

# Introduction to CSS3

**CSS**





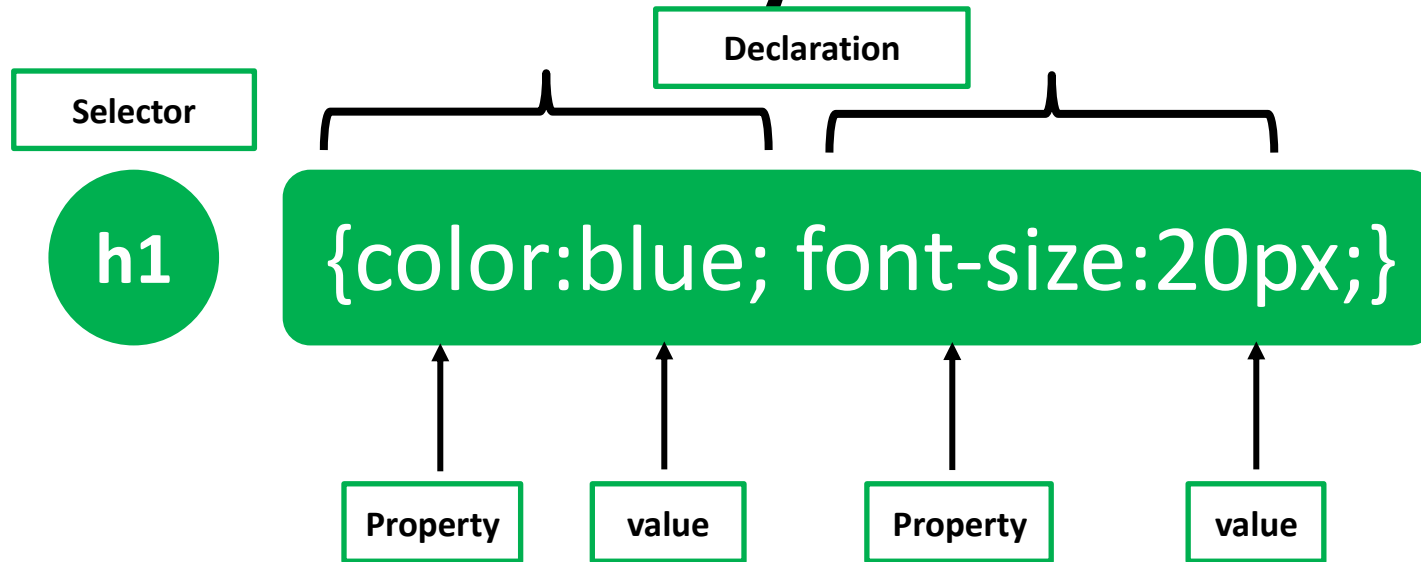
# What is CSS3

What is **CSS3**?

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files



# CSS3 Syntax





# CSS3 Types

Three Ways to Insert CSS

There are three ways of inserting a style sheet:

**External CSS**

**Internal CSS**

**Inline CSS**



# CSS3 selectors

## CSS Selectors

CSS selectors are used to "find" (or select) the HTML elements you want to style.

We can divide CSS selectors into five categories:

Simple selectors (select elements based on name, id, class)

Combinator selectors (select elements based on a specific relationship between them)

Pseudo-class selectors (select elements based on a certain state)

Pseudo-elements selectors (select and style a part of an element)

Attribute selectors (select elements based on an attribute or attribute value)



# CSS3 Colors

Colors are specified using predefined color names, or **RGB**, **HEX**, **HSL**, **RGBA** values.



# CSS3 Background

In these chapters, you will learn about the following CSS background properties:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position
- Background-size



# CSS3 Borders

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

A hidden border.

A mixed border.





# CSS3 Border-Property shortcut

Border: width style color;

Property

value

value

value



# CSS3 outline

- **dotted** - Defines a dotted outline
- **dashed** - Defines a dashed outline
- **solid** - Defines a solid outline
- **double** - Defines a double outline
- **groove** - Defines a 3D grooved outline
- **ridge** - Defines a 3D ridged outline
- **inset** - Defines a 3D inset outline
- **outset** - Defines a 3D outset outline
- **none** - Defines no outline
- **hidden** - Defines a hidden outline



