



MASYNCTECH  
SCHOOL

# FULLSTACK WEB DEV

# CSS





MASYNCTECH  
SCHOOL

# LEARNING OBJECTIVES



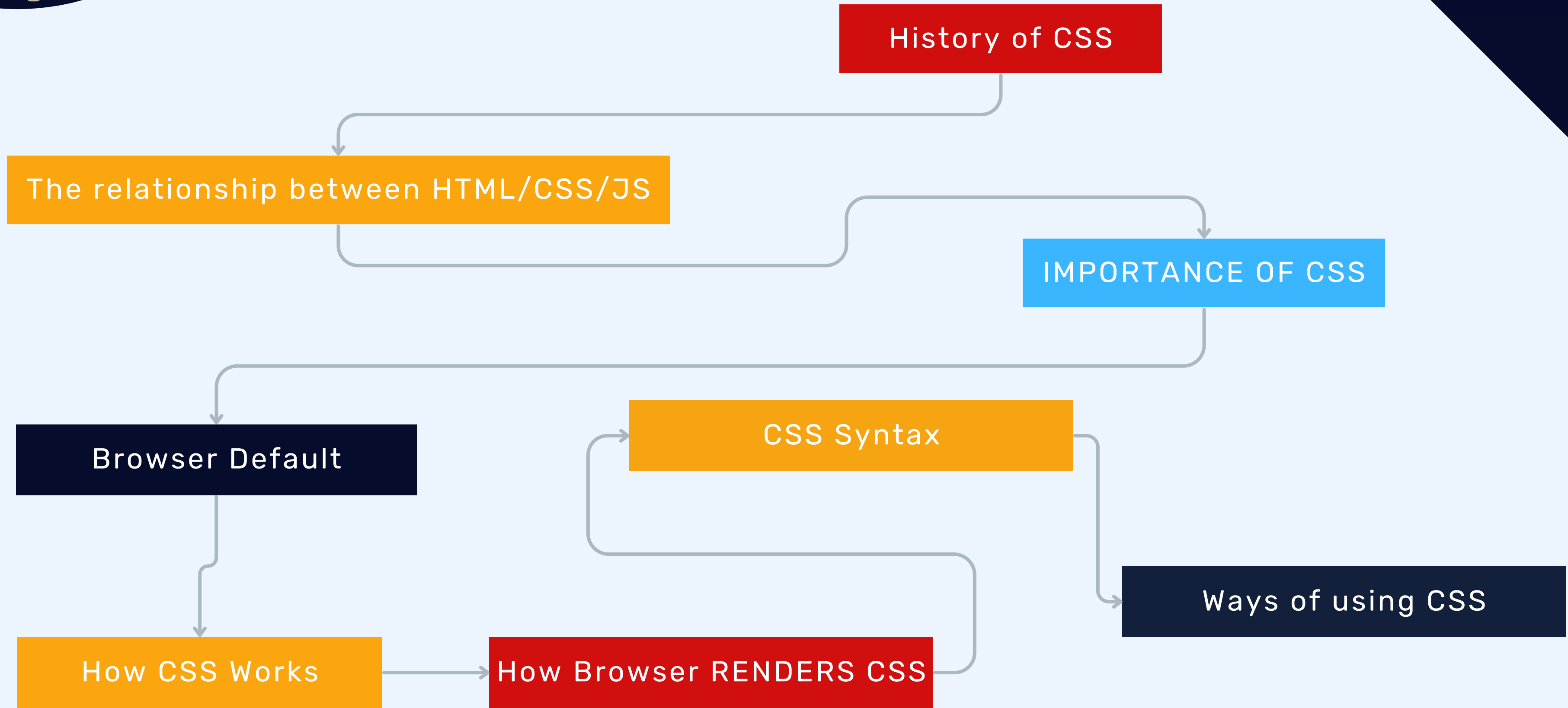
CSS



BASICS



# LEARNING OBJECTIVES





# HISTORY OF

# CSS



BASICS



# HISTORY OF CSS

1

CSS: Style sheet language for web design, alongside HTML and JavaScript

2

CSS proposed '94 to style web pages, addressing HTML's limitation in visual layout

3

Lie proposed CSS to separate presentation from content, enhancing appearance control

4

CSS1 (Dec '96) and CSS2 (May '98) published by W3C, introducing positioning capabilities

5

CSS2.1 fixed errors, clarified specs, and reinstated dropped aspects



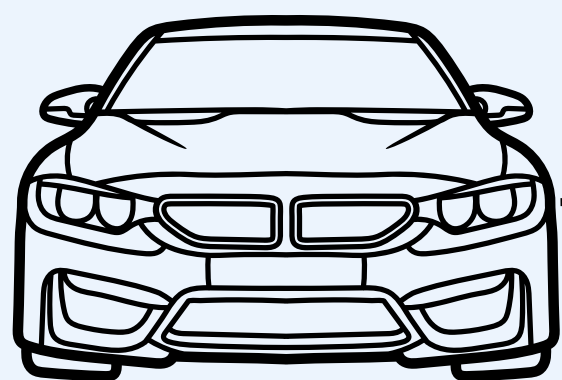
# THE RELATIONSHIP BETWEEN HTML CSS JAVASCRIPT



