IS231: Web Technology
Cascading Style Sheets

By: Neamat El Tazi

References

- **Some Figures and Slides References**
  - **W3shools.com**

HTML Style Attribute

```
<h1 style="font-family:verdana;">This is a heading</h1>
<p style="font-family:courier;">This is a paragraph.</p>
```

This is a heading

This is a paragraph.

```
<!DOCTYPE html>
<html>
<body>

<p>I am normal</p>
<p style="color:red;">I am red</p>
<p style="color:blue;">I am blue</p>
<p style="font-size:50px;">I am big</p>

</body>
</html>
```

I am Red

I am Blue

I am Big

Cascading Style Sheets (CSS)

- CSS is the language we use to style an HTML document.
- CSS describes how HTML elements should be displayed on Screen, paper, or in other media

Why do we need CSS?

- Messy Style attributes in HTML
- Separate style from content to add it only once and remove duplicates inside the HTML document
- 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 (separately)

CSS Solved a Big Problem

- HTML was NEVER intended to contain tags for formatting a web page!
- HTML was created to **describe the content** of a web page, like:
  - <h1>This is a heading</h1>
  - This is a paragraph.
- When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.
- To solve this problem, the World Wide Web Consortium (W3C) created CSS.
- CSS removed the style formatting from the HTML page!

CSS Saves a Lot of Work

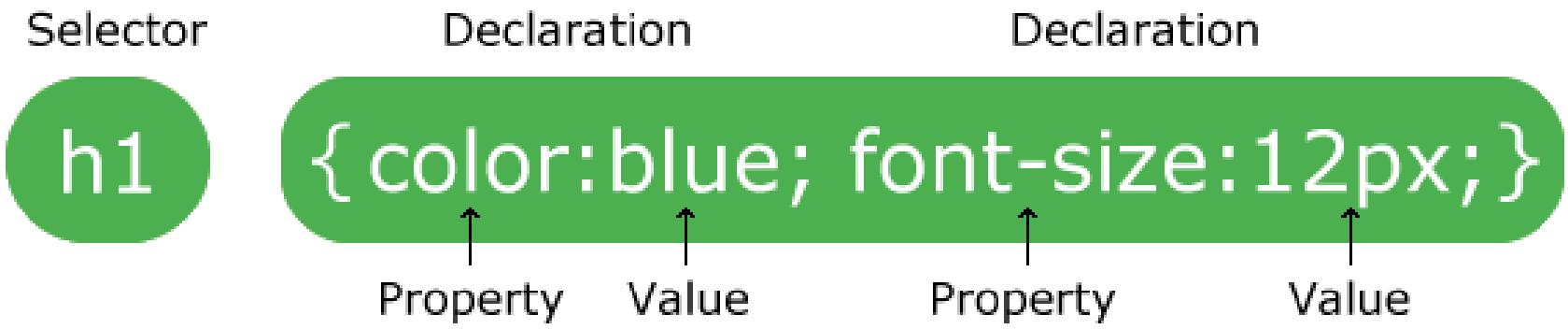
- The style definitions are normally saved in external .css files.
- With an external stylesheet file, you can change the look of an entire website by changing just one file!

CSS Demo

- Run the HTML page "css_demo.html" to see how can you change the style of the whole page using different "Styles" elements.

CSS Syntax

- A CSS rule-set consists of a selector and a declaration block:



- 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.

How To Add CSS

- When a browser reads a style sheet, it will format the HTML document according to the information in the style sheet.
- Three Ways to Insert CSS
  - External CSS
  - Internal CSS
  - Inline CSS

External CSS

- With an external style sheet, you can change the look of an entire website by changing just one file!
- Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

External CSS

- An external style sheet can be written in any text editor, and must be saved with a .css extension.
- The external .css file should not contain any HTML tags.

```
body {
  background-color: lightblue;
}

h1 {
  color: navy;
  margin-left: 20px;
}
```

Internal CSS

- An internal style sheet may be used if one single HTML page has a unique style.
- The internal style is defined inside the <style> element, inside the head section.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Inline CSS

- An inline style may be used to apply a unique style for a single element.
- To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```

Multiple Style Sheets

- If some properties have been defined for the same selector (element) in different style sheets, the value from the last read style sheet will be used.