# CSS3 Margins & Padding

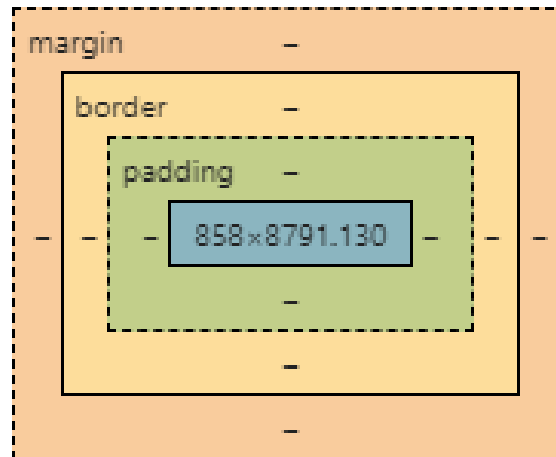
The CSS **margin** properties are used to create space around elements, outside of any defined borders.

With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

The CSS **padding** properties are used to generate space around an element's content, inside of any defined borders. With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).



# Example of Margin & Padding





# Margin & Padding

## Margin - Individual Sides

CSS has properties for specifying the margin for each side of an element:

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

## Padding - Individual Sides

CSS has properties for specifying the margin for each side of an element:

- `Padding-top`
- `Padding-right`
- `Padding-bottom`
- `Padding-left`



# Margin & Padding shortcut

## Margin

Margin: top right bottom left;

## Padding

Padding: top right bottom left;



# Height & Width

## height and width

The **height** and **width** properties are used to set the height and width of an element.

The height and width properties do not include padding, borders, or margins. It sets the height/width of the area inside the padding, border, and margin of the element.



# CSS Min, Max Property

Max-Height, Width


Min-Height, Width







# CSS3 Text Color

```
<style>
  h1 {
    color: pink;
  }
</style>
</head>
<body>
  <h1>Hello world</h1>
</body>
</html>
```



# **Text Formatting**



# Text Alignment

## Text Alignment

The `text-align` property is used to set the horizontal alignment of a text.

A text can be `left` or `right` aligned, `centered`, or `justified`.



# Text Alignment

```
<style>
  h1 {
    text-align: center; ←
  }
  h2 {
    text-align: left; ←
  }
  h3 {
    text-align: right; ←
  }
</style>
</head>
<body>
  <h1>this is h1</h1>
  <h2>this is h2</h2>
  <h2>this is h3</h2>
</body>
</html>
```



# Text Decoration

```
<style>
  h1 {
    text-decoration: underline; ←
  }
  h2 {
    text-decoration: wavy; ←
  }
  h3 {
    text-decoration: line-through; ←
  }
</style>
</head>
<body>
  <h1>this is h1</h1>
  <h2>this is h1</h2>
  <h3>this is h1</h3>
</body>
</html>
```



# Text Decoration

```
<style>
  h1 {
    text-decoration: underline;
  }
  h2 {
    text-decoration: wavy;
  }
  h3 {
    text-decoration: line-through;
  }
  a {
    text-decoration: none;
  }
</style>
</head>
<body>
  <h1>this is h1</h1>
  <h2>this is h1</h2>
  <h3>this is h1</h3>
  <a href="#">Youtube.com</a>
</body>
</html>
```



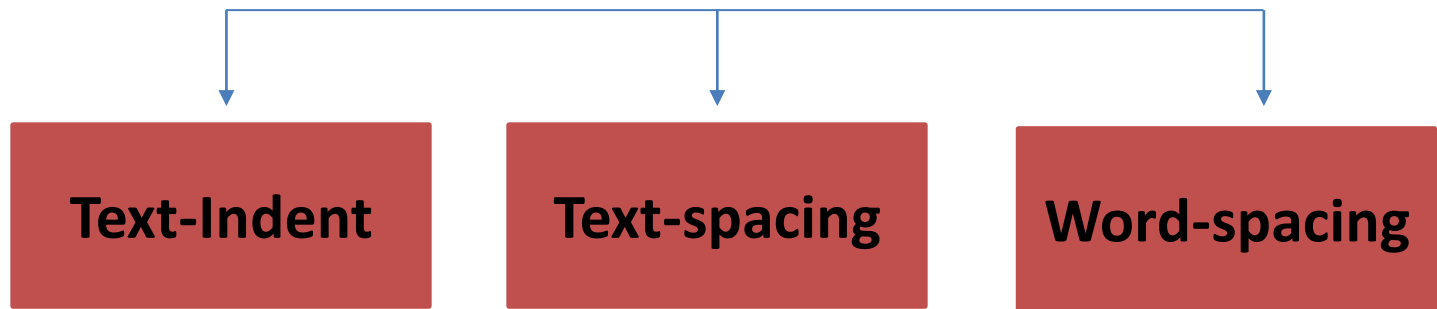
# Text Transformation

```
<style>
  h4 {
    text-transform: uppercase; ←
  }
  h4 {
    text-transform: lowercase; ←
  }

  h4 {
    text-transform: capitalize; ←
  }
</style>
</head>
<body>
  <h4>Hello world</h4>
  <h4>Hello world</h4>
  <h4>Hello world</h4>
</body>
</html>
```