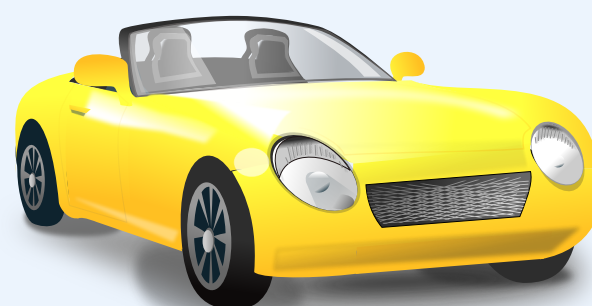
BASICS



# THE RELATIONSHIP BETWEEN HTML, CSS, AND JAVASCRIPT.



HTML  
  
Noun



CSS  
  
Adjective



JS  
  
Verb



# IMPORTANCE OF CSS

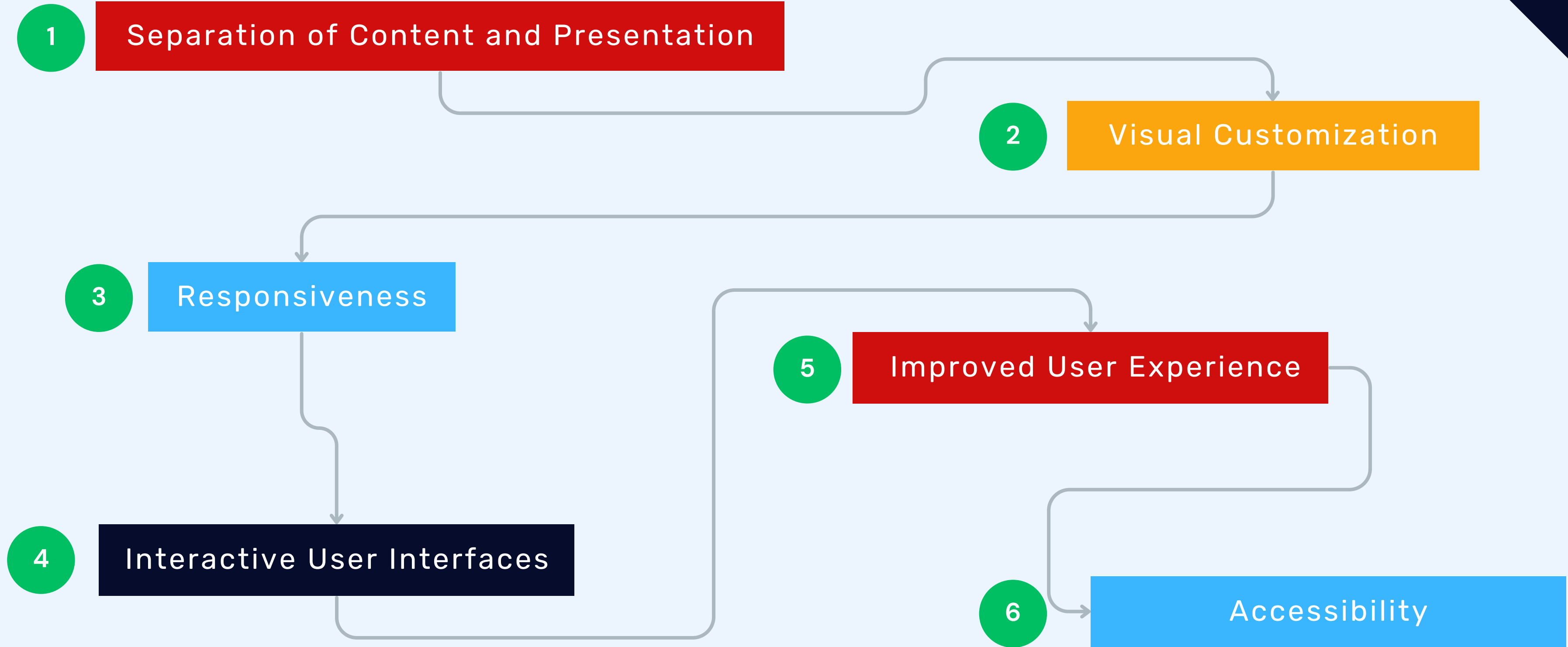


BASICS





# IMPORTANCE OF CSS





# BROWSER DEFAULT STYLESHEET



BASICS



## High-Level Explanation

Browser default stylesheet: Built-in style applied when no custom CSS defined. Provides basic readable layout.