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="mystyle.css">
  <link rel="stylesheet" href="mystyle2.css">
 <style>
    h1{color:green;}
 </style>
</head>
    <body>

    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>

    </body>
    </html>
```

```
/* File mystyle.css*/
h1   {color: blue;}
p    {color: red;}
```

```
/* File mystyle2.css*/
h1   {color: red;}
```

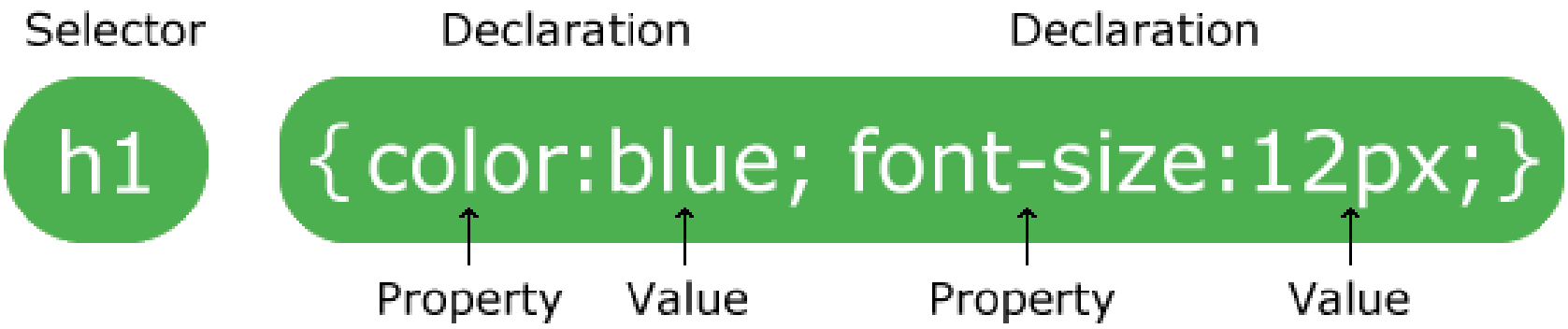
Cascading Order

- What style will be used when there is more than one style specified for an HTML element?

- All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:
  1. Inline style (inside an HTML element)
  2. External and internal style sheets (in the head section)
  3. Browser default

- So, an inline style has the highest priority, and will override external and internal styles and browser defaults.

CSS Syntax


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

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:

1. **Simple selectors** (select elements based on name, id, class)
2. **Combinator selectors** (select elements based on a specific relationship between them)
3. **Pseudo-class selectors** (select elements based on a certain state)
4. **Pseudo-elements selectors** (select and style a part of an element)
5. **Attribute selectors** (select elements based on an attribute or attribute value)

The CSS element Selector

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

```
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!
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.

The CSS id Selector

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

Check demo (css_1.html)

The CSS class Selector

- The class selector selects HTML elements with a specific class attribute.

- To select elements with a specific class, write a period (.) character, followed by the class name.

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

Check demo (css_2.html)

CSS Class Selector

- You can also specify that only specific HTML elements should be affected by a class.

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

```
<p class="center">This paragraph is styled
with class center.</p>
```

Check demo (css_3.html)

CSS Class Selector

- HTML elements can also refer to more than one class.

```
<p class="center large">This paragraph
refers to two classes.</p>
```

Check demo (css_4.html)

The CSS Universal Selector

- The universal selector (*) selects all HTML elements on the page.

```
* {
  text-align: center;
  color: blue;
}
```

Check demo (css_5.html)

The CSS Grouping Selector

- The grouping selector selects all the HTML elements with the same style definitions.
- Look at the following CSS code (the h1, h2, and p elements have the same style definitions):

```
h1 {
  text-align: center;
  color: red;
}

h2 {
  text-align: center;
  color: red;
}

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

The CSS Grouping Selector

- It will be better to group the selectors, to minimize the code.
- To group selectors, separate each selector with a comma.

