **column-reverse** – the main axis is oriented from bottom to top

```
.menu {
    display: flex; /* the menu is a flex-container now */
    flex-direction: column; /* the child elements are lined up vertically now */
}
```

align-items

align-items is a property that helps to align the elements along the **cross axis**.
It's also specified for the flex-container and can take the following values:

- **flex-start** – the elements are placed at the start of the cross axis
- **flex-end** – the elements are placed at the end of the cross axis
- **center**- the elements are centered in the cross axis
- **baseline** – the elements are aligned such as their baselines align
- **stretch** - the elements stretch to the height of the cross axis.

justify-content

justify-content is a property that sets the way the elements are distributed along the main axis of the container. Note that before setting this property, you need to set the necessary size and margin to the elements.

justify-content is set to the flex-container and can take the following values:

- **flex-start** – the elements are packed toward the start of the main axis.
- **flex-end** – the elements are packed toward the end of the main axis.
- **center**- the elements are centered along the the main axis
- **space-between** – the elements are evenly distributed in the line; first item is on the start line, last item on the end line

justify-content

- **space-around** – the elements are distributed evenly along the main axis, the distance between the outermost elements and the border of the container is equal to a half of distance between adjacent elements.
- **space-evenly** - the elements are evenly distributed along the main axis, the distance between the adjacent elements and an element and a border is even.