h1

having different font sizes

Links

being blue and underlined

Text inside

`<strong>` tags being displayed in bold





# CODE DEMO



CSS BASICS



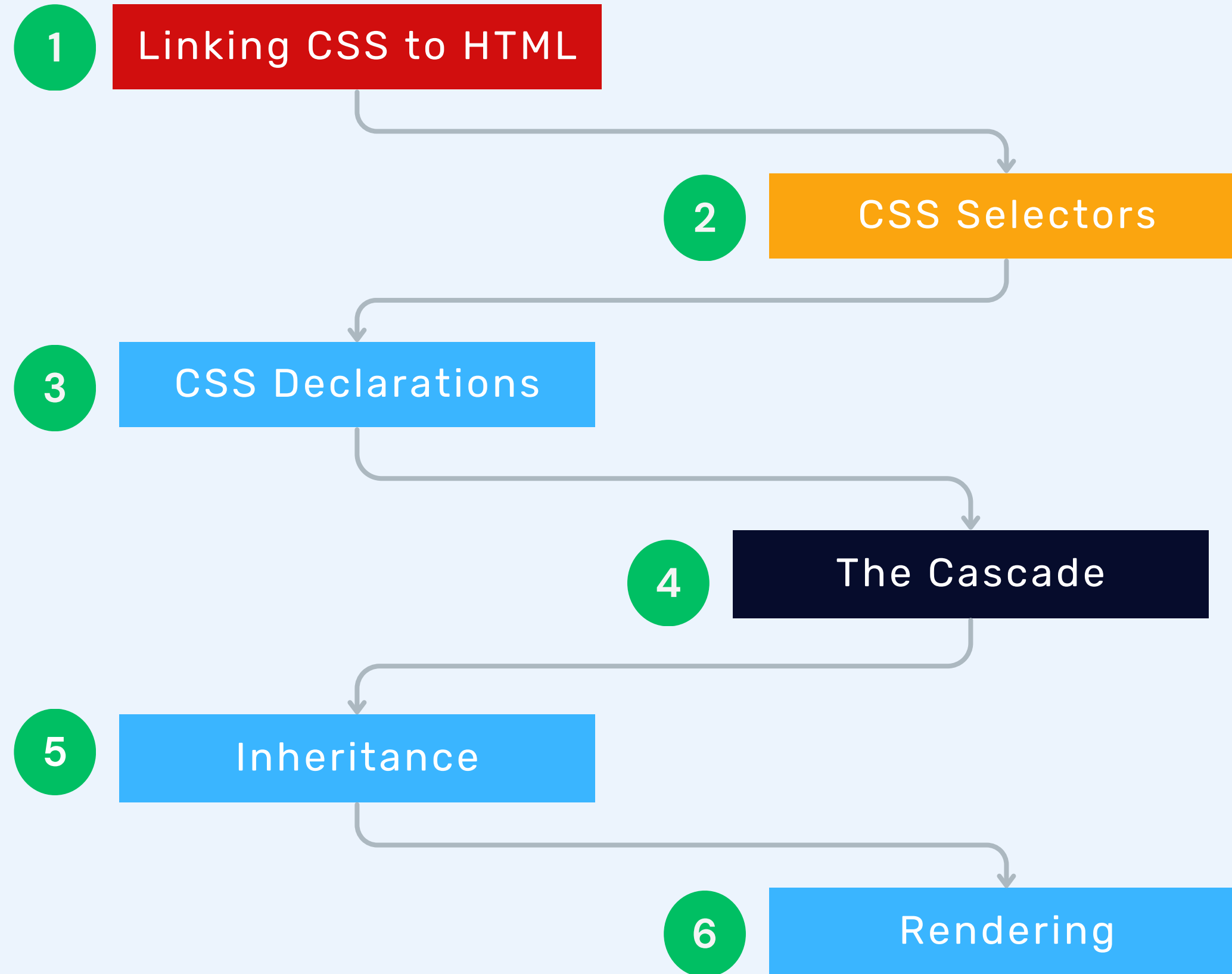
# HOW CSS WORKS



BASICS



# HOW CSS WORKS





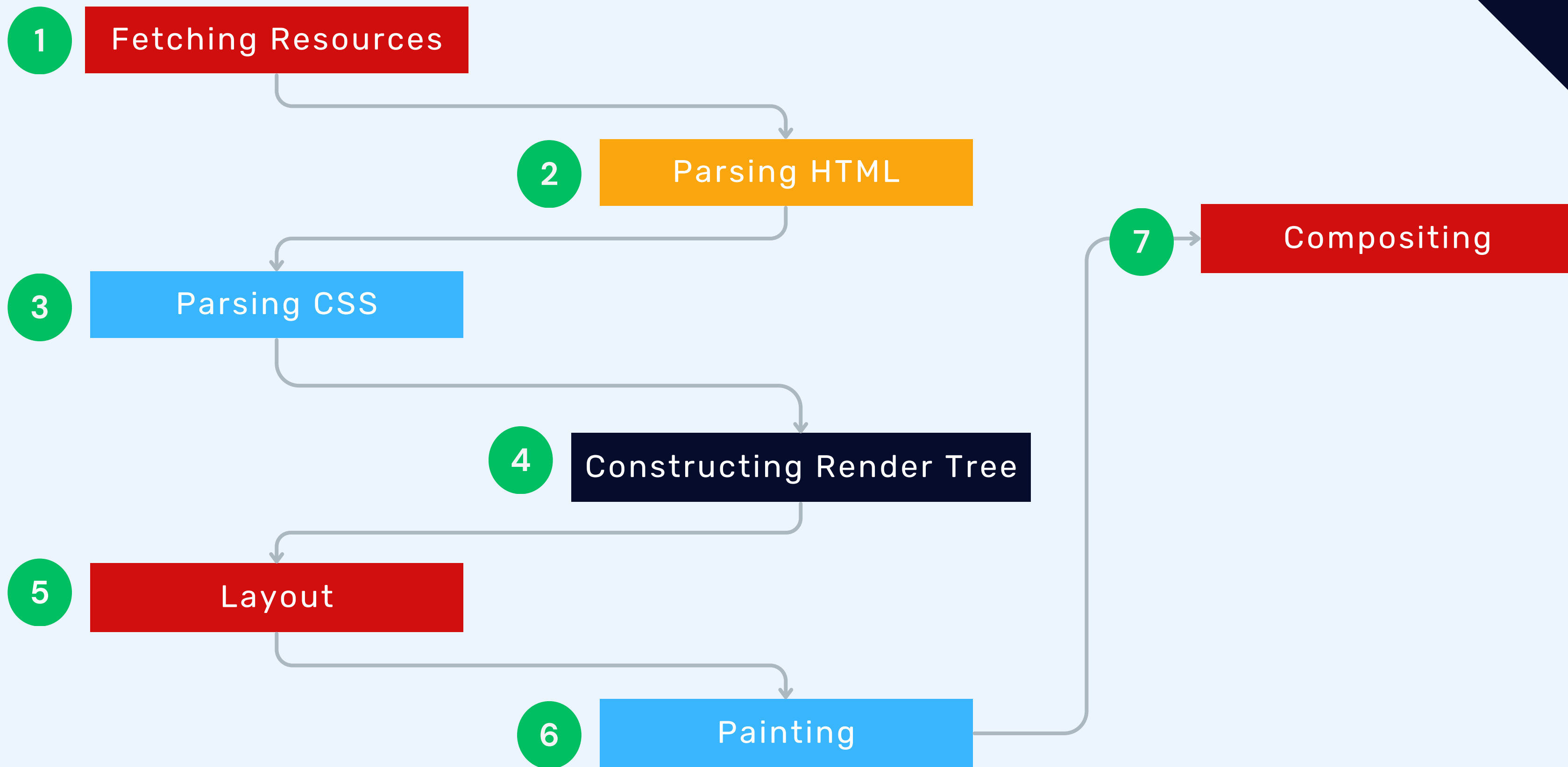
# HOW BROWSER RENDERS CSS



BASICS



# HOW BROWSER RENDERS CSS







# CSS SYNTAX

```
1 selector {  
2     property: value;  
3     property: value;  
4 }  
5
```

CSS



BASICS

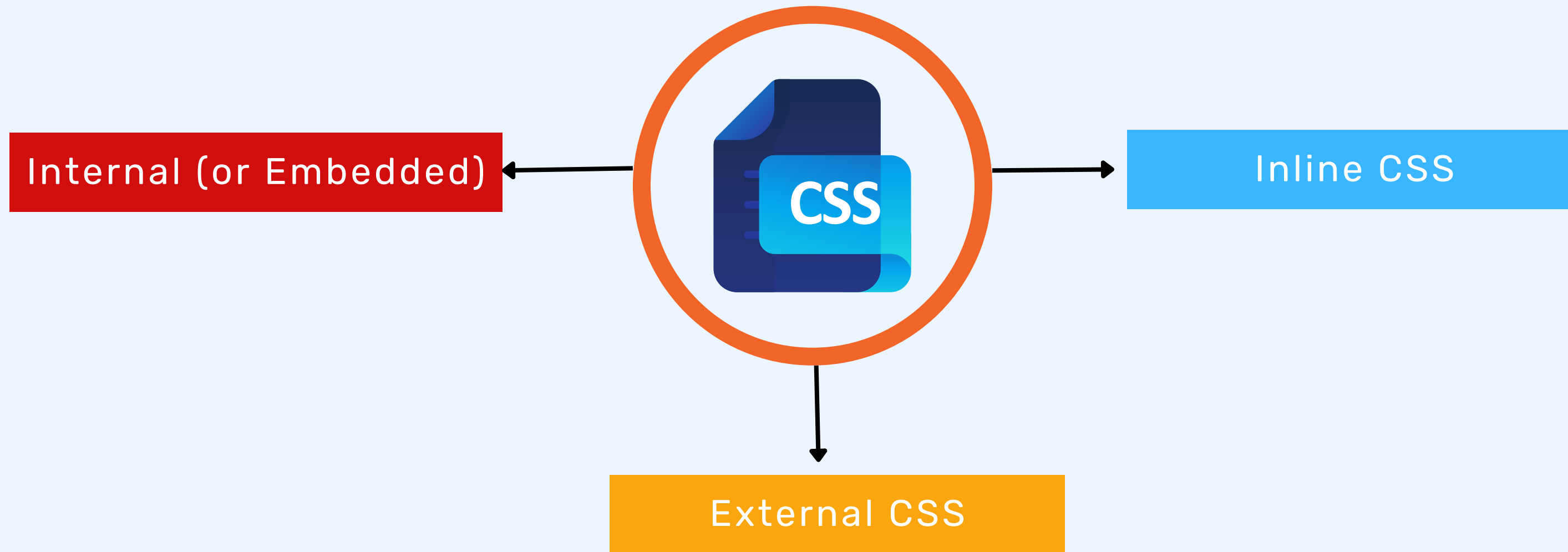


# WAYS OF USING CSS IN HTML





# WAYS OF USING CSS IN HTML



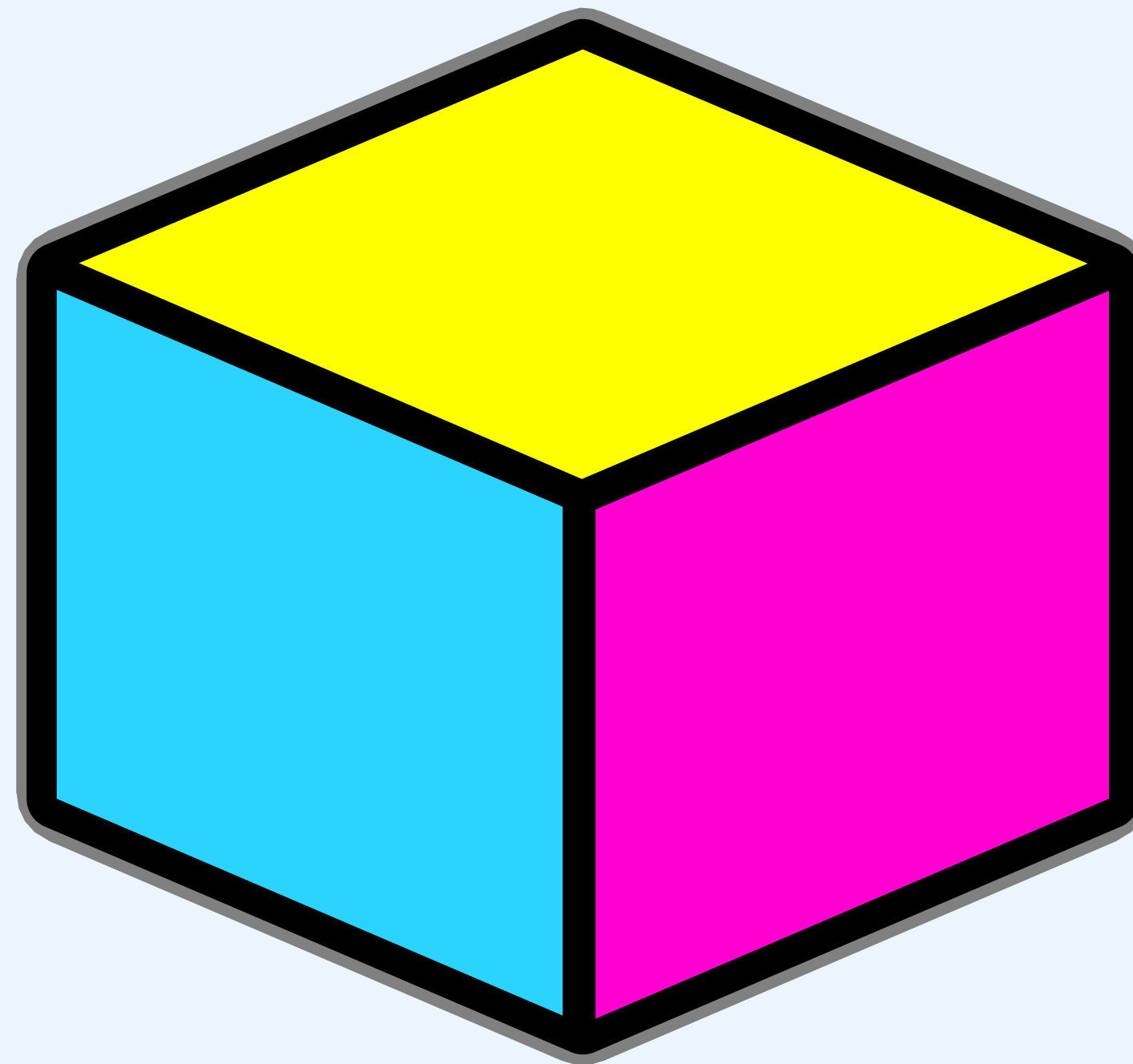


# CODE DEMO





# CSS BOX MODEL





# LEARNING OBJECTIVES



CSS



BASICS



# LEARNING OBJECTIVES

The CSS Box Model concept