```
h1, h2, p {
  text-align: center;
  color: red;
}
```

Check demo (css_6.html)

Summary of All CSS Simple Selectors

Selector	Example	Example description
.class	.intro	Selects all elements with class="intro"
#id	#firstname	Selects the element with id="firstname"
*	*	Selects all elements
element	p	Selects all <p> elements
element,element,..	div, p	Selects all elements and all <p> elements

CSS Combinator Selectors

- A combinator is something that explains the relationship between the selectors.
- A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator.
- There are four different combinators in CSS:

1. descendant selector (space)
2. child selector (>)
3. adjacent sibling selector (+)
4. general sibling selector (~)

Descendant Selector

- The descendant selector matches all elements that are descendants of a specified element.

- The following example selects all <p> elements inside elements:

```
div p {
  background-color: yellow;
}
```

Check demo (css_7.html)

Child Selector

- The child selector selects all elements that are the children of a specified element.

- The following example selects all <p> elements that are children of a element:

```
div > p {
  background-color: yellow;
}
```

Check demo (css_7.html), add “>” to the style and see what Will happen

Adjacent Sibling Selector

- The adjacent sibling selector selects all elements that are the adjacent siblings of a specified element.
- Sibling elements must have the same parent element, and "adjacent" means "immediately following".
- The following example selects all <p> elements that are placed immediately after elements:

```
div + p {
  background-color: yellow;
}
```

Check demo (css_7.html), add “+” to the style and see what Will happen

General Sibling Selector

- The general sibling selector selects all elements that are siblings of a specified element.
- The following example selects all <p> elements that are siblings of elements:
- ```
  div ~ p {
    background-color: yellow;
  }
  ```

Check demo (css_7.html), add “~” to the style and see what Will happen
  ```

All CSS Combinator Selectors

Selector	Example	Example description
element element	div p	Selects all <p> elements inside elements
element>element	div > p	Selects all <p> elements where the parent is a element
element+element	div + p	Selects all <p> elements that are placed immediately after elements
element1~element2	p ~ ul	Selects every element that are preceded by a <p> element

CSS 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

The syntax of pseudo-classes:

```
selector:pseudo-class {
  property: value;
}
```

Anchor Pseudo-classes

- Links can be displayed in different ways:

```
/* unvisited link */
a:link {
  color: #FF0000;
}

/* visited link */
a:visited {
  color: #00FF00;
}

/* mouse over link */
a:hover {
  color: #FF00FF;
}

/* selected link */
a:active {
  color: #0000FF;
}
```

Check demo (css_8.html)

Pseudo-classes and CSS Classes

- Pseudo-classes can be combined with CSS classes:
- When you hover over the link in the example, it will change color:

```
a.highlight:hover {
  color: #ff0000;
}
```

Hover on

- An example of using the :hover pseudo-class on element:

```
div:hover {
  background-color: blue;
}
```

Check demo (css_9.html)

Simple Tooltip Hover

- Hover over a element to show a <p> element (like a tooltip):

```
p {
  display: none;
  background-color: yellow;
  padding: 20px;
}

div:hover p {
  display: block;
}
```

Check demo (css_10.html)

All CSS Pseudo Classes

Selector	Example	Example description
:active	a:active	Selects the active link
:checked	input:checked	Selects every checked <input> element
:disabled	input:disabled	Selects every disabled <input> element
:empty	p:empty	Selects every <p> element that has no children
:enabled	input:enabled	Selects every enabled <input> element
:first-child	p:first-child	Selects every <p> elements that is the first child of its parent
:first-of-type	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
:focus	input:focus	Selects the <input> element that has focus
:hover	a:hover	Selects links on mouse over
:in-range	input:in-range	Selects <input> elements with a value within a specified range

- Check the rest in https://www.w3schools.com/css/css_pseudo_classes.asp

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:

1. **Simple selectors** (select elements based on name, id, class)
2. **Combinator selectors** (select elements based on a specific relationship between them)
3. **Pseudo-class selectors** (select elements based on a certain state)
4. **Pseudo-elements selectors** (select and style a part of an element)
5. **Attribute selectors** (select elements based on an attribute or attribute value)

CSS Pseudo-elements

- A CSS pseudo-element is used to style specified parts of an element.
- For example, it can be used to:
  - Style the first letter, or line, of an element
  - Insert content before, or after, the content of an element

```
selector::pseudo-element {
  property: value;
}
```

The ::first-line Pseudo-element

- The ::first-line pseudo-element is used to add a special style to the first line of a text.

- The following example formats the first line of the text in all <p> elements:

```
p::first-line {
  color: #ff0000;
  font-variant: small-caps;
}
```

Check Demo (css_11.html)

All CSS Pseudo Elements

Selector	Example	Example description
::after	p::after	Insert something after the content of each <p> element
::before	p::before	Insert something before the content of each <p> element
::first-letter	p::first-letter	Selects the first letter of each <p> element
::first-line	p::first-line	Selects the first line of each <p> element
::selection	p::selection	Selects the portion of an element that is selected by a user

CSS Attribute Selectors

- Style HTML Elements With Specific Attributes
- It is possible to style HTML elements that have specific attributes or attribute values.

