CSS

LET’S MAKE IT LIKE AN ARTIST

INTRODUCTION TO CSS

Cascading Style Sheets

CSS describes the visual style and presentation of the content written in HTML

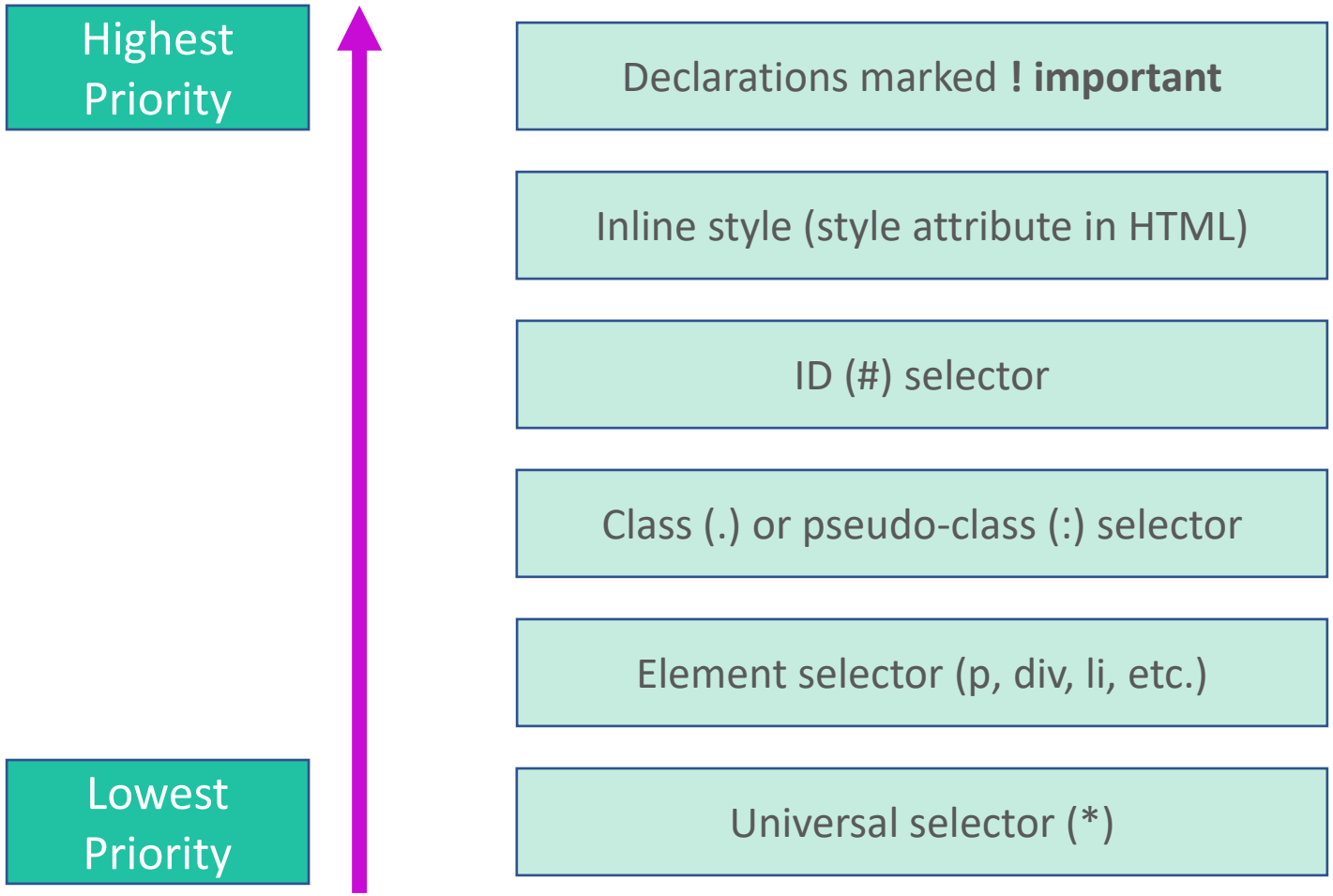
CSS consists of countless properties that developers use to format the content: properties about font, text, spacing, layout, etc.

