

# Introduzione a CSS - Parte 2

## Il ritorno dello stilista

Prof. Federico Dossena



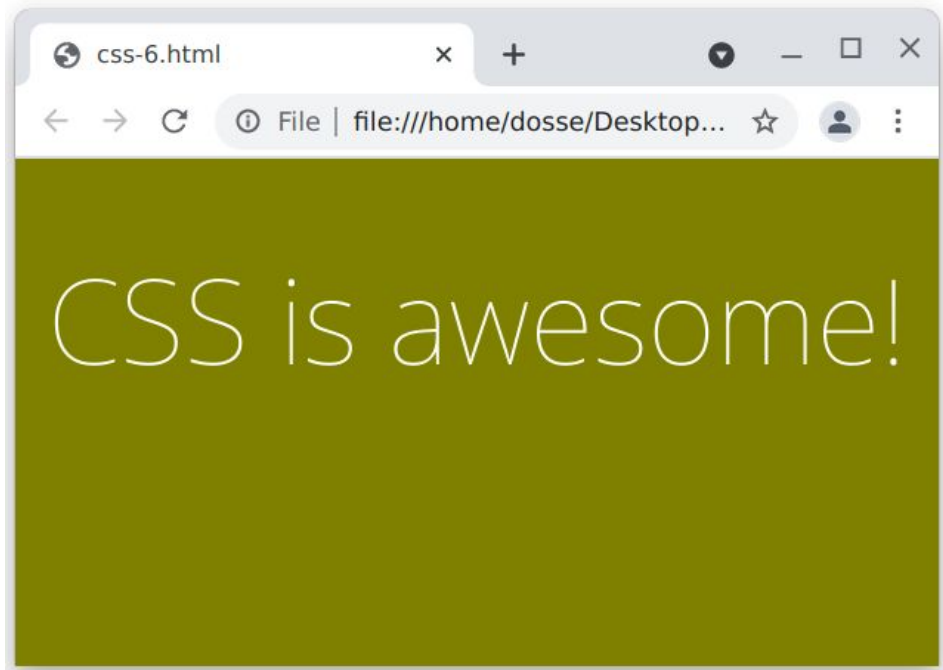
# Parte 2

- Cosa vediamo?
  - Sfondi
  - Colori personalizzati
  - Immagini
  - Tabelle

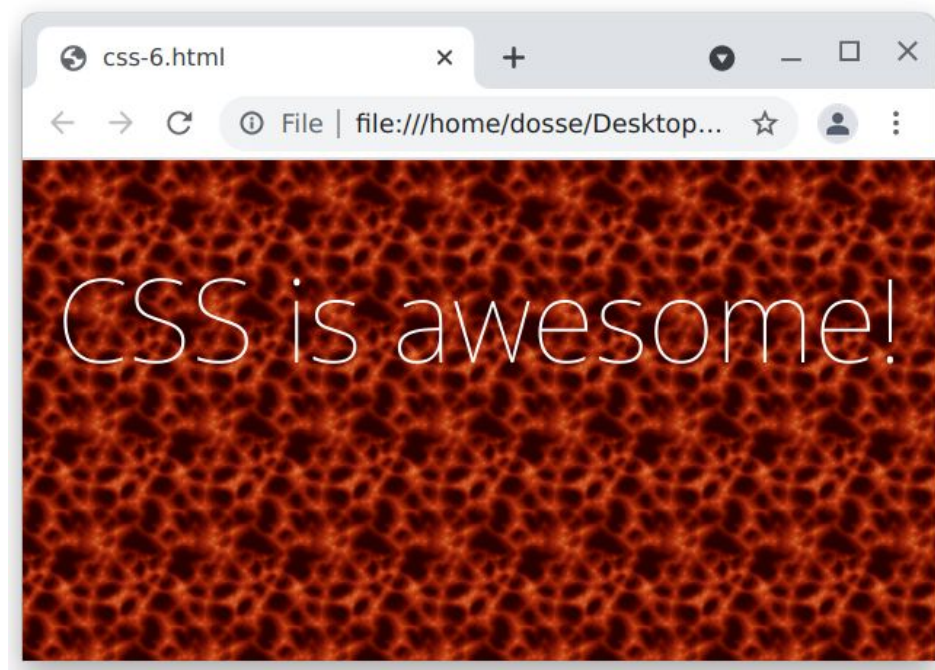
# Sfondi

- La proprietà background applicata al body permette di aggiungere uno sfondo alla pagina
- Lo sfondo può essere un semplice colore  
`background: darkblue;`
- ...oppure un'immagine  