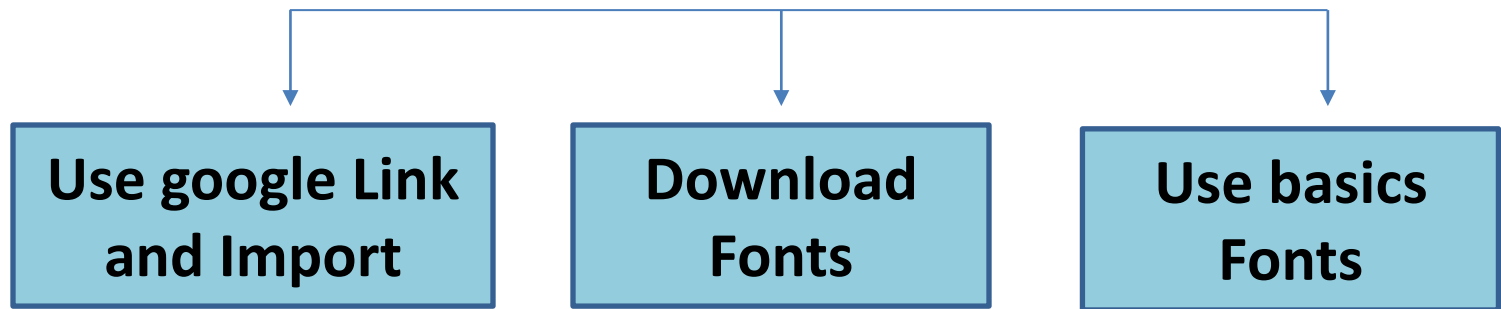
# Text Spacing







# CSS Fonts-Family





# CSS icons

**Website icons are visual cues that help communicate concepts on your website pages. they can act as a visual language to prevent the user from interpreting a concept incorrectly and to ensure effective communication. Part of great website design are the icons you use on your website.**



# CSS Display Property





# CSS Positioning

Static

An element with `position: static;` is not positioned in any special way; it is always positioned according to the normal flow of the page

Relative

An element with `position: Relative;` is positioned related to its real position.

Absolute

An element with `position: absolute;` is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed.)

Fixed

An element with `position: fixed;` is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.

Sticky

An element with `position: sticky;` is positioned based on the user's scroll position.



# CSS Layout - The z-index Property





# CSS Overflow

- **visible** - Default. The overflow is not clipped. The content renders outside the element's box
- **hidden** - The overflow is clipped, and the rest of the content will be invisible
- **scroll** - The overflow is clipped, and a scrollbar is added to see the rest of the content
- **auto** - Similar to **scroll**, but it adds scrollbars only when necessary



# CSS Layout - float and

The CSS **float** property specifies how an element should **float**.