Components of the Box Model

Inline vs. Block Elements

width and height of the content area

Definition and usage of padding

The use of 'box-sizing' property

Introduction to border properties

Understanding the content area

Understanding the purpose of margins



# BOX MODEL CONCEPT



CSS



BASICS



# BOX MODEL CONCEPT



## High-Level Explanation

CSS Box Model: HTML elements represented as rectangular boxes. Components: content, padding, border, margin. Define space and interactions

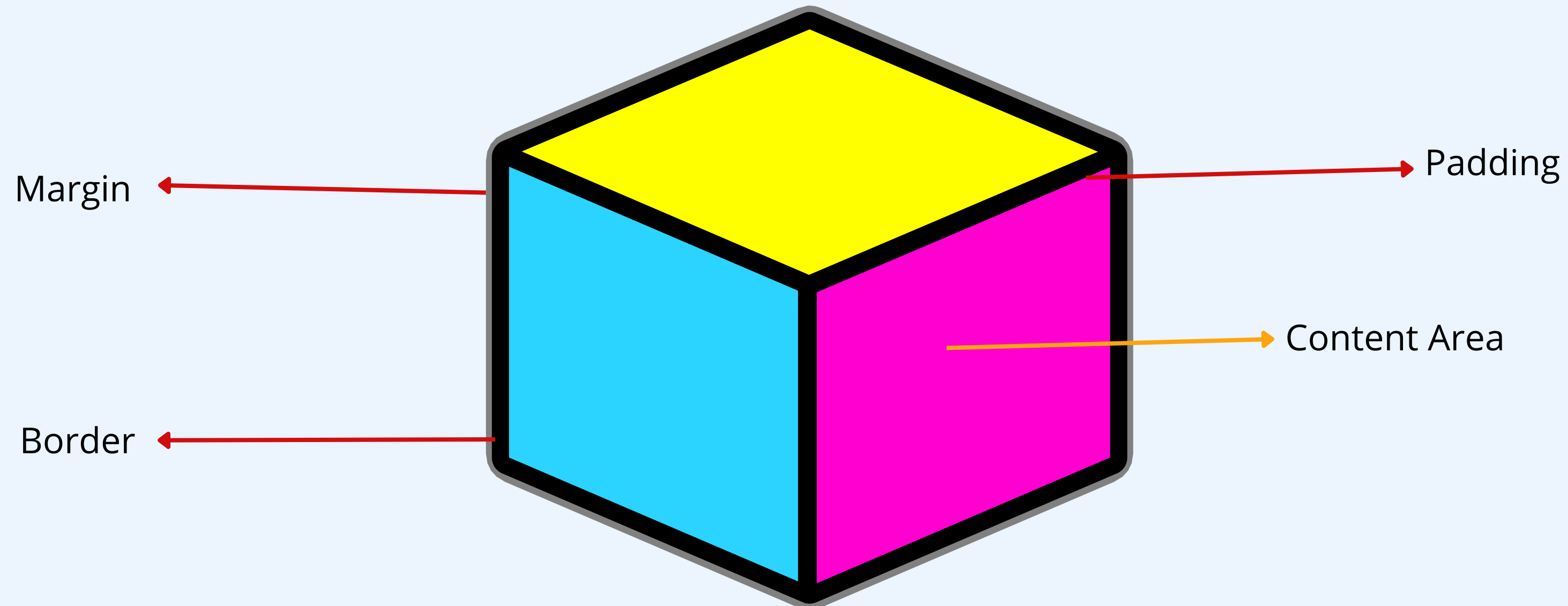


# BOX MODEL CONCEPT



## Deep Dive

CSS Box Model: Content, Padding, Border, Margin. Determines element's layout. Considered for positioning and alignment.