`background: url('immagini/sfondo.png');`
- ...e molto altro, ma non lo vedremo qui

# Esempio



```
body{  
    background: olive;  
}
```

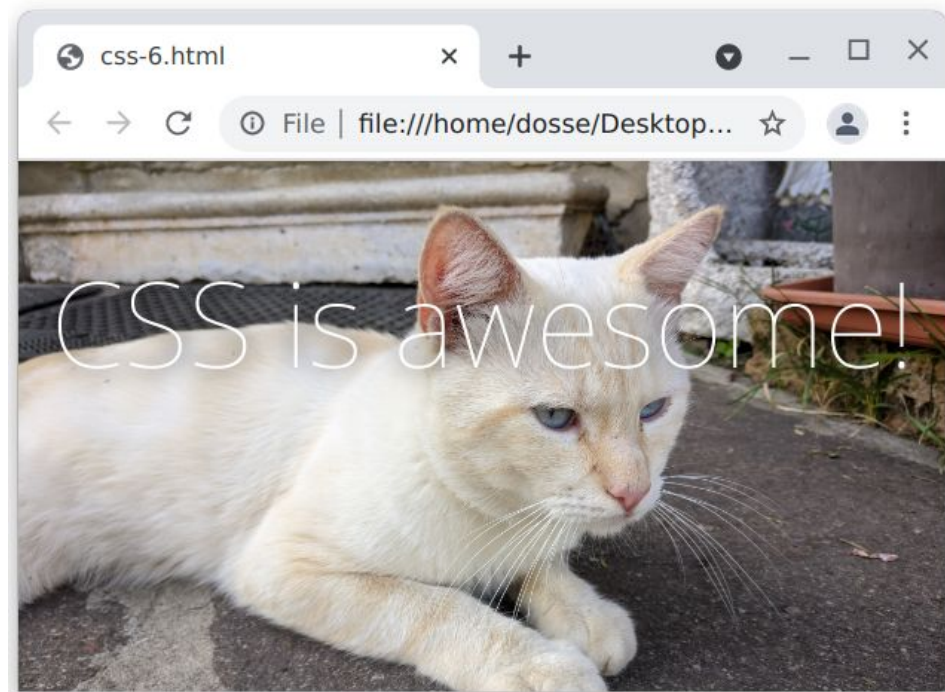


```
body{  
    background:url('lava.png');  
}
```

# Sfondi

- Se si vuole inserire un'immagine in modo che copra tutto lo spazio anzichè ripetersi, si può usare questa regola

```
body{  
    background:url('gatto.jpg');  
    background-size:cover;  
    background-attachment:fixed;  
}
```



