Introduction To Web Development

for Absolute Beginners

VER 0.3.5

slides link

MY NAME IS

Yan

Github.Com/Metaory

JS, Lua, Shell, Vim, Linux lover

Developer Advocate

SOLUTION ARCHITECT AT HELLOGOLD

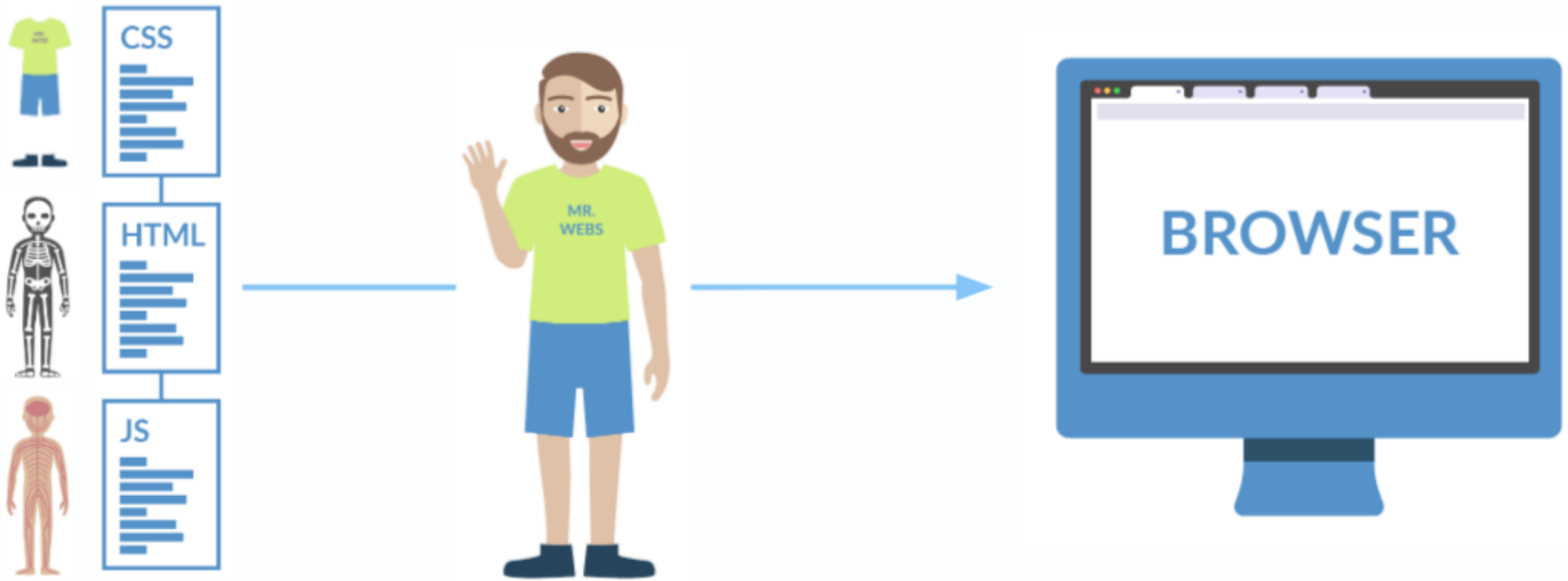
Tooling

- A **text editor**, to write code in. This could be a text editor (e.g. **Visual Studio Code**, **Notepad++**, **Sublime Text**, **Atom**, **GNU Emacs**, or **VIM**), or a hybrid editor (e.g. **WebStorm**). Office document editors are not suitable for this use, as they rely on hidden elements that interfere with the rendering engines used by web browsers.

- **Web browsers**, to test code in. Currently, the most-used browsers are **Firefox**, **Chrome**, **Opera**, **Safari**, **Internet Explorer** and **Microsoft Edge**. You should also test how your site performs on mobile devices and on any old browsers your target audience may still be using (such as IE 8–10).


