SOCIAL HACKERS ACADEMY

HTML-CSS

April 5th, 2018

Who we are

Aristeidis Bampakos (Aris)

Front End Web Developer @ Plexscape

@aris

@abampakos

https://github.com/bampakoa

George Sisko (Jkr)

.NET Software Engineer @ INTRASOFT International

@George Sisko(Jkr)

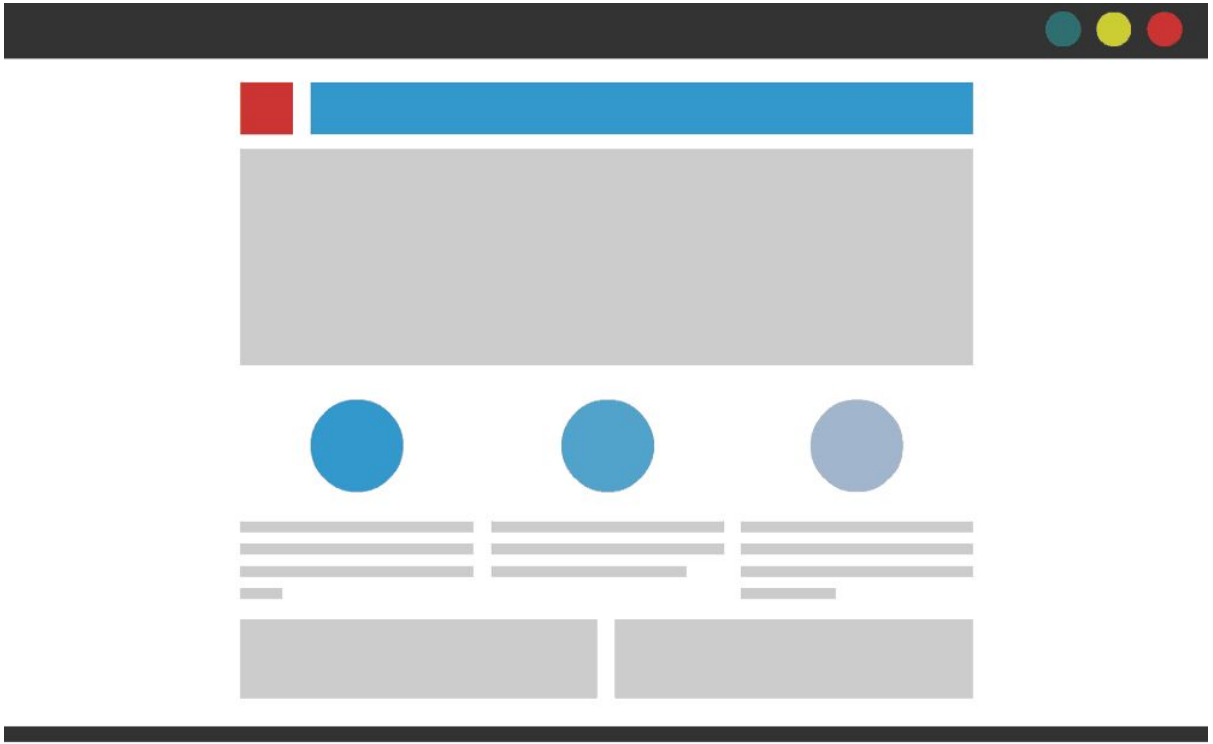
@GeorgeJkrr

https://github.com/jkrgS

Agenda

- Responsive web development
- Units of measurement
- Media queries
- Positioning elements
- Flexbox design

Fixed layout



- Fixed width in pixels (**px**).
- Container of website does not move.
- Width stays same independently of screen size and resolution.
- Bad UX due to horizontal scrolling.
- Full control over the UI.

Fluid layout

- Width in percentages (%).
- Screen size changes, proportion of elements not.
- Narrow columns in smaller screens (images, video).
- Too crowded on small screens, too empty on big ones.

Adaptive layout

- Different versions of the layout based on the screen size.
- Several fixed layouts for each screen size.
- Optimal UX.
- Loads fast.
- Need to design each layout separately (time consuming).

Responsive layout

- Combination of fluid and adaptive.
- Elements stretch/shrink accordingly (**breakpoints**).
- SEO friendly.
- Same UX across screens and devices.
- Extra design effort than adaptive.

