

CSS(Cascading Style Sheets)

- codepen:<https://codepen.io/>

- Introduction

- visual style and visualization
- Properties: font, text, spacing, layout...
- Example:
 - `h1{` <!--h1 is a selector, after the selector is the declaration block. Together they compose a **CSS Rule**-->
 - `color: blue;`
 - `text-align: center;`
 - `font-size: 20px;` <!--each of style is a declaration with **property:value**-->
 - `}`

- Places Build CSS--Inline/Internal/External

- Inline

-

```
<header>
  <h1 style="color: royalblue"> The Code Magazine</h1>
```

- Internal

-

```
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>The basic language of the web:HTML</title>
  <style>
    h1 {
      color: royalblue;
    }
  </style>
</head>
```

- External (more useful)

- 1)write an external document



```
# style.css
1 h1 {
2   color: royalblue;
3 }
```

- 2)give a link in index.html file, tell the program there exists an external style settings

```
<head>
  <meta charset="UTF-8" />
  <link
    href="/Users/macbookpro/Desktop/03-CSS-Fundamentals/style.css"
    rel="stylesheet"
  />
  <title>The basic language of the web:HTML</title>
</head>
```

- Styling Text (properties)
 - font-size/font-family/text-transform/font-style/line-height/text-align
- Combining Selectors
 - put them together into one list then state the common properties, like a simple combination

```

1  h1,
2  h2,
3  h3,
4  h4,
5  p,
6  li {
7    font-family: "Times New Roman", Times, serif;
8  }

```

- Sets the font-size of all paragraph in footer

```

footer p {
  font-size: 16px;
}

```

- Effect in web page



- [Why JavaScript is Awesome](#)

By Matilda

Copyright © 2027 by the Code Magazine

- Class and ID Selectors

- ID

- simply give some selectors a brief name to avoid the complexity
- .html: <p id="given_name">
- .css: #given_name{font_size=16px;...}
- comment of css
 - /*.....*/
 - in VSCode, we can easily type **command+slash** to give all of content we select as a comment

```

/*article header p {
  font-style: italic;
} */

```

- we can set each id name only once (uniquely)

- Class

- .html: <p class="name">

- .css: .name{...}

```
.related-author {
  font: size 18px;
  font-weight: bold;
}
```

- In real world, we have few times to use ID. While, we use classes usually

- Colors

- RGB Model: red, green, blue

- give every color of RGB a weight of combination~(0,255)

- Define color {color:#1098ab;}

- RGB Notation

- rgb(x,y,z)
- rgba(x,y,z,alpha) alpha:transparency

- Hexadecimal Notation

- from 0 to ff
- example:#00ffff (#rrggbb)
- most used in reality

- Grey:

- if the number of three channels are the same, we get grey color
- all of them close to 0-->black; all of them close to biggest notation--->white

- Background color setting {background-color:#f7f7f7;}

- Border

- size, style, color

```
aside {
  background-color: #f7f7f7;
  border: 5px solid #0198ad;
}
```

- border-right/left/top/bottom

- Pseudo Classes:matching exist html structure

- select all the first child element of parent ol element

```
li:first-child {
  font-weight: bold;
}
```

- first-child, last-child, nth-child(x),nth-child(odd/even)

- Pseudo Elements (inline-element)

- First letter or the first line

- `h1::first-letter{...}` , we use double `:` to represent pseudo elements
- Example: We change the font size of the first letter.

The Basic Language of the Web: HTML

All modern websites and web applications are built using three *fundamental* technologies: HTML, CSS and JavaScript. These are the languages of the web.

- After&Before Pseudo Elements

- `h2::after{content:"Top";}` /*add after pseudo element */
- Example:

```
h2 {
  position: relative;
}

h2::after {
  content: "TOP";
  background-color: yellow;
  color: black;
  font-size: 16px;
  font-weight: bold;
  display: inline-block;
  padding: 5px 10px;
  position: absolute;
  top: -10px;
  right: -100px;
}
```

The Basic Language of the Web: HTML

TOP

All modern websites and web applications are built using three *fundamental* technologies: HTML, CSS and JavaScript. These are the languages of the web.