JS
HTML
CSS

WEB ANATOMY



Structure (HTML)

HTML to define the **content** of web pages

Style (CSS)

CSS to specify the **layout** and **style** web pages

Functionality (JavaScript)

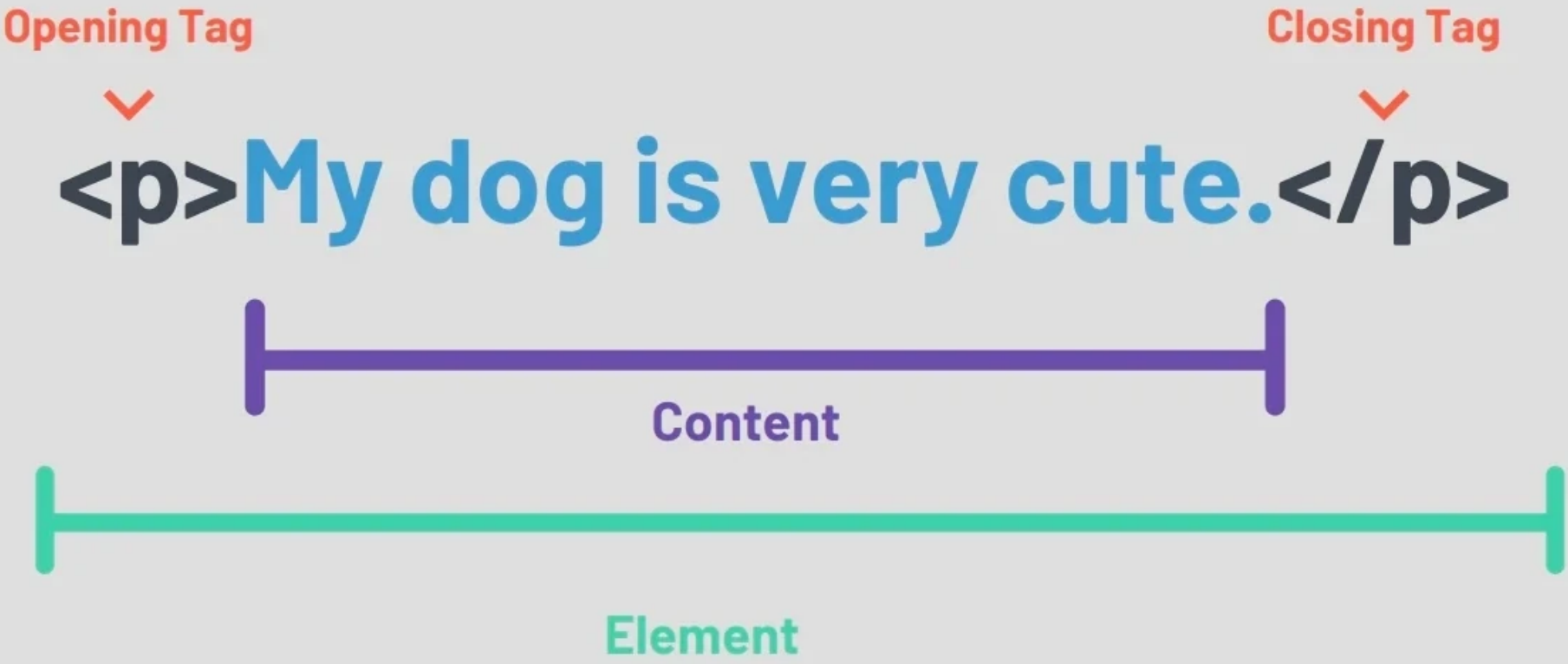
JavaScript to program the **behavior** of web pages

HTML

HTML is the standard markup language for creating Web pages.

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content

HTML ELEMENT STRUCTURE



HTML Element Parts

- **The opening tag**: This consists of the name of the element, wrapped in opening and closing angle brackets. This states where the element begins or starts to take effect.
- **The closing tag**: This is the same as the opening tag, except that it includes a forward slash before the element name. This states where the element ends.
- **The content**: This is the content of the element.
- **The element**: The opening tag, the closing tag, and the content together comprise the element.

A Simple HTML Document

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
  </body>
</html>
```

Example Explained

- The `<!DOCTYPE html>` declaration defines that this document is an HTML5 document
- The `<html>` element is the root element of an HTML page
- The `<head>` element contains meta information about the HTML page
- The `<title>` element specifies a title for the HTML page
- The `<body>` element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The `<h1>` element defines a large heading
- The `<p>` element defines a paragraph

HTML Headings

<h1> Heading 1 </h1>

<h2> Heading 2 </h2>

<h3> Heading 3 </h3>

<h4> Heading 4 </h4>

HTML Paragraphs

<P> And <Pre>

<p>

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

<pre>

