

# **CODING 101**

## **Web Development Essentials**

**HTML, CSS, Javascript**



# Today's Agenda

- Pseudo-classes & Elements
- CSS Box Model
- Media Queries
- FlexBox



# What are Pseudo-classes?

A ***Pseudo-class*** is a keyword added to a selector that specifies a special state of the selected element(s). For example, **:hover** can be used to change a button's background color when the user's pointer hovers over it.

```
Button: hover {  
    background-color: #e3e3e3;  
    font-weight: bold;  
}
```



## Some popular pseudo-classes

SELECTOR	DESCRIPTION
:hover	Selects elements on mouse over
:focus	Selects the <i>input</i> element that has focus
:first-child	Selects every elements that is the first child of its parent
:last-child	Selects every elements that is the last child of its parent
:nth-child(n)	Selects every element that is the n child of its parent



# What are pseudo elements?

A **pseudo-element** is a keyword added to a selector that lets you style a specific part of the selected element(s). For example, **::first-line** can be used to change the font of the first line of a paragraph.

```
p::first-line {  
  color: #ff0000;  
}
```

```
p::first-letter {  
  font-size: 2rem;  
}
```



# All pseudo elements

SELECTOR	DESCRIPTION
::after	Insert something after the content an element
::before	Insert something before the content an element
::first-letter	Selects the first letter of each <p> element
::first-line	Selects the first line of each <p> element
::marker	Selects the markers of list items
::selection	Selects the portion of an element that is selected by a user



# CSS Box Model

When laying out a document, the browser's rendering engine represents each element as a rectangular box according to the standard **CSS basic box model**. CSS determines the size, position, and properties (color, background, border size, etc.) of these boxes.



# The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.

- ❖ **Content:** The content of the box, where text and images appear
- ❖ **Padding:** Clears an area around the content. The padding is transparent
- ❖ **Border:** A border that goes around the padding and content
- ❖ **Margin:** Clears an area outside the border. The margin is transparent







# Media Queries

Media queries are useful when you want to modify your site or app depending on a device's general type (such as print vs. screen) or specific characteristics and parameters (such as screen resolution or browser viewport width).

```
@media screen and (max-width: 460px){  
  body{  
    background-color: #e3e3e3;  
  }  
}
```



# CSS FlexBox Module

The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using float or positioning.

To start using the Flexbox model,  
you need to first define:

- ❖ **Flex Container**
- ❖ **Flex Items**



# Flex Container

The flex container becomes flexible by setting the **display** property to **flex**:

Property	Description
<u><a href="#">align-content</a></u>	Modifies the behavior of the flex-wrap property. It is similar to align-items, but instead of aligning flex items, it aligns flex lines
<u><a href="#">align-items</a></u>	Vertically aligns the flex items when the items do not use all available space on the cross-axis
<u><a href="#">display</a></u>	Specifies the type of box used for an HTML element
<u><a href="#">flex-direction</a></u>	Specifies the direction of the flexible items inside a flex container
<u><a href="#">flex-flow</a></u>	A shorthand property for flex-direction and flex-wrap
<u><a href="#">flex-wrap</a></u>	Specifies whether the flex items should wrap or not, if there is not enough room for them on one flex line
<u><a href="#">justify-content</a></u>	Horizontally aligns the flex items when the items do not use all available space on the main-axis



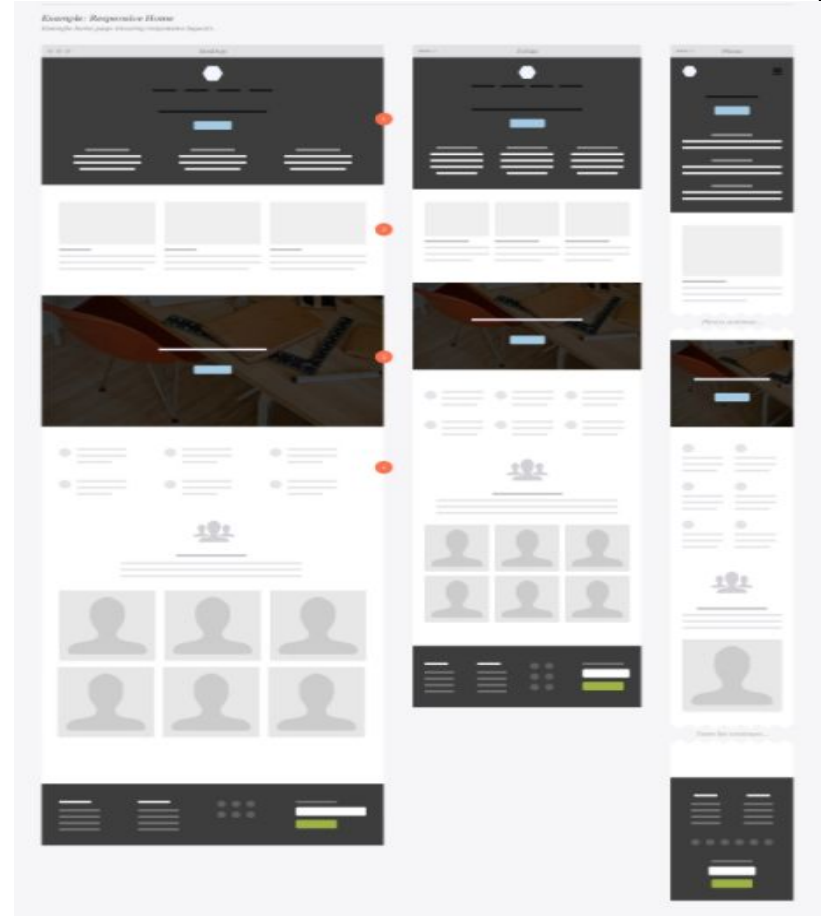
# Flex Items

The direct child elements of a flex container automatically becomes flexible (flex) items.

Property	Description
<u><a href="#">align-self</a></u>	Specifies the alignment for a flex item (overrides the flex container's align-items property)
<u><a href="#">flex</a></u>	A shorthand property for the flex-grow, flex-shrink, and the flex-basis properties
<u><a href="#">flex-basis</a></u>	Specifies the initial length of a flex item
<u><a href="#">flex-grow</a></u>	Specifies how much a flex item will grow relative to the rest of the flex items inside the same container
<u><a href="#">flex-shrink</a></u>	Specifies how much a flex item will shrink relative to the rest of the flex items inside the same container
<u><a href="#">order</a></u>	Specifies the order of the flex items inside the same container

# Challenge 03

- Please use this wireframe to create a responsive home-page.
- Feel free to use any icons and images.
- Source here :  
<https://dribbble.com/shots/1947750-Responsive-Website-Wireframe-Kit/attachments/337158?mode=media>





# Thank you!

Let's get in touch!

- ❖ Discord channels
- ❖ [mostafa.hazareh@theshortcut.org](mailto:mostafa.hazareh@theshortcut.org)