# BOX MODEL CONCEPT



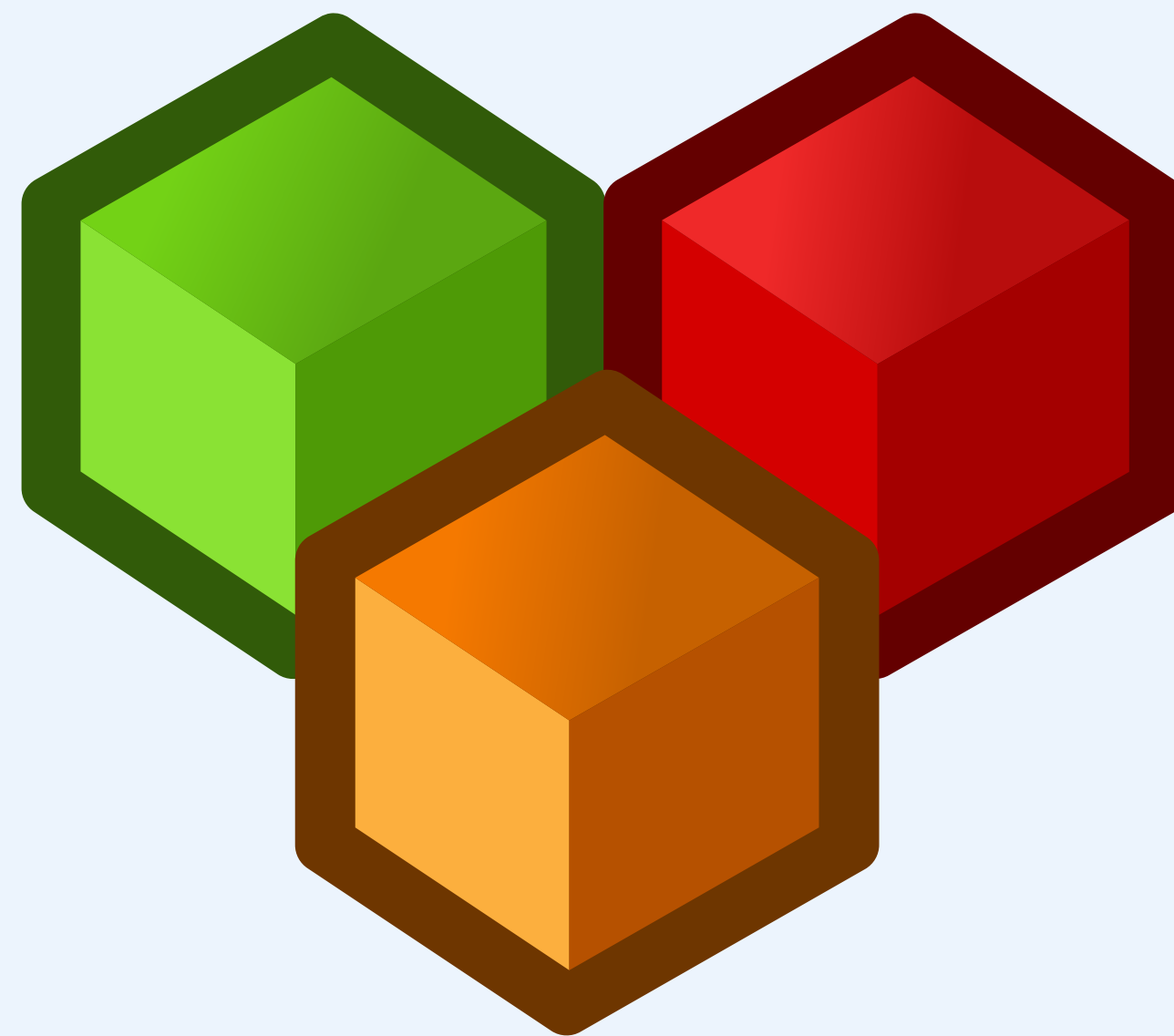
## ELI5 (Explain Like I'm Five) Explanation

CSS Box Model: Blocks represent webpage elements. Content, padding, border, margin ensure layout and spacing.



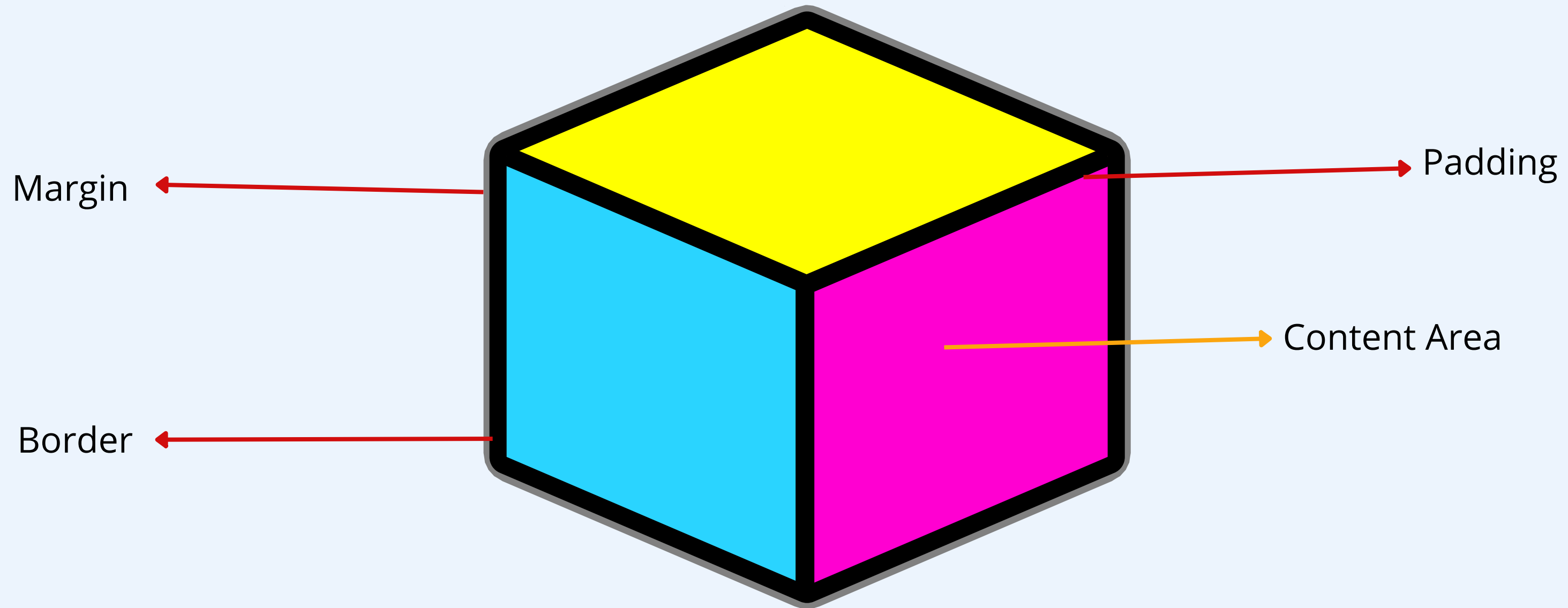


# BOX MODEL COMPONENTS



BASICS

# COMPONENTS OF THE BOX MODEL





HOW TO CONTROL THE WIDTH AND HEIGHT OF

# CONTENT AREA



BASICS



## TIP:

In CSS, you can control the dimensions of the content area using the ``width`` and ``height`` properties. Here's how to do it:



# CODE DEMO







# DEFINITION AND USAGE OF PADDING

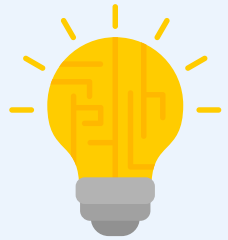


BASICS



## High-Level Explanation

CSS Box Model: Padding creates space between content and border. Enhances legibility and aesthetics



## Usage of Padding

Set element's padding using CSS `padding` property. Units: px, em, rem, %.