```
Lorem ipsum dolor sit amet
consectetur adipiscing elit
```

HTML Horizontal Rules

The `<hr>` tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule.

The `<hr>` element is used to separate content (or define a change) in an HTML page:

HTML Line Breaks

The HTML `<br>` element defines a line break.

Use `<br>` if you want a line break (a new line) without starting a new paragraph:

HTML Block-Level Elements

A block-level element always starts on a new line, and the browsers automatically add some space (a margin) before and after the element.

Two commonly used block elements are: `<p>` and `<div>`.

HTML Links

Links allow users to click their way from page to page.

HTML Attributes

HTML attributes provide additional information about HTML elements.

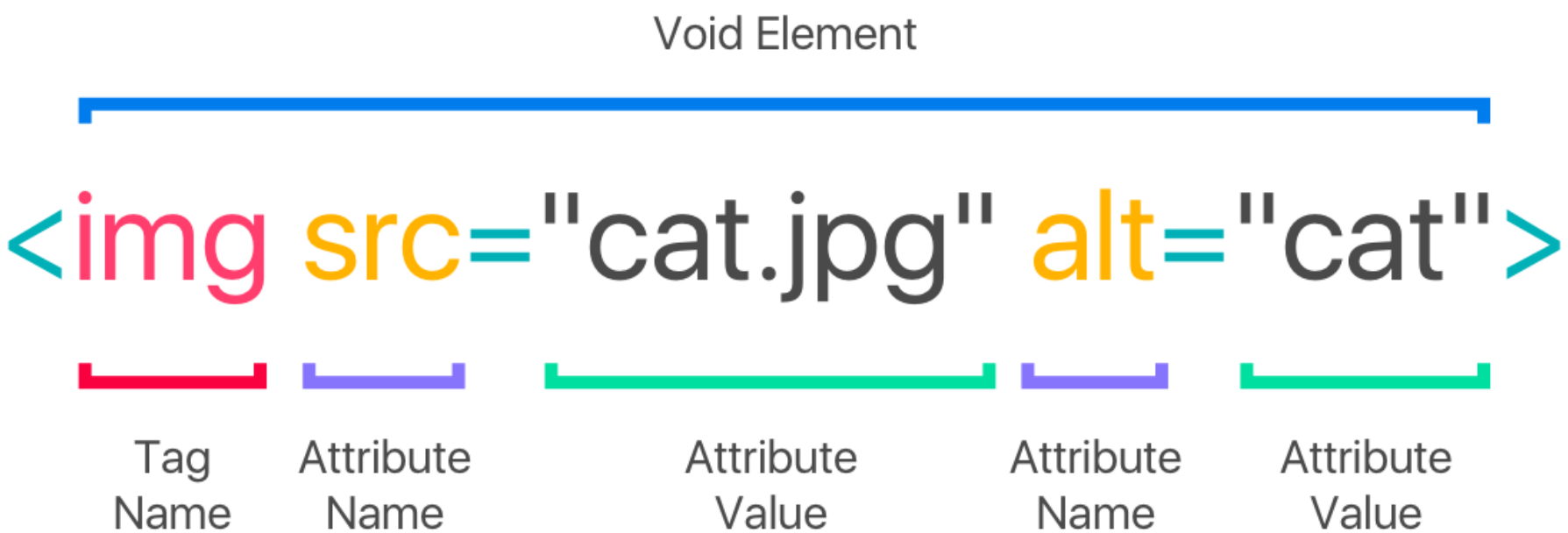
- All HTML elements can have attributes
- Attributes provide additional information about elements
- Attributes are **always** specified in the **start tag**
- Attributes usually come in **name/value** pairs like: `name="value"`

HTML Image Tag

The `<img>` tag is used to embed an image in an HTML page.

The src `attribute` specifies the path to the image to be displayed