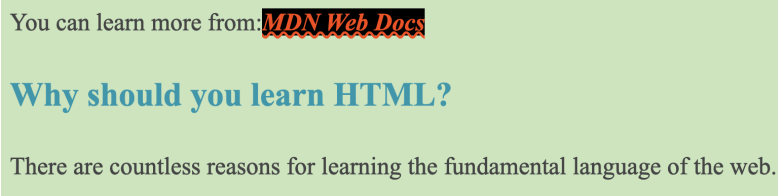
- Adjacency element/sibling selector
 - The part of the same element inside the same container, closely next (the very first child) to the parent selector
 - **`h3 + p::first-line {...}`**
- Style Hyperlinks
 - use pseudo class to style them
 - `a:link {color:#111111;}`
 - link, visited, hover, active

```

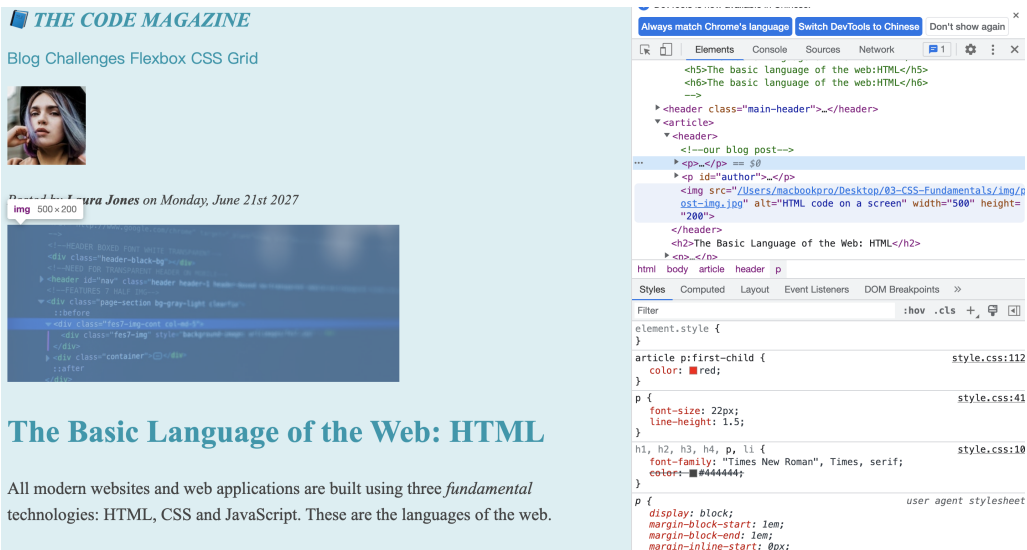
/* style link */
a:link {
  color: #0198ad;
  text-decoration: none; /* get rid of underline */
}
a:visited {
  color: #0198ad;
}
a:hover {
  color: orangered;
  font-weight: bold;
  text-decoration: underline wavy orangered;
}
a:active {
  background-color: black;
  font-style: italic;
}

```

- Result



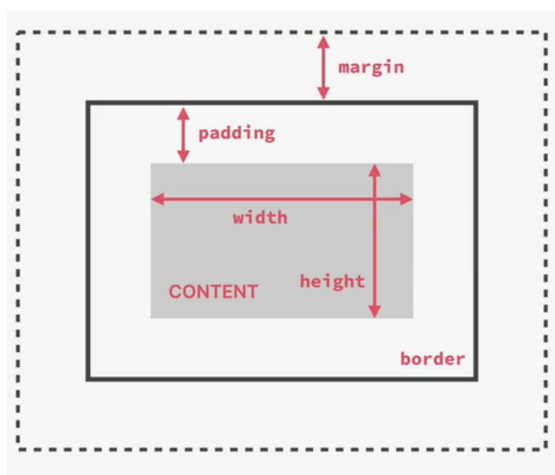
- Google Chrome Dev-tools
- right-click->inspection



- CSS Theory

- 1.Conflicts between Selectors:what happen if multiple css rules apply to the same elements?
 - all rules applied
 - priority: (From highest to lowest)
 - Declaration marked ! important
 - Inline Style
 - ID selectors: if multiple, choose the last one (#)

- class/pseudo class selector (. :)
- element selector(p,div,li...)
- Universal selector
- 2. **Inheritance and the Universal Selector:**
 - Inheritance
 - get from **parent** elements to child
 - Inheritance property is easily overwritten by any rule for the same element. Namely, it has the lowest priority.
 - Universal Selector
 - simply select single element on the page, no inheritance involved
 - *{...}
- 3. **CSS Box Model:** how elements are displayed on a webpage and how they are sized
 - Content: text, images, video, specify both height or width area {height:auto; width:100%;}
 - Border: line around the elements still inside the elements
 - {border-top:10px solid #1098ab }
 - Padding: invisible space, inside the element
 - {padding-right:20px;}
 - Margin: space outside the element
 - {margin-bottom: 10px;}
 - Fill area: can be filled with background color or background image
 - Calculation of element height and width. A default behavior



- Center our page
 - use <div></div> in html to build a container
 - .container {width: 800px; margin:0, auto;}
- 4. **Types of Boxes**
 - Block Level: elements are formatted as blocks, occupy 100% of parent elements' width, change inline boxes->block level boxes, Box model applies

{display:block;}

- Inline Elements:padding and margins are applied only for left&right, height and width not apply, cause no line breaks, only occupy its content space (a, img, strong, em, button...) **{display:inline;}**

- inline-block: Box model applies, only occupy its content space, cause no line breaks **{display:inline-block;}**

- 5. **Absolute Positioning**

- Normal Flow: default positioning, element simply lay out according to the order in html code **{position: relative;}**
- Absolute Positioning: out of flow-element is removed from the normal flow, no impact on surrounding elements, use top/bottom/right/left properties to set a relatively positioned container **{position: absolute;}**

以上内容整理于 [幕布文档](#)