# CODE DEMO





# INTRODUCTION TO BORDER PROPERTIES



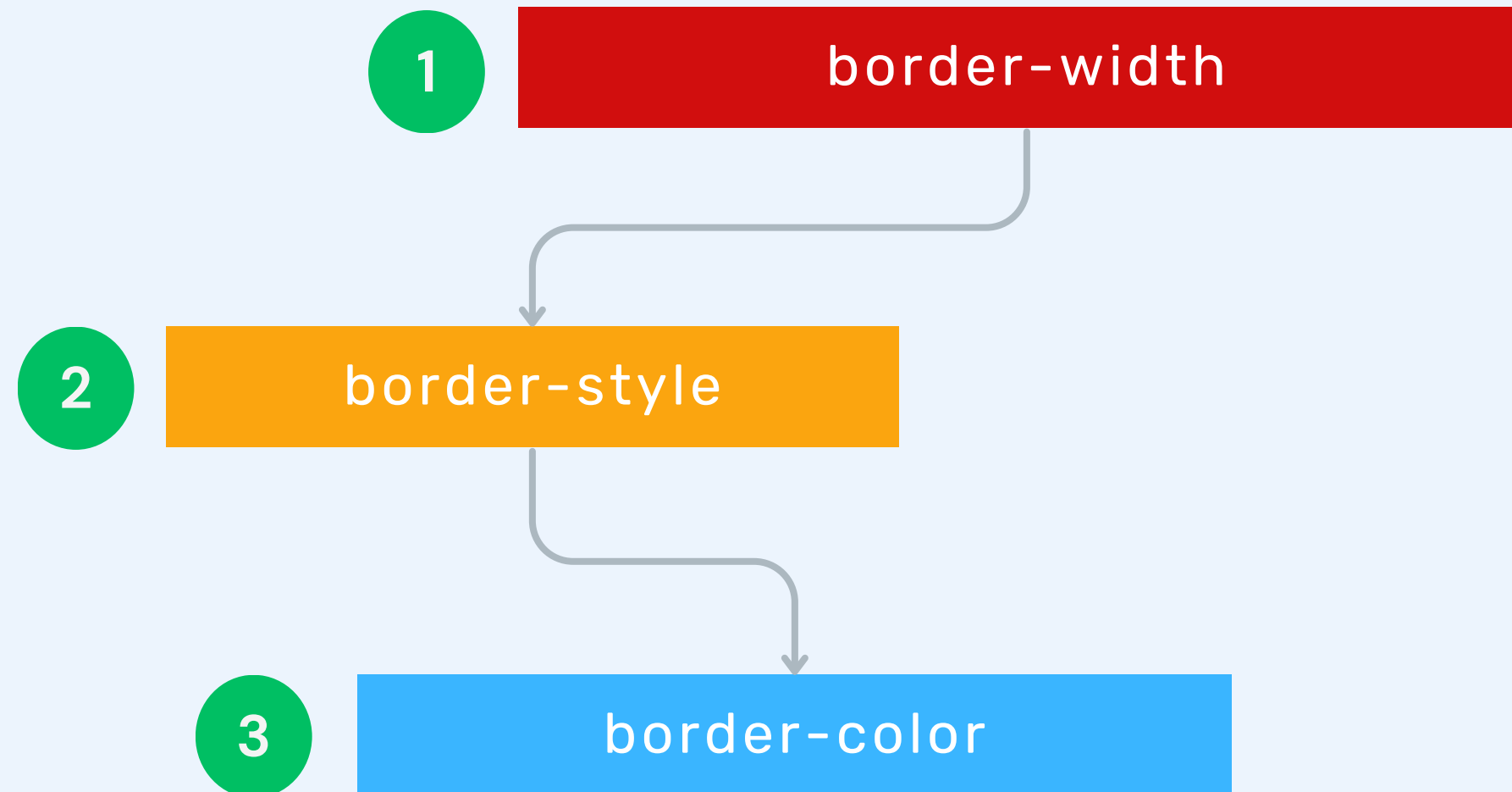
BASICS



# INTRODUCTION TO BORDER PROPERTIES

TIP

CSS border encircles padding and content. Key component of Box Model. Control appearance with CSS properties.





# CODE DEMO



CSS BASICS



# UNDERSTANDING THE PURPOSE OF MARGINS



CSS



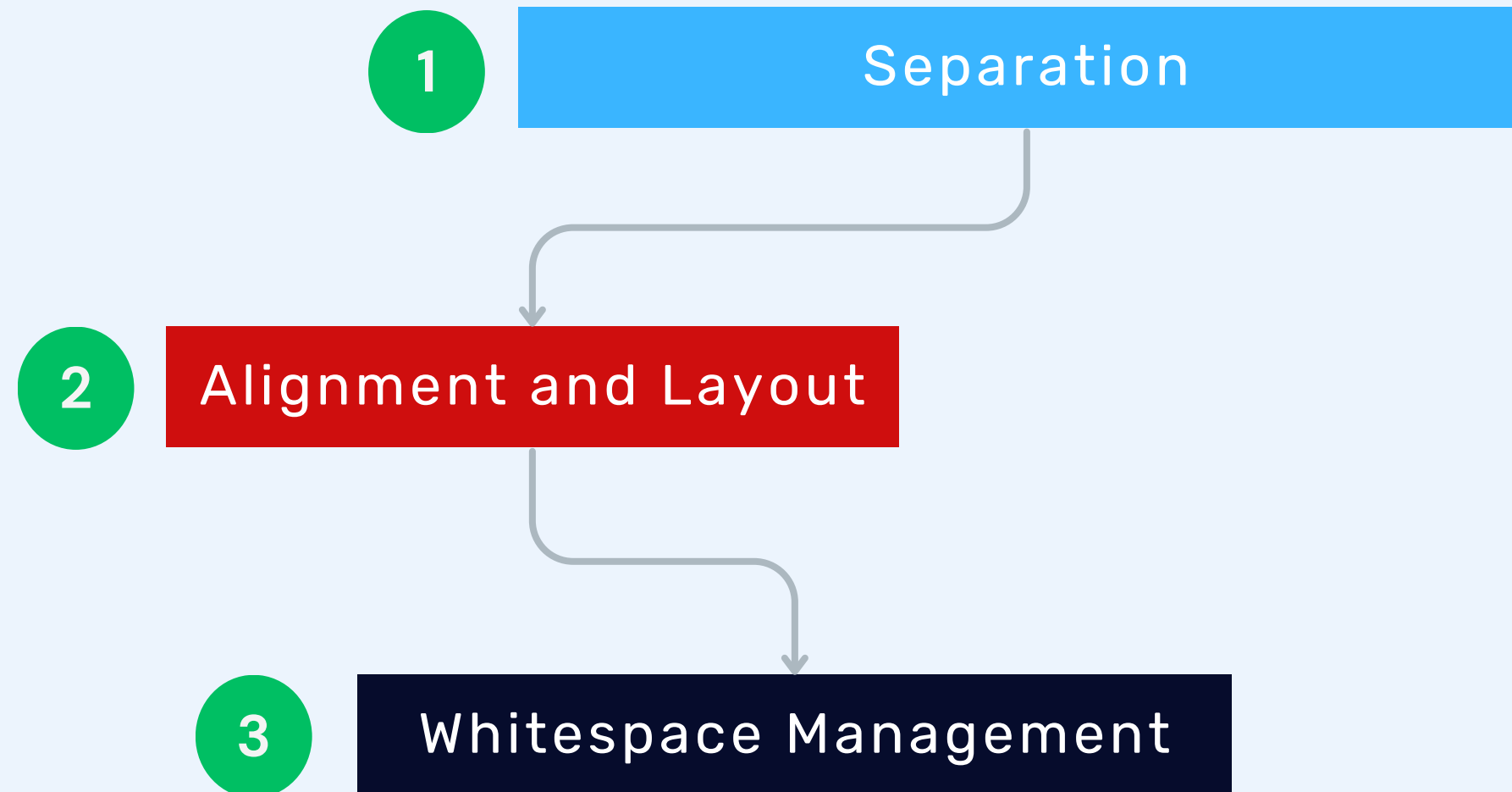
BASICS

# UNDERSTANDING THE PURPOSE OF MARGINS



TIP

CSS margins provide space between HTML elements, controlling layout separation. Integral part of Box Model







# CODE DEMO





# UNDERSTANDING THE CONTENT AREA



BASICS

# UNDERSTANDING THE CONTENT AREA



1

Content area displays content

2

CSS width/height sets content area size.