```
<img src="img_girl.jpg">
```


Void Element
<img src="cat.jpg" alt="cat">
Tag Name
Attribute Name
Attribute Value
Attribute Name
Attribute Value

HTML Class Attribute

The HTML `class` attribute is used to specify a class for an HTML element.

Multiple HTML elements can share the same class.

HTML Id Attribute

The HTML `id` attribute is used to specify a unique id for an HTML element.

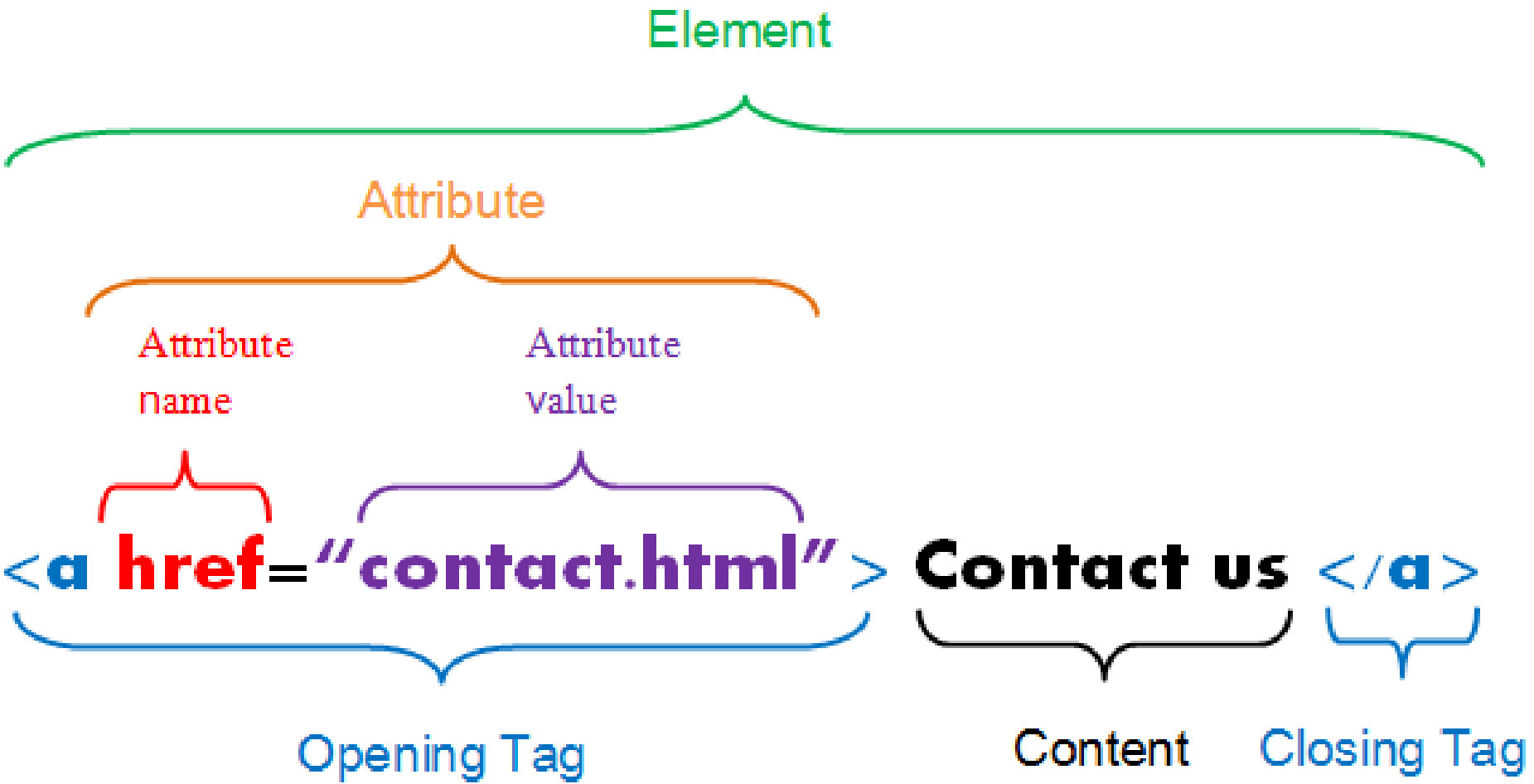
You **cannot** have more than one element with the same **id** in an HTML document.

HTML Link Tag

The `<a>` tag defines a hyperlink.

The href attribute specifies the URL of the page the link goes to

```
<a href="https://www.w3schools.com">Visit W3Schools</a>
```


Element
Attribute
Attribute name
Attribute value

Contact us

Opening Tag
Content
Closing Tag

Nested HTML Elements

HTML elements can be nested.

All HTML documents consist of nested HTML elements.

The following example contains four HTML elements (`<html>`, `<body>`, `<h1>` and `<p>`):