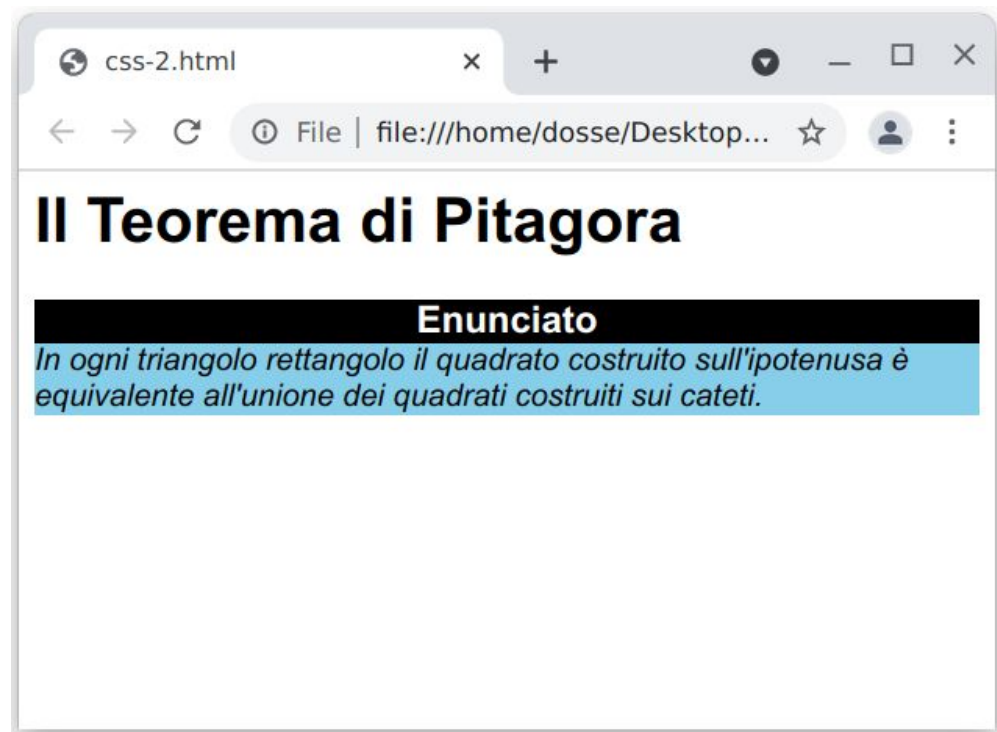
# Sfondi

- Attenzione:
  - L'utilizzo di immagini di sfondo può rendere poco leggibile il testo. Si può renderlo più leggibile aggiungendo una leggera ombreggiatura sotto al testo con  
`text-shadow:0 0 0.2rem black;`
  - Immagini molto grandi (più di 1-2 MB) potrebbero rendere lento il caricamento della pagina via internet, per cui è meglio comprimerle prima

# Sfondi

- Gli sfondi si possono applicare anche ad altri elementi oltre al body, ad esempio div e paragrafi

```
body{
  font-family:sans-serif;
}
div.riquadro{
  background:skyblue;
}
div.riquadro > div.titolo{
  font-size:1.2em;
  font-weight:bold;
  background:black;
  color:white;
  text-align:center;
}
div.riquadro > div.corpo{
  font-style:italic;
}
```



# Colori personalizzati

- Oltre ai colori predefiniti come red, yellow, blue, ecc., CSS consente di utilizzare dei colori propri definiti con la notazione RGB, RGBA, HSL, e altre
- Ci concentriamo sulla notazione RGB, che è quella più utilizzata
- I **colori RGB** possono essere scritti in due modi:
  - **Per esteso**, ad esempio: `rgb(255,0,0)` (che è rosso)
  - Con la **notazione compatta** esadecimale, ad esempio: `#FF0000` (sempre rosso)