Viewport

- It is the area of a web page that is visible from the user. Varies according to the device.
- `<meta name="viewport" content="width=device-width, initial-scale=1.0" />`

Without the viewport meta tag

With the viewport meta tag

Measurement units

- Define the length of an element through properties such as size, margin and padding.

```
<img width="300px" height="200px" />

<p style="margin: 2em;">This is a paragraph</p>
```

- The unit can be omitted when the length is **0**.

```
<h1 style="padding: 0;">This is a heading</h1>
```

- Negative lengths are allowed in some cases.

Types of units

Absolute

- Fixed length.
- Appear exactly the size of the length.
- **cm:** centimetres
- **mm**: millimetres
- **in**: inches
- **px**: pixels

Relative

- Relative to the length of another property.
- Most frequently used.
- **%**: percentage, relative to the same property of the parent element.
- **em**: relative to font size of the element.
- **rem**: relative to font size of the root element.
- **vw**: relative to 1% of viewport width.

Media queries

- Apply different CSS styles depending on the type of a device or its specifics characteristics.

Media queries

Type

- Targets the type of the device.
- **all:** targets all devices.
- **screen:** intended for color computer screens.

Feature

- Targets the specific characteristic of a device.
- Range features (**“min-”, “max-”**).
- If a feature does not apply to a device, the styles of that feature will be ignored and never be applied.

Breakpoints

- The point at which the content of your site will respond to provide the user with the best possible layout to consume the information.
- Introduce them gradually.
    - Design for the smallest viewport first.
    - Expand the view at the point at which the design seems broken.
    - Add a media rule.
- Common device breakpoints.
- *Consider to always design for mobile first*.

Positioning

- Positioning allow us to override the basic document flow behaviour, to produce interesting effects.
- There are a number of different types of positioning that you can put into effect on HTML elements. To make a specific type of positioning active on an element, we use the “position” property.
    1. Absolute
    2. Relative
    3. Fixed

Absolute/Relative/Fixed

Absolute

We can create isolated UI features that don't interfere with the position of other elements on the page — for example popup information boxes and control menus, rollover panels, UI features that can be dragged and dropped anywhere on the page.

Relative

With Relative positioning you have to modify its final position , including making overlap other elements on the page. That elements are top, bottom, left and right.

Fixed

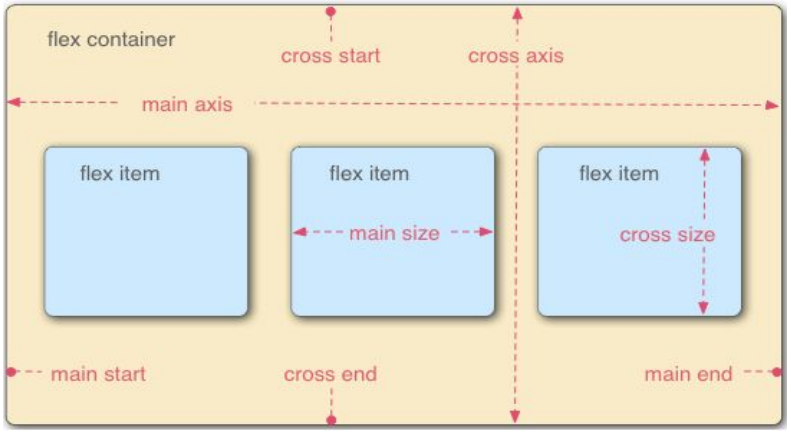
This works in exactly the same way as *Absolute* positioning, with one key difference — whereas absolute positioning fixes an element in place relative to the *<html>* element or its nearest positioned ancestor, fixed positioning fixes an element in place relative to the browser viewport itself.

Flexbox

- Vertically centering a block of content inside its parent.
- Making all the children of a container take up an equal amount of the available width/height, regardless of how much width/height is available.
- Making all columns in a multiple column layout adopt the same height even if they contain a different amount of content.

Flex Model

- The parent element that has display: flex set on it, is called the **flex container**.
- The **main axis** is the axis running in the direction the flex items are being laid out in (e.g. as rows across the page, or columns down the page.) The start and end of this axis are called the **main start** and **main end**.
- The **cross axis** is the axis running perpendicular to the direction the flex items are being laid out in. The start and end of this axis are called the **cross start** and **cross end**.
- The items being laid out as flexible boxes inside the flex container are called **flex items** .



Questions?

Thank you!

Aristeidis Bampakos

@abampakos

George Sisko

@GeorgeJkrr