```
<!DOCTYPE html>
<html>
  <body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
  </body>
</html>
```

Case Sensitivity

HTML is Not Case Sensitive

HTML tags are not case sensitive: `<P>` means the same as `<p>`.

HTML Text Formatting

This text is bold

This text is italic

This is _{subscript} and ^{superscript}

- `<b>` - Bold text
- `<strong>` - Important text
- `<i>` - Italic text
- `<em>` - Emphasized text
- `<mark>` - Marked text
- `<small>` - Smaller text
- `<del>` - Deleted text
- `<ins>` - Inserted text
- `<sub>` - Subscript text
- `<sup>` - Superscript text
- `...`

HTML Semantic Elements

Semantic elements = elements with a meaning.

A **semantic** element clearly describes its meaning to both the browser and the developer.

Examples of non-semantic elements: `<div>` and `<span>` - **Tells nothing about its content.**

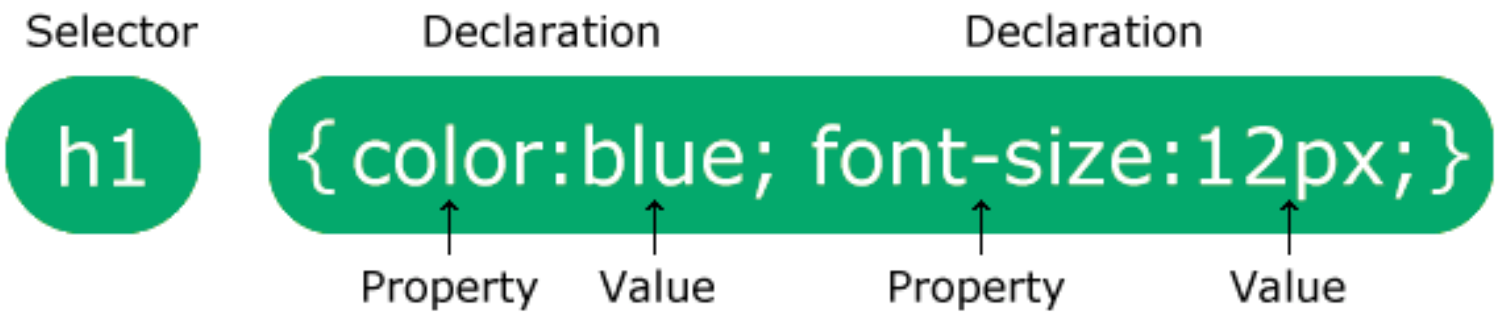
Examples of semantic elements: `<form>`, `<table>`, and `<article>` - **Clearly defines its content.**

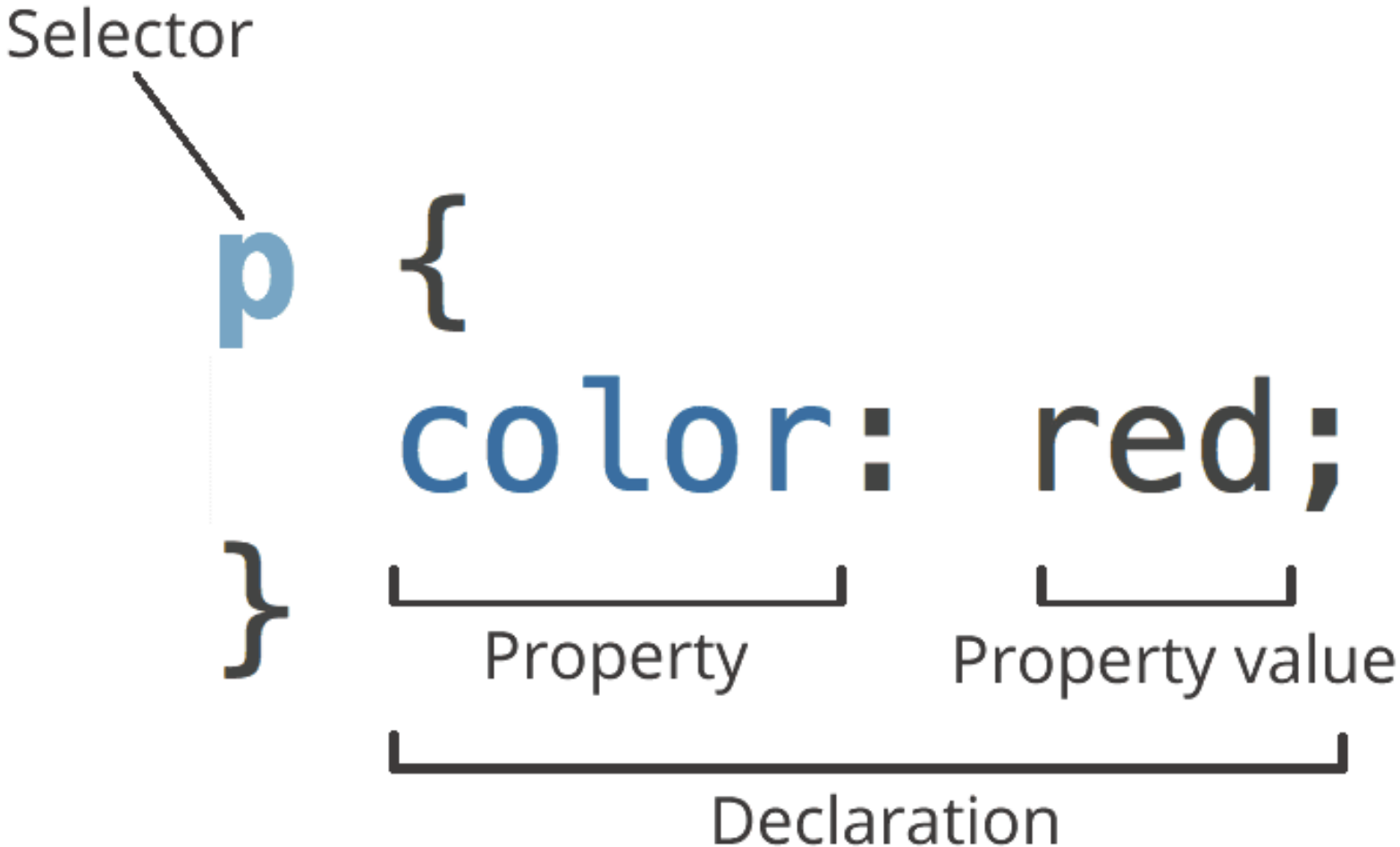
CSS

CSS stands for Cascading Style Sheets.

CSS can control the layout of multiple web pages all at once.

With CSS, you can control the color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes, and much more!


Selector
Declaration
Declaration
h1
{color:blue; font-size:12px;}
Property
Value
Property
Value


Selector
p {
color: red;
}
Property
Property value
Declaration

CSS Syntax Example