Web browsers understand HTML and render HTML code as websites

CONFLICTING SELECTORS AND DECLARATIONS

Multiple Selectors

RESOLVING CONFLICTING DECLARATIONS



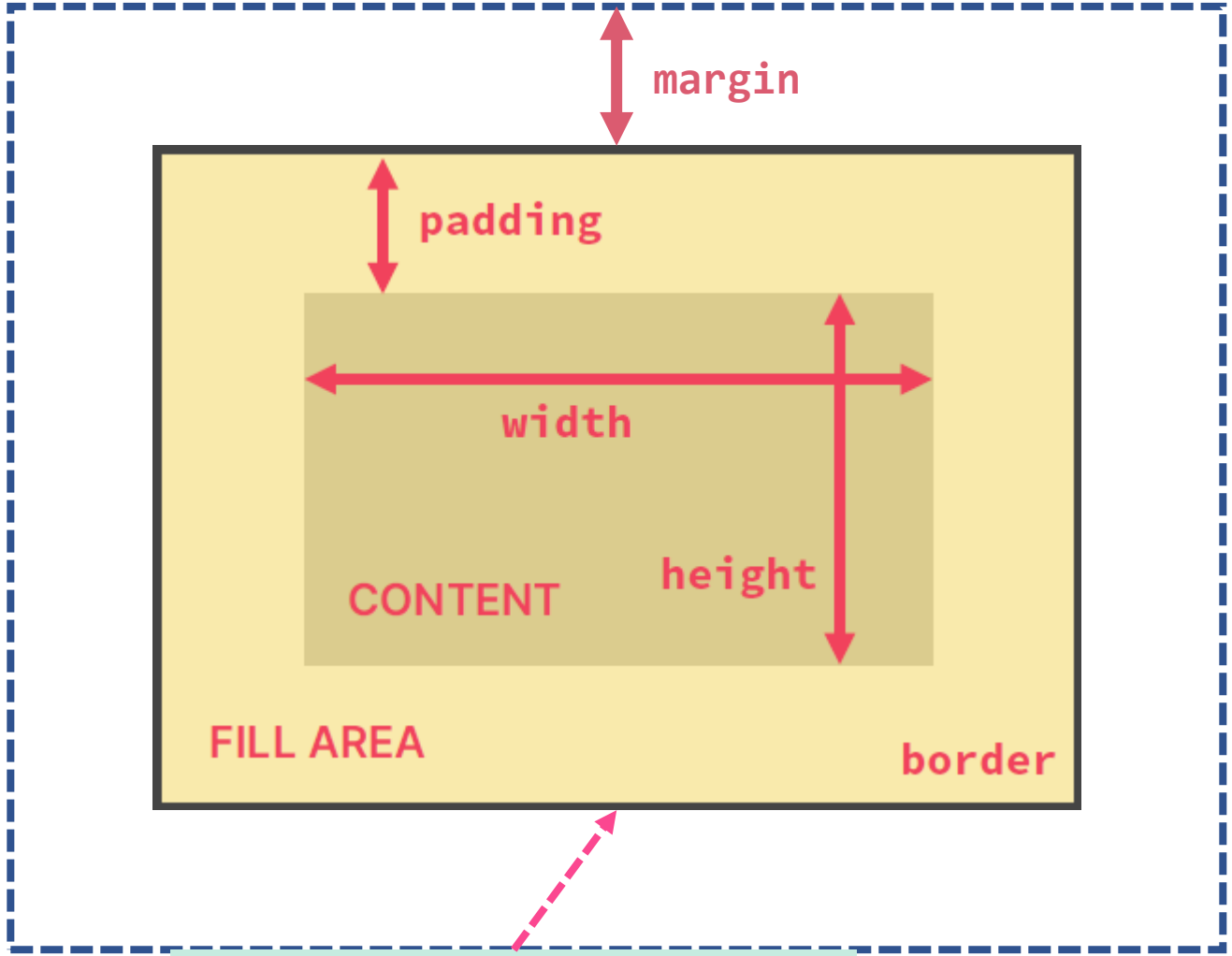
CSS : Inheritance

In CSS, inheritance controls what happens when no value is specified for a property on an element.