3

Box size determined by content, padding, border, margin. Total size includes all.

4

CSS properties control content: ``overflow``, ``scroll``, ``clip`` handle overflow

5

box-sizing: border-box includes padding, border in size. Margin not included



# CODE DEMO



CSS BASICS



# UNDERSTANDING THE BOX-SIZING PROPERTY



BASICS



## High-Level Explanation

The `box-sizing` property in CSS controls the box model, which determines how the total width and height of an element are calculated. It has two values: `content-box` (default) and `border-box`.



## Deep Dive

CSS `box-sizing` controls how width and height are calculated: `content-box` (default) excludes padding/border, `border-box` includes them.



## ELI5 (Explain Like I'm Five) Explanation

A toy box represents an element in CSS. The box's outer edge is like the border, inside is the content. Padding is like bubble wrap for toys. With `border-box`, toys and bubble wrap must fit within the box's original size.





## When To use

Use ``border-box`` for maintaining a specific size regardless of padding/border, ideal for fluid layouts and responsive designs. ``content-box`` allows element size to grow based on content, padding, and border.





# CODE DEMO





# UNDERSTANDING INLINE VS BLOCK ELEMENTS



BASICS



# UNDERSTANDING INLINE VS BLOCK ELEMENTS

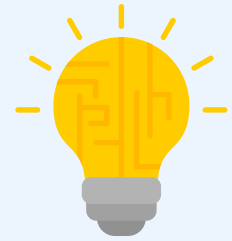


CSS



BASICS

# INLINE VS. BLOCK ELEMENTS



## Deep Dive

Block-level elements create visible blocks on a page, starting on a new line. Examples: ``<div>``, ``<p>``, ``<h1>`` - ``<h6>``, ``<ul>``, ``<li>`. Inline elements don't start on a new line and occupy content width only. Examples: ``<span>``, ``<a>``, ``<img>``, ``<em>``, ``<strong>`. CSS ``display`` property modifies default display behavior, with values like ``block``, ``inline``, ``inline-block``, ``flex`, etc.

```
-----  
| <div> Block |  
-----  
| <p> Block |  
-----  
| <h1> Block |  
-----
```