```css
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

CSS Syntax Explained

The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

Using CSS

CSS Inline

by using the `style` attribute inside HTML elements

CSS Internal

by using a `<style>` element in the `<head>` section

CSS External

External - by using a `<link>` element to link to an external CSS file

Inline CSS

An inline CSS is used to apply a unique style to a single HTML element.

An inline CSS uses the `style` attribute of an HTML element.

The following example sets the text color of the `<h1>` element to blue

```
<h1 style="color:blue;">A Blue Heading</h1>
```

Internal CSS

An internal CSS is used to define a style for a single HTML page.

An internal CSS is defined in the `<head>` section of an HTML page, within a `<style>` element.

The following example sets the text color of ALL the `<h1>` elements (on that page) to blue, and the text color of ALL the `<p>` elements to red.

```
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
```

External CSS

An external style sheet is used to define the style for many HTML pages.

To use an external style sheet, add a link to it in the `<head>` section of each HTML page:

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```

CSS Colors

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

These values represent the same color in different formats:

```
h1 {color: #ff0000;}
h2 {color: rgb(255, 0, 0);}
h3 {color: hsl(0, 100%, 50%);}
```

Background Color

```
<p style="background-color:Tomato;">Lorem ipsum ... </p>
```

Text Color

```
<h1 style="color:Tomato;">Hello World</h1>
```

Border Color

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
```

CSS Units

CSS has several different units for expressing a length.

Many CSS properties take **"length"** values, such as `width`, `margin`, `padding`, `font-size`, etc.

Length is a number followed by a length unit, such as `10px`, `2em`, etc.