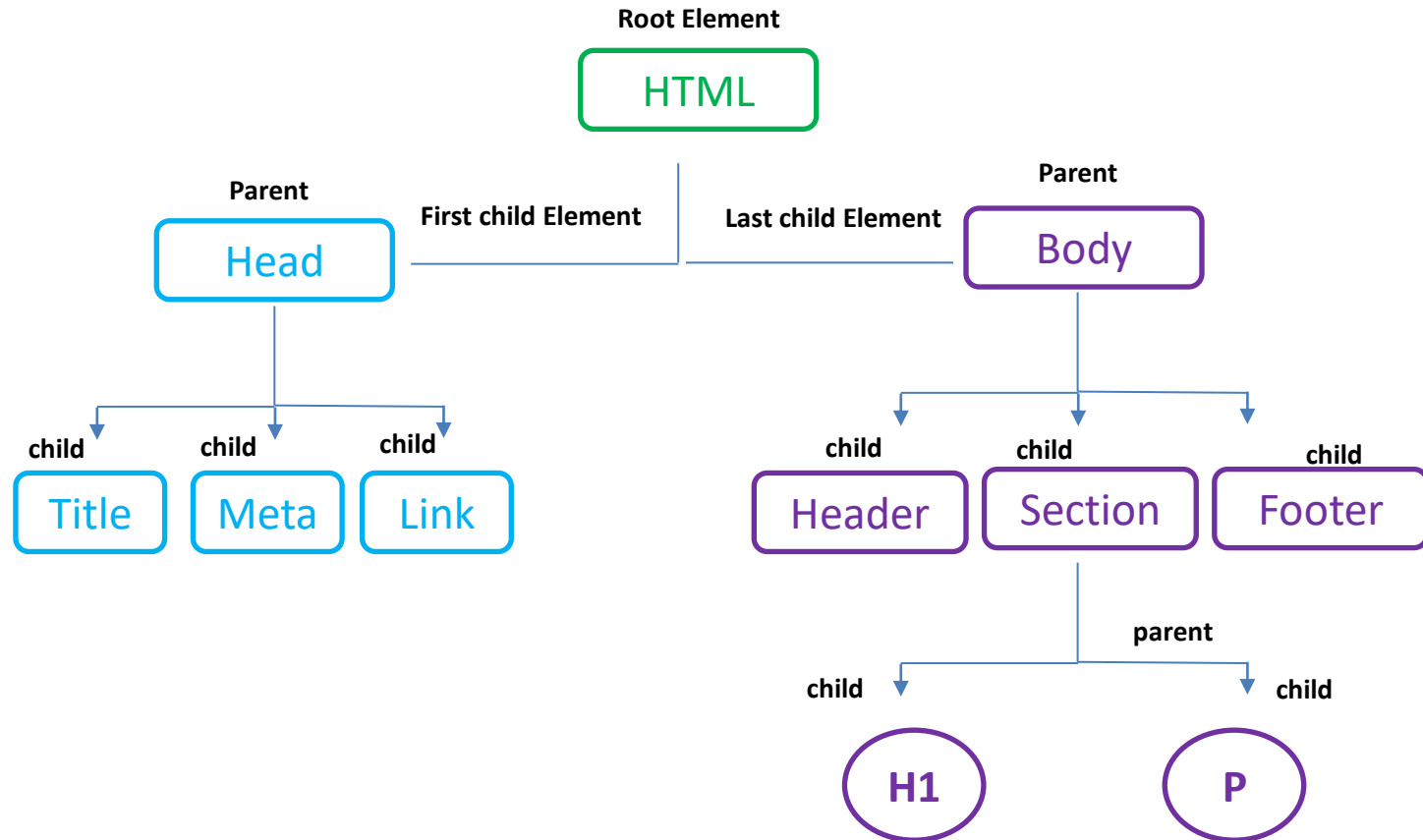
The CSS clear property specifies what elements can float beside the cleared element and on which side.