Block Elements

```
-----  
| <span> | | <img> | | <em> |  
-----
```

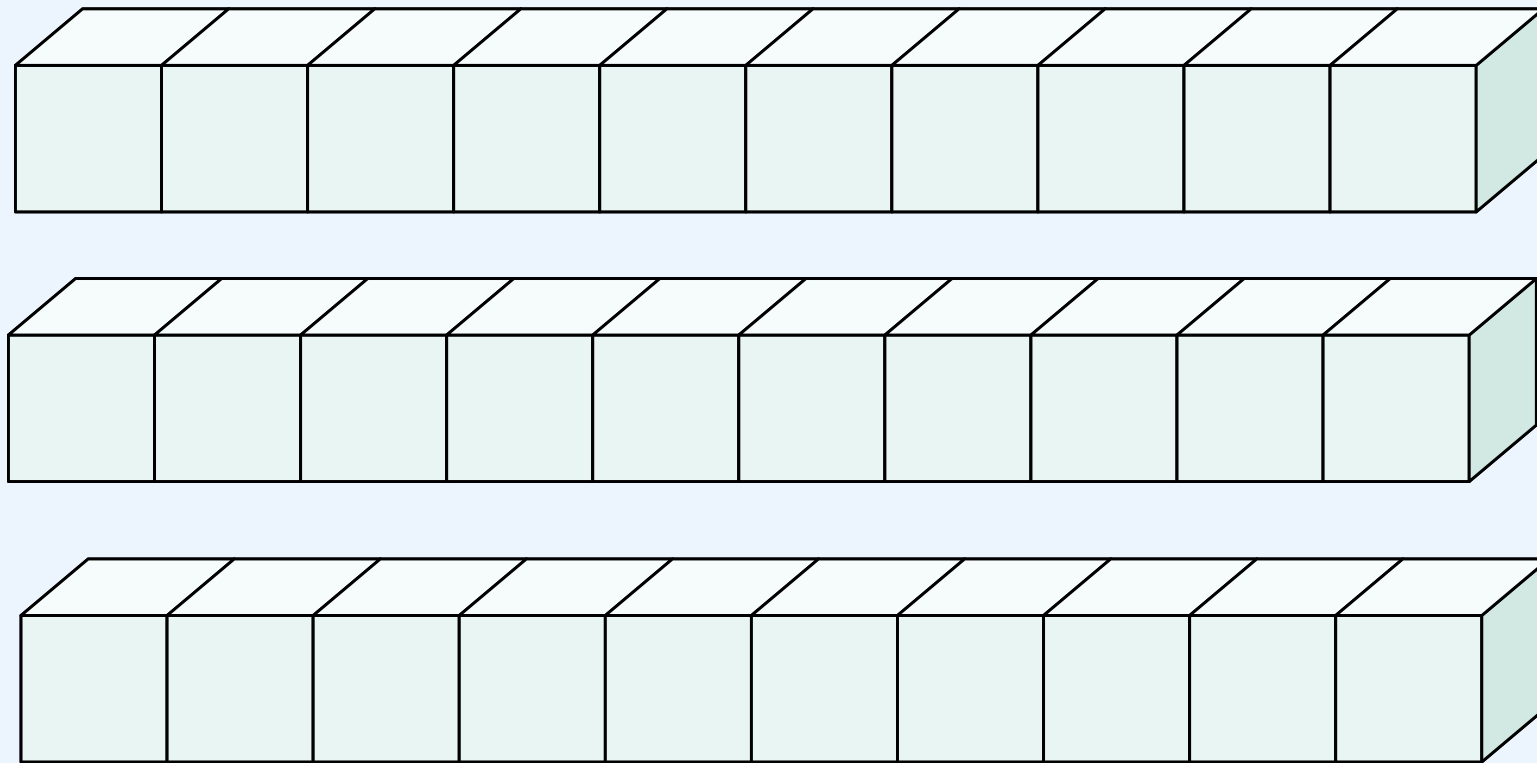
Inline

# INLINE VS. BLOCK ELEMENTS

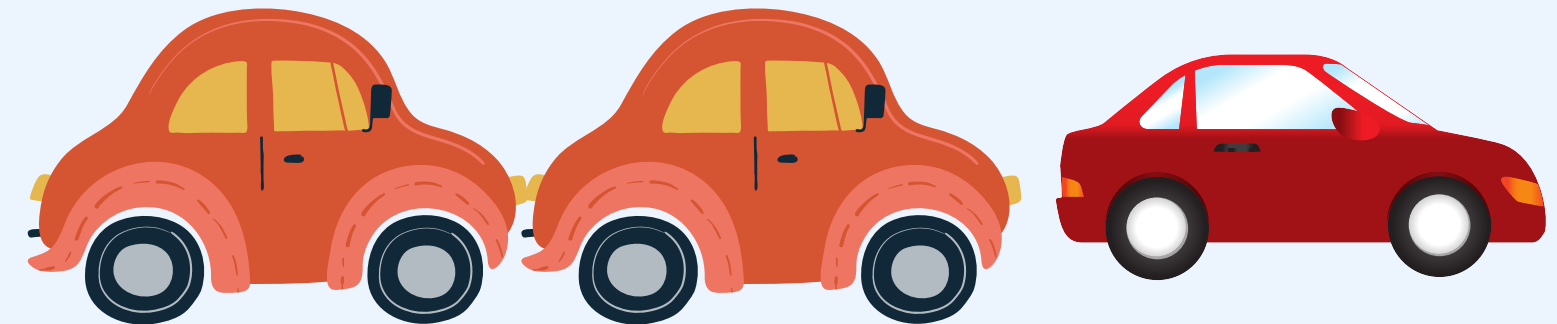


## ELI5 (Explain Like I'm Five) Explanation

Block elements are like big building blocks, each having their own line. Inline elements are like small cars, sitting next to each other on the same line, requiring minimal space.



Block Elements



Inline

# INLINE VS. BLOCK ELEMENTS



## When To use

You would use block elements when you want to structure your content in sections or "blocks," which take the full width of the page. Inline elements are useful when you want to style a part of a text differently, insert an image within a paragraph, or when you want elements like links or buttons to sit next to each other without breaking the line.



# CODE DEMO





# IMPACT OF CSS ON INLINE VS BLOCK ELEMENTS

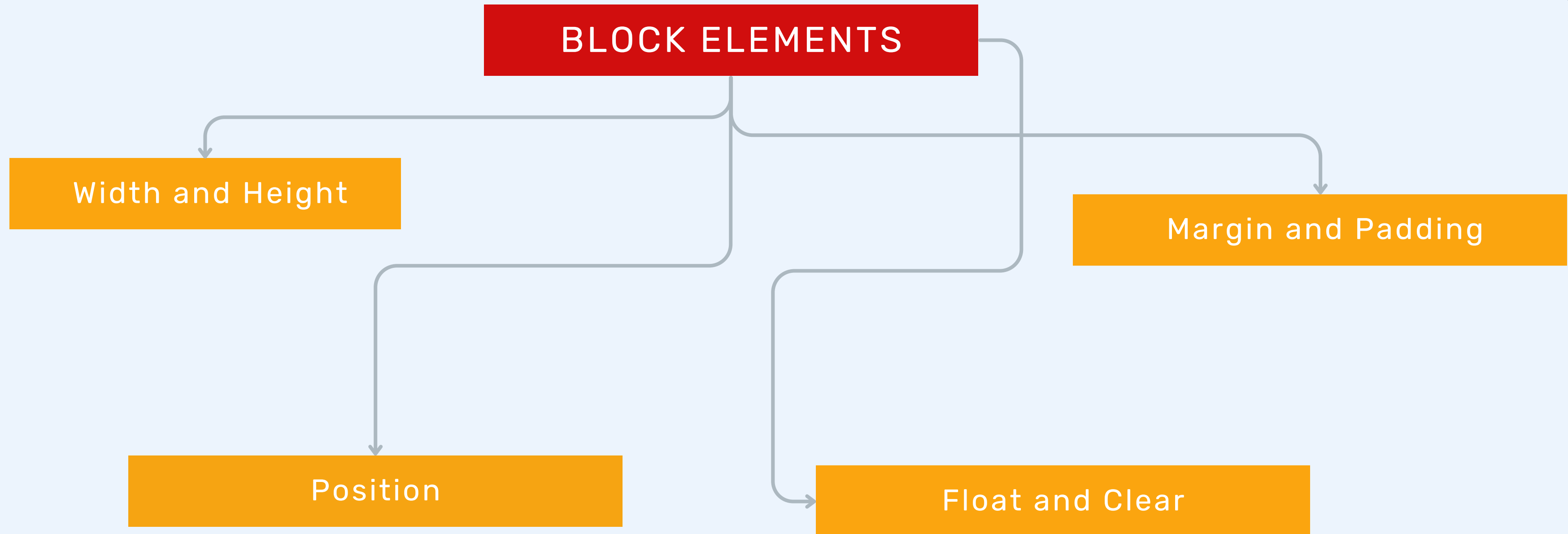


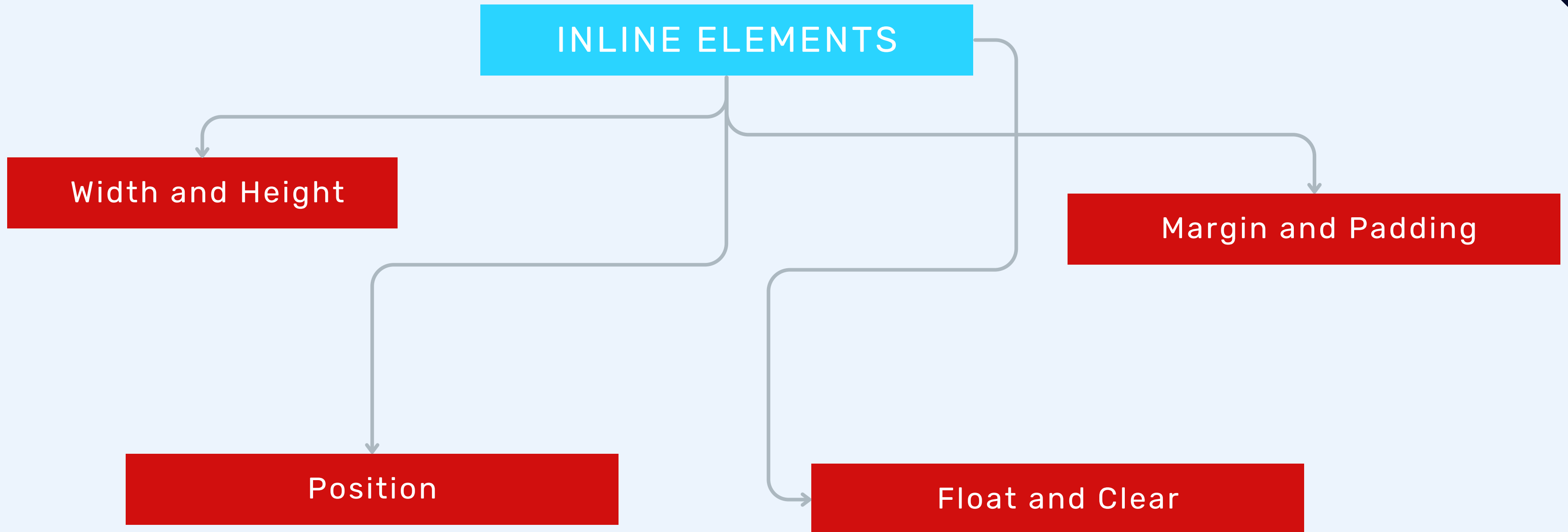
CSS



BASICS









# CODE DEMO