```
h1 {
  font-size: 60px;
}

p {
  font-size: 25px;
  line-height: 50px;
}
```

Absolute

- `cm` centimeters
- `mm` millimeters
- `in` inches (1in = 96px = 2.54cm)
- `px` pixels (1px = 1/96th of 1in)
- `pt` points (1pt = 1/72 of 1in)
- `...`

Relative

- `em` Relative to the font-size of the element
- `rem` Relative to font-size of the root element
- `vw` Relative to 1% of the width of the viewport
- `vh` Relative to 1% of the height of the viewport
- `%` Relative to the parent element
- `...`

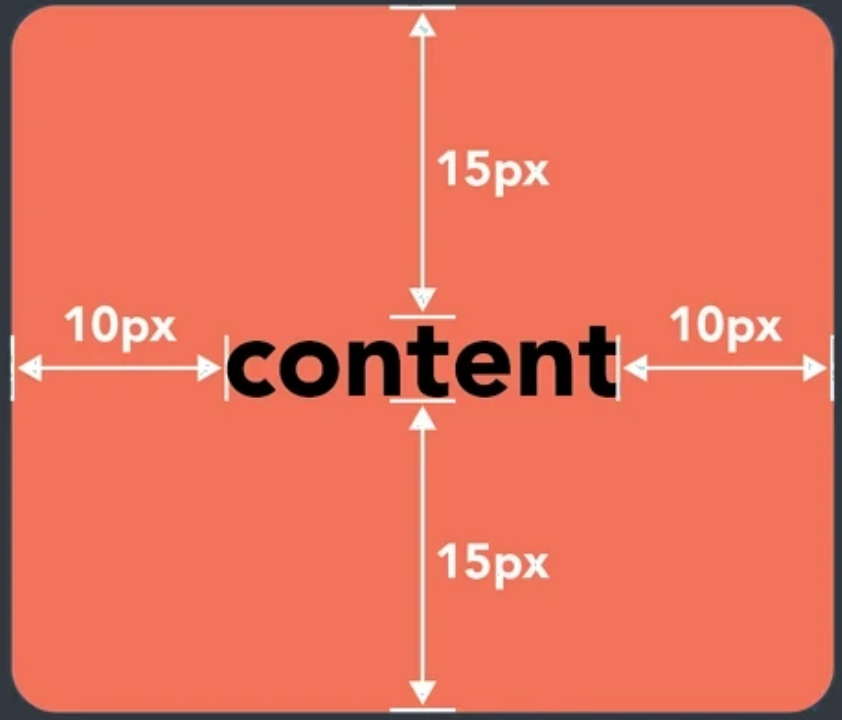
CSS Padding

An element's padding is the space between its content and its border.

Padding is used to create space around an element's content, inside of any defined borders.

The padding property is a **shorthand** property for:

- `padding-top`
- `padding-right`
- `padding-bottom`
- `padding-left`

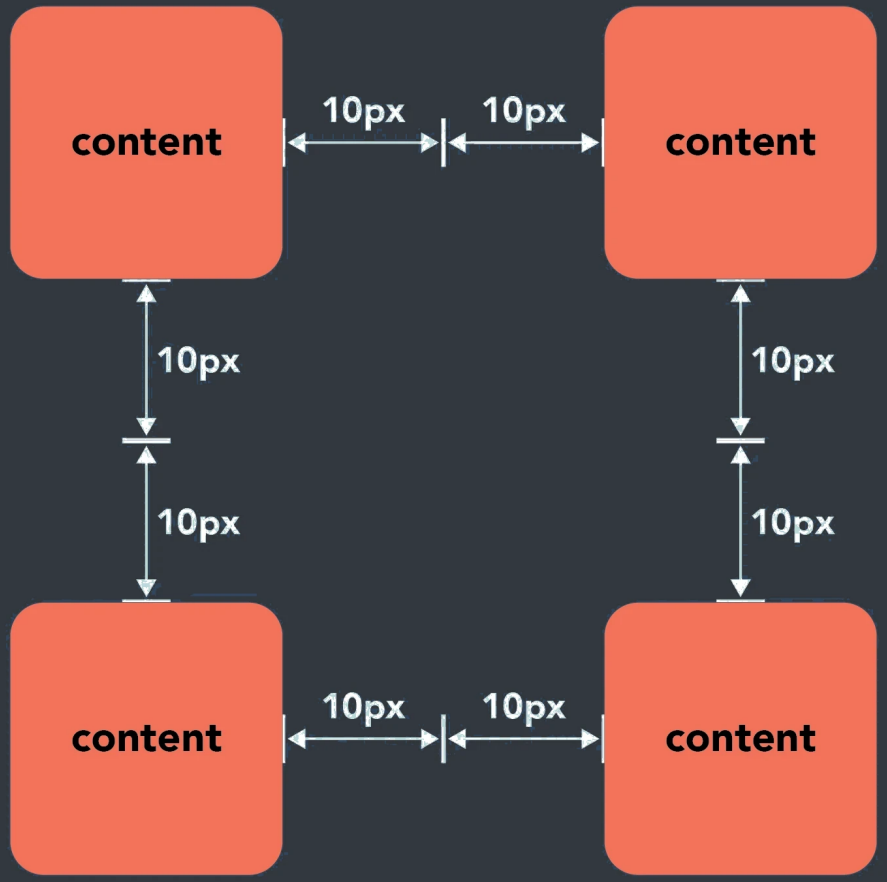

15px
10px
content
10px
15px

CSS Margin

Margins are used to create space around elements, outside of any defined borders.

The margin property is a **shorthand** property for:

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


content
10px
10px
content
10px
10px
10px
10px
content
10px
10px
content

CSS 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`.

- **The `vertical-align` property sets the vertical alignment of an element.**

- `color` - Sets the color of text
- `background-color` - Specifies the background color of an element
- `background-image` - Specifies one or more background images for an element
- `border` - A shorthand property for border-width, border-style and border-color
- `box-shadow` - Attaches one or more shadows to an element
- `display` - Specifies how a certain HTML element should be displayed
- `font-weight` - Specifies the weight of a font
- `margin` - Sets all the margin properties in one declaration
- `padding` - A shorthand property for all the padding-* properties
- `opacity` - Sets the opacity level for an element
- `text-align` - Specifies the horizontal alignment of text
- `...`

CSS Selectors

A CSS selector selects the HTML element(s) you want to style.

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

CSS selectors 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)

CSS Simple Selectors

Selector name	What does it select	Example
Element selector	All HTML elements of the specified type.	`p` selects `<p>`
ID selector	The element on the page with the specified ID.	`#my-id` selects `<p id="my-id">`
Class selector	The element(s) on the page with the specified class.	`.my-class` selects `<p class="my-class">`

The CSS Element Selector

The element selector selects HTML elements based on the element name.

Here, all `<p>` elements on the page will be **center-aligned**, with a **red text color**:

```
p {
  text-align: center;
  color: red;
}
```

The CSS Id Selector

The id selector uses the id attribute of an HTML element to select a specific element.

The **id** of an element is **unique** within a page, so the **id selector** is used to select **one** unique element!

The CSS rule below will be applied to the HTML element with `id="para1"`:

```
#para1 {
  text-align: center;
  color: red;
}
```

Using The Class Attribute

The `class` attribute is often used to point to a class name in a style sheet.

```
<!DOCTYPE html>
<html>
  <head>
  <style>
  .city {
    background-color: tomato;
    color: white;
    border: 2px solid black;
    margin: 20px;
    padding: 20px;
```

Selecting Multiple Elements

You can also select multiple elements and apply a single ruleset to all of them. Separate multiple selectors by commas.

```
p, li, h1 {
  color: red;
}
```

Advance CSS Selectors

Selector	Example	Example Description
`.class`	`.intro`	Selects all elements with `class="intro"`
`.class1.class2`	`.name1.name2`	Selects all elements with both `name1` and `name2` set within its class attribute
`.class1 .class2`	`.name1 .name2`	Selects all elements with `name2` that is a descendant of an element with `name1`
`#id`	`#firstname`	Selects the element with `id="firstname"`
`*`	`*`	Selects all elements

Advance CSS Selectors

Selector	Example	Example Description
`element`	`p`	Selects all `<p>` elements
`element.class`	`p.intro`	Selects all `<p>` elements with `class="intro"`
`element,element`	`div, p`	Selects all `<div>` elements and all `<p>` elements
`element element`	`div p`	Selects all `<p>` elements inside `<div>` elements
`element>element`	`div > p`	Selects all `<p>` elements where the parent is a `<div>` element
`element+element`	`div + p`	Selects the first `<p>` element that is immediately after `<div>` elements

JavaScript

JavaScript

JavaScript is the world's most popular programming language.

JavaScript is the programming language of the Web.

JavaScript is easy to learn.

VARIABLES

Variables *are containers for storing data (storing data values).*

Variable Types

- global

```
myVariable = 'Steve';
```

- let/var

```
let myVariable = 'Steve';
myVariable = 'new name';
var name = 'John';
```

- const

```
const myVariable = 'Steve';
```

Data Types

String

```
let myVariable = 'Bob';
```

Number

```
let myVariable = 10;
```

Boolean

```
let myVariable = true;
```

Object

```
let myVariable = { name: 'John' }
```

Array

```
let myVariable =
[1,'Bob','Steve',10];
```

Variable Examples

```
var firstName = 'Pou' // string
var lastName = 'Yan' // string
var fullName = firstName + lastName // string
var age = 34 // number
var single = true // boolean

console.log(firstName)
console.log('fullname:', fullName)
```

One of many JavaScript HTML methods is `getElementById()`.

```
document.getElementById("demo").innerHTML = fullName
```

References

- W3 HTML Tutorial
- W3 CSS Tutorial
- W3 CSS Selectors Reference
- W3 JavaScript Tutorial
- W3 Howto
- MDN Getting started with the Web Tutorial
- MDN Front-end Web Developer Tutorial
- MDN HTML Elements reference
- MDN HTML reference
- MDN CSS reference
- MDN JavaScript reference