**Left**

**Right**



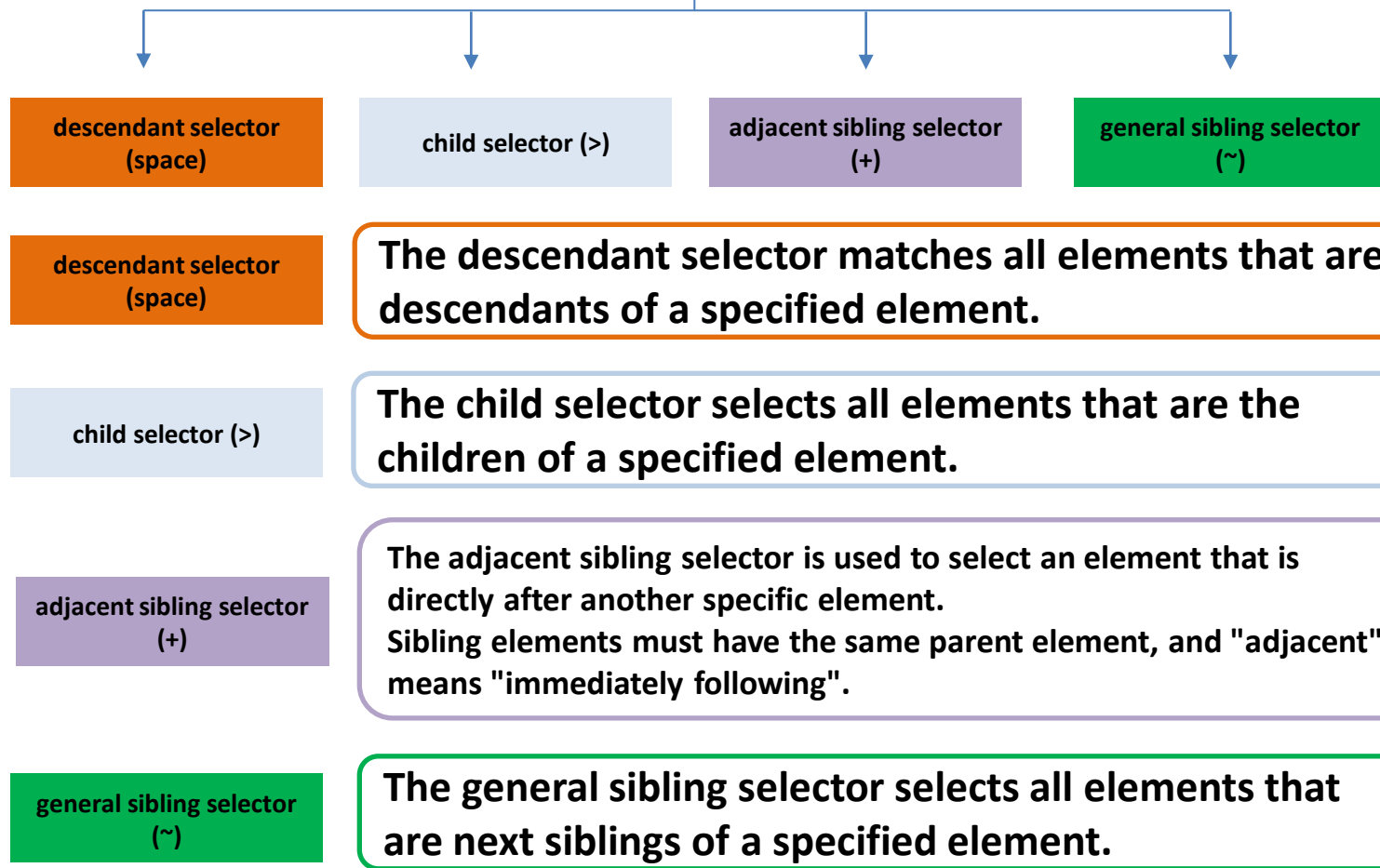
# DOM Model







# CSS Advance Selectors





# Pseudo-classes

A **pseudo-class** is used to define a special state of an element.

For example, it can be used to:

Style an element when a user mouses over it

Style visited and unvisited links differently

Style an element when it gets focus



# Pseudo-classes

:first-child

:link

:enabled

:last-child

:hover

:invalid

:nth-child

:active

:read-only

:nth-last-child()

:target

:valid

:nth-of-type()

:focus

:required

:empty

:checked

:not()

:disabled



# Pseudo-elements

**::placeholder**

The **::placeholder** selector selects form elements with placeholder text, and let you style the placeholder text. The placeholder text is set with the placeholder attribute, which specifies a hint that describes the expected value of an input field.

**::after**

The **::after** pseudo-element can be used to insert some content after the content of an element.

**::selection**

The **::selection** pseudo-element matches the portion of an element that is selected by a user.

**::marker**

The **::marker** pseudo-element selects the markers of list items.



# Opacity & important

The **opacity property** specifies the opacity/transparency of an element.

The **!important** rule in CSS is used to add more importance to a property/value than normal. In fact, if you use the !important rule, it will override ALL previous styling rules for that specific property on that element!



# Border-Radius

The CSS **border-radius** property defines the radius of an element's corners.

Tip: This property allows you to add rounded corners to elements!

1. Rounded corners for an element with a specified background color:
2. Rounded corners for an element with a border:
3. Rounded corners for an element with a background image:



# Box-Shadow

The **box-shadow** property attaches one or more shadows to an element.



# Grid System

The CSS **Grid** Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.