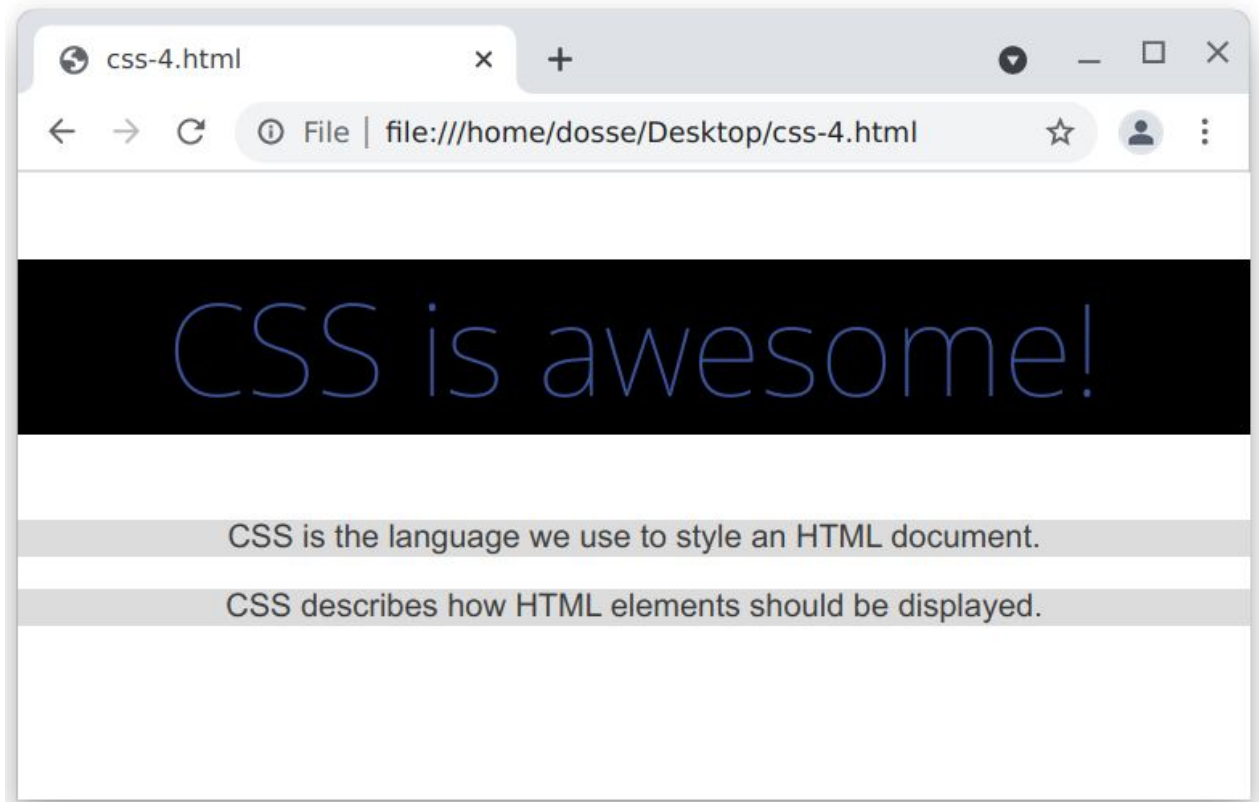


# Colori personalizzati

- Poichè non conoscete ancora le conversioni in esadecimale, utilizzeremo quasi sempre la notazione estesa
- **Mescolando i tre colori primari rosso (R), verde (G) e blu (B) è possibile creare qualsiasi colore visibile all'occhio umano**
- Attenzione: la luce è additiva, non è sottrattiva come le tempere!
- Per imparare, questo strumento è molto utile:  
[https://www.w3schools.com/colors/colors\\_picker.asp](https://www.w3schools.com/colors/colors_picker.asp)  
(oppure anche il selettore dei colori di Paint)