CSS properties can be categorized in two types:

Inherited properties	Non-inherited properties
By default, are set to the computed value of the parent element	By default, are set to initial value of the property

THE CSS BOX MODEL



Content: Text, images, etc.

Border: A line around the element, still **inside** of the element

Padding: Invisible space around the content, **inside** of the element

Margin: Space **outside** of the element, between elements

Fill area: Area that gets filled with **background color** or **background image**

BLOCK-LEVEL ELEMENTS

Elements are formatted visually as blocks

Elements occupy 100% of parent element's width, no matter the content

Elements are stacked vertically by default, one after another

The box-model applies as showed earlier

Default elements: `body, main, header, footer, section, nav, aside, div, h1-h6, p, ul, ol, li, etc.`

With CSS: `display: block`

INLINE ELEMENTS

Occupies only the space necessary for its content

Causes no line-breaks after or before the element

Box model applies in a different way: heights and widths do not apply

Paddings and margins are applied only horizontally (left and right)

Default elements: `a, img, strong, em, button, etc.`

With CSS: `display: inline`

WHAT IS RESPONSIVE DESIGN?

Design technique to make a webpage adjust its layout and visual style to **any possible screen size** (window or viewport size)

In practice, this means that responsive design makes websites usable on all devices, such as **desktop computers, tablets, and mobile phones**.

It's a set of practices, **not a separate technology**. It's all just CSS!

RESPONSIVE DESIGN INGREDIENTS

FLUID LAYOUTS

To allow webpage to adapt to the current viewport width (or even height)

Use **%** (or vh / vw) unit instead of px for elements that should adapt to viewport (usually layout)

Use **max-width** instead of width

RESPONSIVE UNITS

Use **rem** unit instead of px for most lengths to make it easy to scale the entire layout down (or up) automatically

Helpful trick: setting 1rem to 10px for easy calculations

FLEXIBLE IMAGES

By default, images don't scale automatically as we change the viewport, so we need to fix that

Always use % for image dimensions, together with the **max-width** property

Use max-width instead of width

MEDIA QUERIES

Bring responsive sites to life!

To change CSS styles on certain viewport widths (called breakpoints)

Use media queries and select breakpoints

DESKTOP-FIRST VS. MOBILE-FIRST DEVELOPMENT

DESKTOP-FIRST

Start writing CSS for the desktop: large screen

Then, media queries shrink design to smaller screens.

MOBILE-FIRST

Start writing CSS for mobile devices: small screen

Then, media queries expand design to a large screen

Forces us to reduce websites and apps to the absolute essentials.

ANIMATION

Animate transitions from one CSS style configuration to another

Animations consist of two components, a style describing the CSS animation and a set of keyframes that indicate the start and end states of the animation's style, as well as possible intermediate waypoints

@keyframes at-rule defines the appearance of the animation

keyframes use a <percentage> to indicate the time during the animation sequence at which they take place. 0% indicates the first moment of the animation sequence, while 100% indicates the final state of the animation

ANIMATION PROPERTIES

Property Name	Description
animation-name	Specifies the name of the @keyframes at-rule describing the animation's keyframes
animation-duration	Configures the length of time that an animation should take to complete one cycle
animation-delay	Configures the delay between the time the element is loaded and the beginning of the animation sequence
animation-iteration-count	Configures the number of times the animation should repeat; you can specify infinite to repeat the animation indefinitely
animation-direction	Configures whether or not the animation should alternate direction on each run through the sequence or reset to the start point and repeat itself
animation-fill-mode	Configures what values are applied by the animation before and after it is executing

REFERENCES

READING MATERIAL

- https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps

VIDEO LINKS

- https://www.youtube.com/playlist?list=PLu0W_9lII9agiCUZYRsvtGTXdxkzPyItg
- https://www.youtube.com/watch?v=1Rs2ND1ryYc&t=2s