CSS

Permette di definire stili in base al tipo di media (screen, print, speech)
Costituito da una sequenza di regole, ciascuna regola e' composta da una lista di selettori e da un insieme di dichiarazioni

```
<body>
    <h1> Titolo </h1>
</body>
```

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

h1 {
    color:blue;
}
```

Selettori

- * qualsiasi elemento
- E un elemento di tipo E (es. h1)
- E.classname elemento con attributo "class" uguale a "classname"
- E#id elemento con id uguale a "id"
- E F elemento F discendente di E
- E > F elemento F figlio di E
- E + F elemento F immediatamente preceduto da E
- E ~ F elemento F preceduto da E

Selettori: attributi

- E[foo] un elemento di tipo E con attributo "foo"

- E[foo=”bar”] un elemento di tipo E con attributo “foo” uguale a “bar”
- E[foo~=”bar”] … “foo” ha uno dei valori (separati da spazio) uguale a “bar”
- E[foo^=”bar”] … “foo” comincia con “bar”
- E[foo$=”bar”] … “foo” termina con “bar”
- E[foo*=”bar”] … “foo” contiene “bar

Selettori: pseudoclassi

- E:root un elemento E radice del documento (in HTML sempre “html”)
- E:nth-child(an+b) an+b-esimo figlio di E○ even == 2n
  - odd == 2n+1
  - E:nth-child(3) == E:nth-child(0n+3)
- E:nth-last-child(an+b) an+b-ultimo figlio di E
- E:nth-of-type(an+b) an+b elemento di tipo E
- E:nth-last-of-type(an+b) an+b elemento di tipo E contando dall’ultimo
- E:first-child primo figlio di E
- E:last-child ultimo figlio di E
- E:first-of-type primo di tipo E
- E:last-of-type ultimo di tipo E
- E:empty elemento di tipo E senza figli
- E:only-child elemento di tipo E figlio unico
- E:only-of-type elemento di tipo E unico figlio di questo tipo
- E:link hyperlink che non è ancora stato visitato
- E:visited hyperlink già stato visitato
- E:active elemento attivato dall’utente (es. nel tempo che fra la pressione del click ed il rilascio)

- E:hover over di un dispositivo di puntamento (tipicamente il mouse)
- E:focus quando l'elemento ha il focus
- E:first-line E:first-lette

Come usare CSS:

Inline

L'attributo "style" degli elementi HTML contiene dichiarazioni CSS e si applica le regole al singolo element

```
<h1 style="color:blue"> Titolo </h1>
```

Interno

Elemento `<style>` dentro l'`<head>` del documento HTML. Per le regole di stile che vengono usate in un solo document

```
<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-s
    <title>Esempio di CSS nel tag head</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f0f0f0;
            color: #333;
        }
```

```
        h1 {
            color: #007bff;
        }

        p {
            font-size: 16px;
        }

        .container {
            max-width: 800px;
            margin: 0 auto;
            padding: 20px;
            background-color: #fff;
            border-radius: 5px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Benvenuto nel nostro sito</h1>
        <p>Questo è un paragrafo di esempio. Lorem ipsum dolor s
    </div>
</body>
</html>
```

Esterno

Le regole CSS sono scritte dentro un file con estensione ***.css***
Si aggiunge un riferimento al file .css dentro l'elemento `<link>` dell'`<head>`

```
<!DOCTYPE html>
<html lang="it">
```

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-s
    <link rel="stylesheet" href="mystyle.css">
</head>
<body>
    .
    .
    .
</body>
</html>
```

Unità di misura

Molte proprietà modificabili tramite CSS richiedono una misura di lunghezza: (width,margin, padding, font-size, ...)

Una lunghezza viene definita da un numero seguito dalla sua unità di misura (es. 10px)

Unità di misura relative più comuni:

- `em` relativo alla dimensione del font (2em = 2 volte la dimensione del font corrente)
- `rem` relativo alla dimensione del font della root
- `vw`, `vh view width`/`height` relativo all'1% della viewport
- `%` relativo all'elemento padr

Box model

Ogni elemento HTML viene avvolto da un box composto da quattro parti

Content:
il contenuto del box, dove testo e immagini appaiono

Padding:
lo spazio attorno al contenuto (è trasparente)

Border:
il bordo che si trova attorno a padding e contenuto

Margin:
lo spazio all'esterno del bordo (è trasparente

Position

La proprietà position specifica il metodo di posizionamento da usare per un elemento.
La posizione viene determinata dalle dichiarazioni
`top`, `bottom`, `left`, `right`

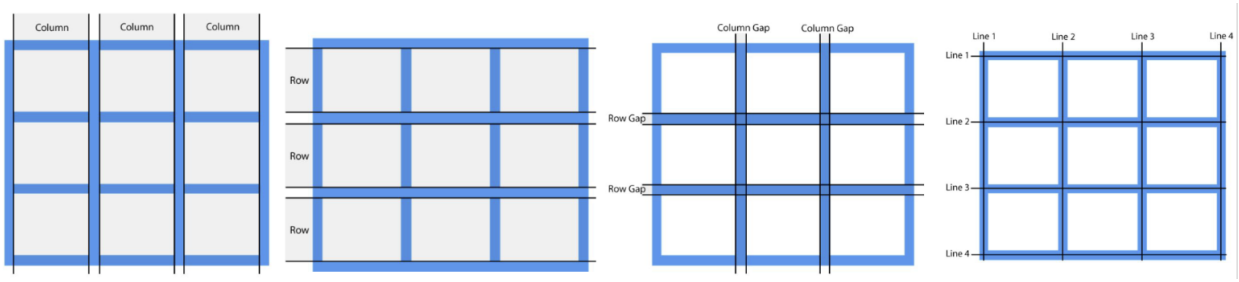
Esistono cinque tipi di posizionamento: `static`, `relative`, `fixed`, `absolute`, `sticky`

- `static` è il posizionamento di default: *top*, *bottom*, *left*, *right* non influenzano il posizionamento ma segue il normale flusso della pagina
- `relative` posiziona l'elemento in maniera relativa al suo normale posizionamento
- `fixed` posiziona l'elemento in relazione alla viewport (indipendentemente dallo scrolling di pagina)
- `absolute` posiziona l'elemento in base al più vicino antenato (spesso in combinazione con padre position relative) o al body se non se ne trova uno

- `sticky` posiziona l'elemento in base alla posizione dello scroll: equivale a relative fino a quando l'elemento sarebbe visibile nella posizione di offset dello scroll, diventa fixed quando l'elemento scomparirebbe

Grid Layout

Si può definire un layout a griglia direttamente in CSS
Una griglia è costituita da un elemento padre con uno o più figli e si definisce usando la dichiarazione
`display: grid;`



Flex

- Permette di distribuire e posizionare gli elementi in maniera agevole
- Semplifica il design di layout responsive
- Ad un elemento padre va assegnata la proprietà "display:flex"
- Si può agire su:
  - direzione (row, column) con flex-direction
  - comportamento di andata a capo wrap/nowrap

    - allineamento orizzontale del contenuto con justify-content
    - allineamento verticale del contenuto con align-content
- I figli di un elemento flex possono essere posizionati cambiando il loro ordine,

Animazioni

Anche le animazioni degli elementi fanno parte di CSS

Un'animazione è il risultato dell'interpolazione dei valori delle proprietà di un elementi

Sono definite tramite `@keyframe` ed esistono diverse funzioni di interpolazione:

- `linear`
- `ease`, `ease-in`, `ease-out`
- `cubic-bezier`