# Esempio

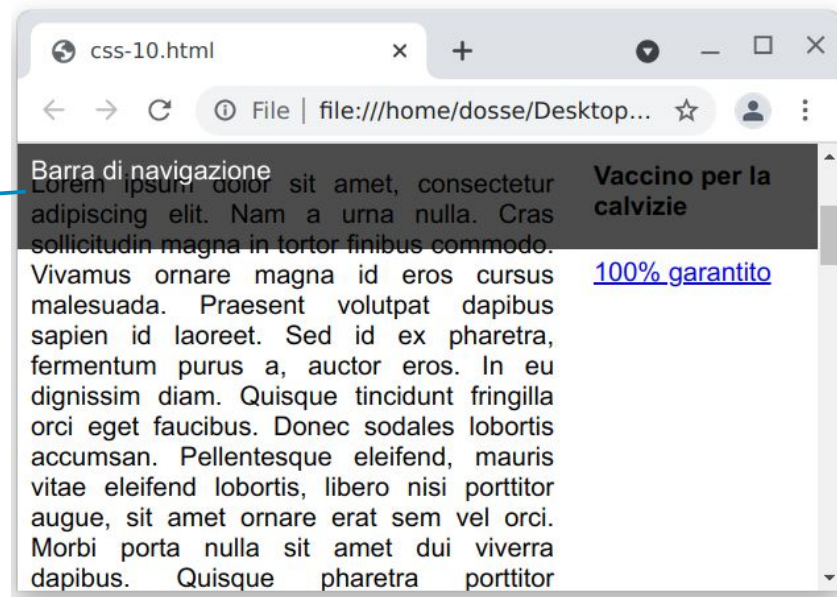
```
h1{
  font-family:"Noto Sans";
  font-size:4rem;
  font-weight:lighter;
  text-align:center;
  color:#3B539B;
  background:#000000;
}
body{
  color:rgb(60,60,60);
  font-family:sans-serif;
  margin:0;
}
p{
  text-align:center;
  background:rgb(220,220,220);
}
```



# Colori personalizzati

- È possibile anche creare colori semitrasparenti (utili per gli sfondi!) utilizzando la notazione RGBA, ad esempio `rgba(255,0,0,0.3)` è un rosso ma opaco al 30%

```
#navigazione{  
  background:rgba(0,0,0,0.7);  
  color:white;  
}
```



# Un po' di colori

| Colore  | R   | G   | B   | Descrizione |
|---|-----|-----|-----|-------------|
|    | 255 | 255 | 255 | Bianco      |
|    | 0   | 0   | 0   | Nero        |
|    | 255 | 0   | 0   | Rosso       |
|    | 0   | 255 | 0   | Verde       |
|    | 0   | 0   | 255 | Blu         |
|    | 255 | 255 | 0   | Giallo      |
|    | 0   | 255 | 255 | Ciano       |
|    | 255 | 0   | 255 | Magenta     |
|   | 255 | 128 | 128 | Rosa        |
|  | 128 | 128 | 255 | Azzurro     |
|  | 255 | 128 | 0   | Arancio     |

| Colore  | R   | G   | B   | Descrizione   |
|---|-----|-----|-----|---------------|
|    | 0   | 128 | 128 | Verde acqua   |
|    | 128 | 0   | 128 | Viola         |
|    | 102 | 0   | 255 | Indaco        |
|    | 90  | 50  | 140 | Indaco        |
|    | 150 | 0   | 50  | Porpora       |
|    | 0   | 0   | 128 | Blu navy      |
|    | 65  | 105 | 225 | Blu royale    |
|    | 59  | 89  | 152 | Facebook      |
|   | 29  | 161 | 242 | Twitter       |
|  | 255 | 153 | 0   | Amazon        |
|  | 238 | 238 | 238 | Grigio chiaro |

# Immagini

- Le principali proprietà CSS utili per le immagini sono:

| Proprietà | Valori possibili   | Significato                        |
|-----------|--------------------|------------------------------------|
| width     | misura oppure auto | Imposta la larghezza dell'immagine |
| height    | misura oppure auto | Imposta l'altezza dell'immagine    |

- Fissando una delle due proprietà ma lasciando l'altra su auto, l'immagine viene ridimensionata mantenendo le proporzioni corrette