- The [attribute] selector is used to select elements with a specified attribute.
- The following example selects all <a> elements with a target attribute:

```
a[target] {
  background-color: yellow;
}
```

Check Demo (css_12.html)

CSS [attribute="value"] Selector

- The [attribute="value"] selector is used to select elements with a specified attribute and value.

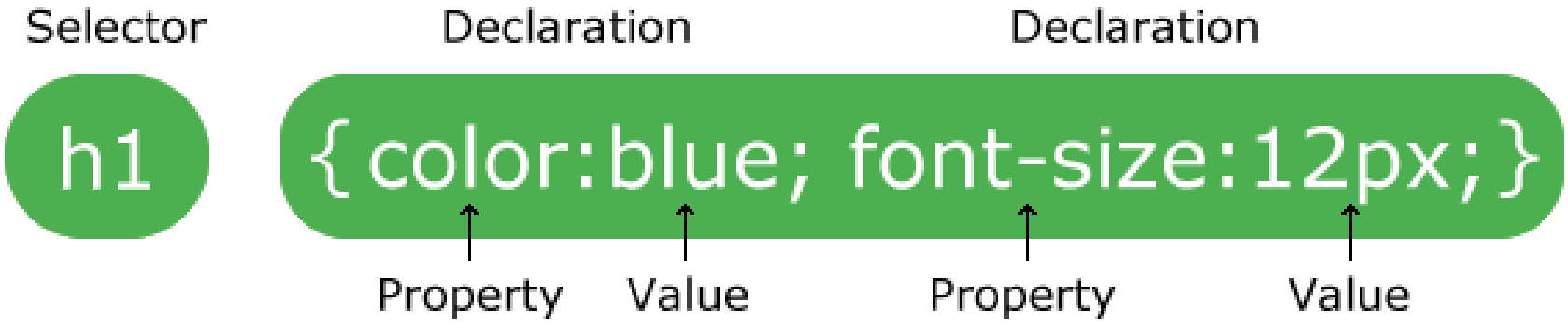
- The following example selects all <a> elements with a target="_blank" attribute:

```
a[target="_blank"] {
  background-color: yellow;
}
```

All CSS Attribute Selectors

Selector	Example	Example description
[*attribute*]	[target]	Selects all elements with a target attribute
[*attribute=value*]	[target=_blank]	Selects all elements with target="_blank"
[*attribute~=value*]	[title~=flower]	Selects all elements with a title attribute containing the word "flower"
[*attribute|=value*]	[lang|=en]	Selects all elements with a lang attribute value starting with "en"
[*attribute^=value*]	a[href^="https"]	Selects every <a> element whose href attribute value begins with "https"
[*attribute$=value*]	a[href$=".pdf"]	Selects every <a> element whose href attribute value ends with ".pdf"
[*attribute*=value*]	a[href*="w3schools"]	Selects every <a> element whose href attribute value contains the substring "w3schools"

CSS Syntax


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

CSS Comments

```css
/* This is a single-line comment */
p {
  color: red;
}

/* This is
a multi-line
comment */

p {
  color: red;
}
```