# Grid properties

Grid-template-rows

Justify-items

Grid-row-start

Grid-template-columns

Grid-row-end

Grid-template-areas

Justify-self

Grid-column-start

Grid-row-gap

align-items

Grid-column-end

Grid-column-gap

Grid-row

Grid-gap

align-self

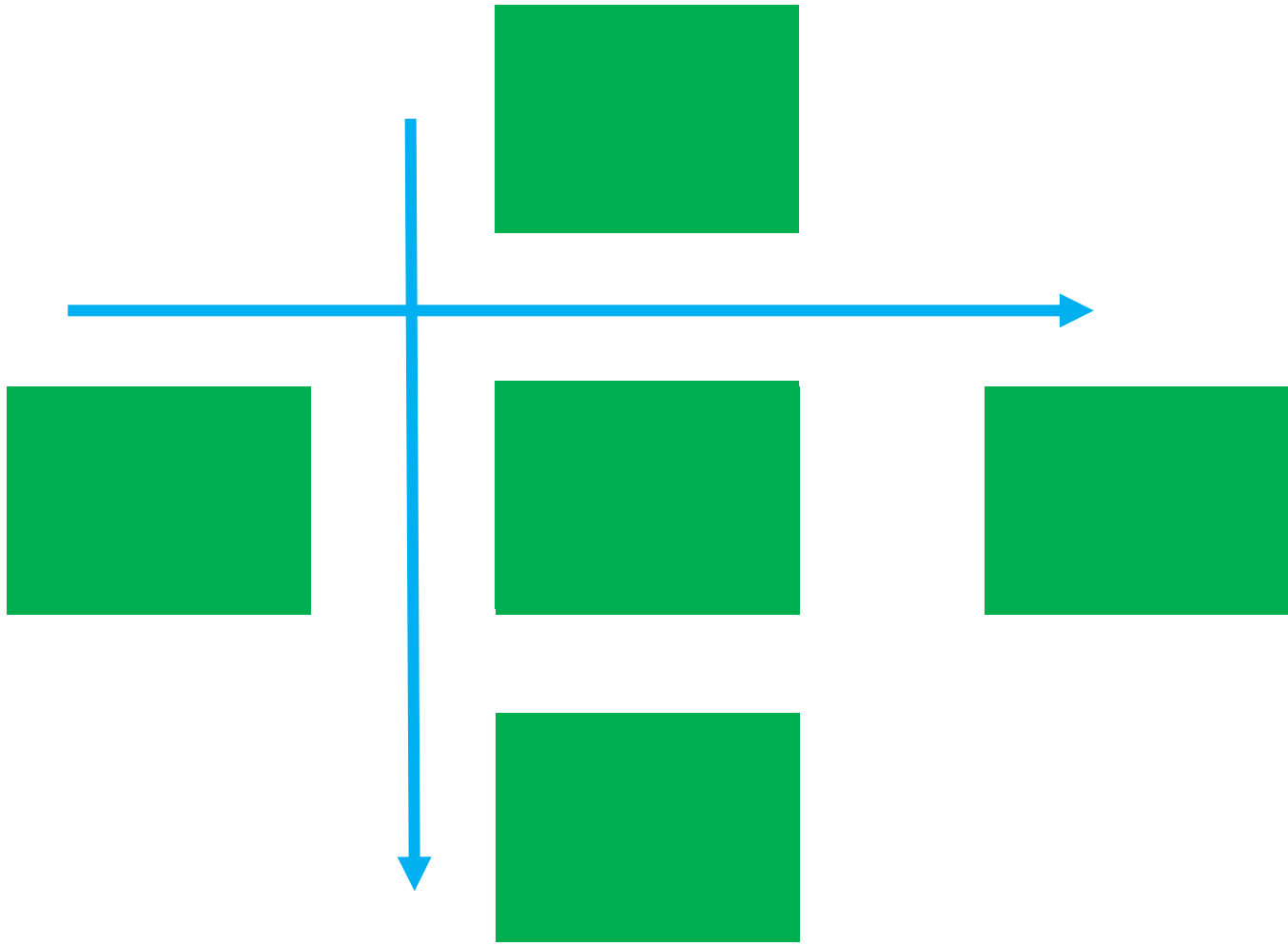
Grid-column





# Flexbox

The flex property **sets the flexible length on flexible items**. Note: If the element is not a flexible item, the flex property has no effect.





# Flex properties

Flex-direction

Flex-wrap

Flex-flow

Flex-grow

Justify-content

Align-content

order

Align-self

align-items





# Media Queries

All

Screen

Print

Speech





# Syntax of Media Queires

@media screen and (max-width:200px) {CSS properties}