# Esempio

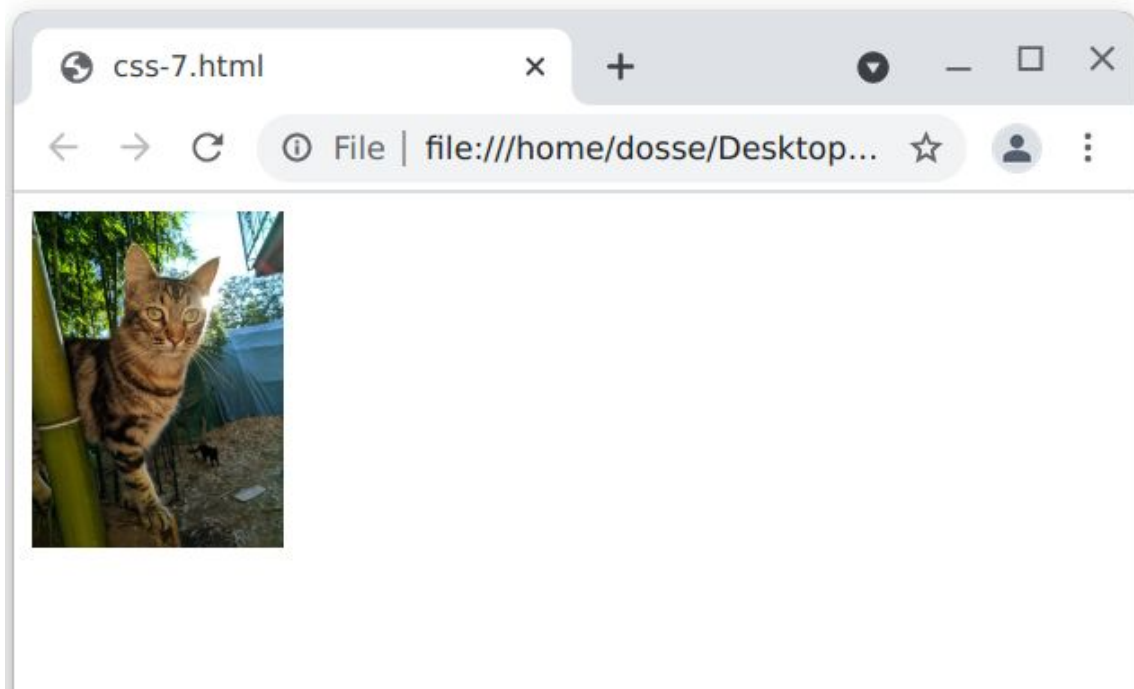
```
img.piccola{  
    width:7rem;  
    height:auto;  
}
```

Giusto!

```
<body>
```

```
  
```

```
</body>
```



# Esempio

```
img.piccola{  
  width:7rem;  
  height:7rem;  
}
```

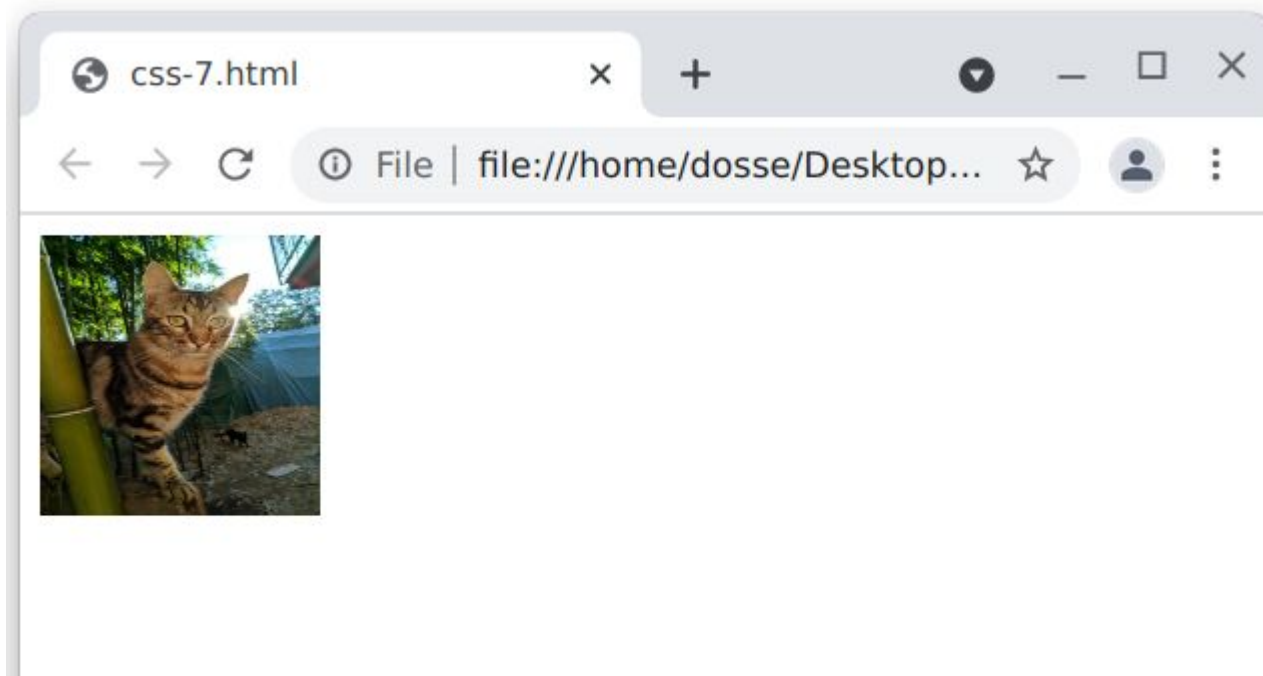
Sbagliato!