CSS Colors

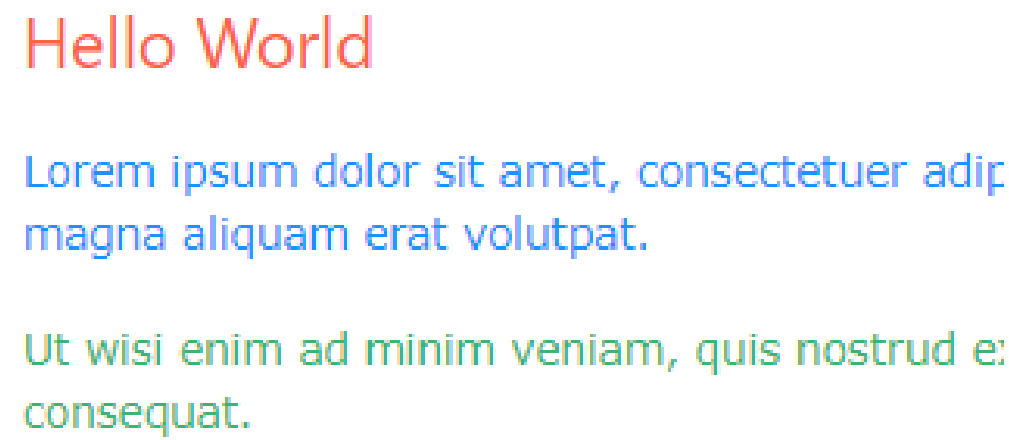

Tomato
Orange
DodgerBlue
MediumSeaGreen
Gray
SlateBlue
Violet
LightGray

CSS Background Color

- You can set the background color for HTML elements:


```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
<p style="background-color:Tomato;">Lorem ipsum...</p>
```

CSS Text Color




```
<h1 style="color:Tomato;">Hello World</h1>
<p style="color:DodgerBlue;">Lorem ipsum...</p>
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

CSS Border Color

- You can set the color of borders:


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

CSS RGB Colors

- In CSS, a color can be specified as an RGB value, using this formula:
- **rgb(*red, green, blue*)**
- Each parameter (red, green, and blue) defines the intensity of the color between 0 and 255.
- For example, rgb(255, 0, 0) is displayed as red, because red is set to its highest value (255) and the others are set to 0.
- rgb(0, 0, 0) is black
- rgb(255,255,255) is White

CSS HEX Colors

- In CSS, a color can be specified using a hexadecimal value in the form:
- ***#rrggbb***
- Where rr (red), gg (green) and bb (blue) are hexadecimal values between 00 and ff (same as decimal 0-255)
- For example, #ff0000 is displayed as red, because red is set to its highest value (ff) and the others are set to the lowest value (00).

CSS HSL Colors

- In CSS, a color can be specified using hue, saturation, and lightness (HSL) in the form:
- **hsl(*hue, saturation, lightness*)**

- Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, and 240 is blue.

- Saturation is a percentage value, 0% means a shade of gray, and 100% is the full color.

- Lightness is also a percentage, 0% is black, 50% is neither light or dark, 100% is white

CSS Backgrounds

The CSS background properties are used to define the background effects for elements.

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

CSS Borders

- The CSS border properties allow you to specify the style, width, and color of an element's border.

```
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
```

CSS Margins

- 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).

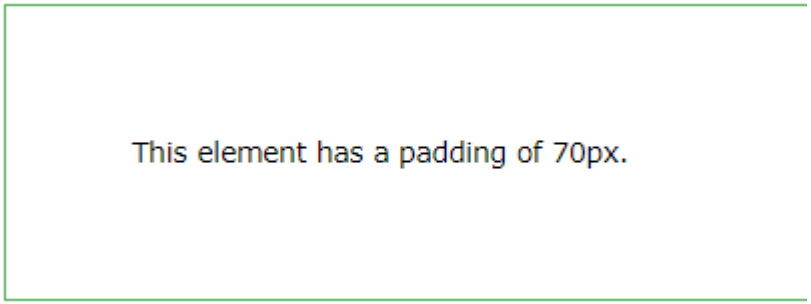
```
p {
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
}
```

CSS Padding

- 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).

```
div {
  padding-top: 50px;
  padding-right: 30px;
  padding-bottom: 50px;
  padding-left: 80px;
}
```



CSS Responsive

- Web pages can be viewed using many different devices: desktops, tablets, and phones. Your web page should look good, and be easy to use, regardless of the device.

- Web pages should not leave out information to fit smaller devices, but rather adapt its content to fit any device

- Check Demo (css_13.html)

CSS Grid

- 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.

```
<div class="grid-container">
  <div class="grid-item">1</div>
  <div class="grid-item">2</div>
  <div class="grid-item">3</div>
  <div class="grid-item">4</div>
  <div class="grid-item">5</div>
  <div class="grid-item">6</div>
  <div class="grid-item">7</div>
  <div class="grid-item">8</div>
  <div class="grid-item">9</div>
</div>
```

1	2	3
4	5	6
7	8	9

Other CSS

- CSS Height/Width
- CSS Box Model
- CSS Outline
- CSS Text
- CSS Fonts
- CSS Icons
- CSS Links
- CSS Tables
- CSS Max-width
- CSS Position

- Check the rest at w3schools.com/css