L'immagine è schiacciata

```
<body>
```

```
  
```

```
</body>
```



# Immagini

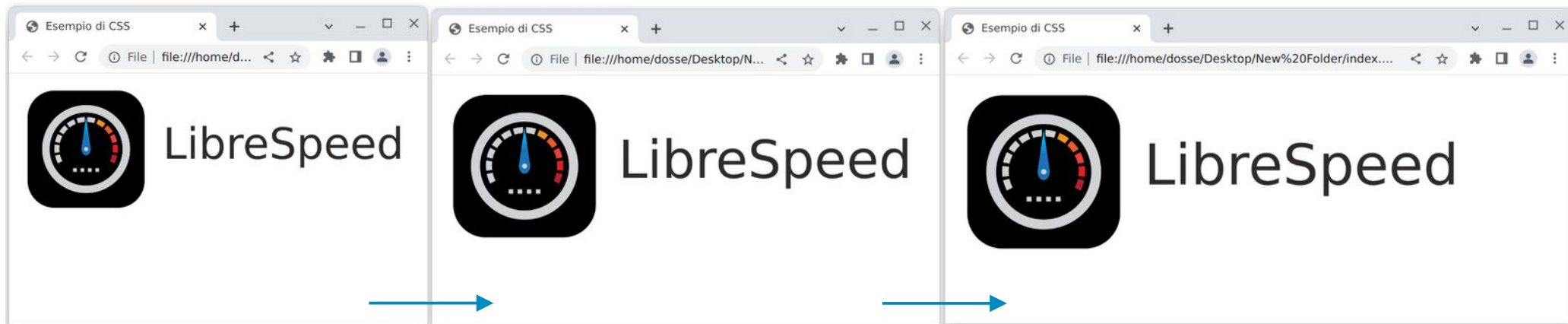
- Altre proprietà CSS utili per le immagini sono:

| Proprietà  | Valori possibili   | Significato   |
|------------|--------------------|---|
| max-width  | misura oppure auto | Imposta la larghezza massima dell'immagine (da usare in combinazione con width) |
| max-height | misura oppure auto | Imposta l'altezza massima dell'immagine (da usare in combinazione con height)   |

- Usando queste proprietà possiamo far scalare un'immagine fino a un limite massimo a nostro piacere



# Esempio



```
#logo{  
  width:100%;  
  max-width:40rem;  
  height:auto;  
}
```

L'immagine con id logo si può allargare fino a un massimo di 40rem, poi non si allarga più e viene semplicemente aggiunto dello spazio vuoto.

# Immagini

- Per centrare un'immagine esistono vari „trucchi“ in CSS, ad esempio:

```
img.centrata{  
  display:block;  
  margin:0 auto;  
}
```

Fa sì che l'immagine occupi un'intera riga, andando a capo automaticamente prima e dopo

Imposta un margine automatico a destra e a sinistra dell'immagine, centrandola sulla riga

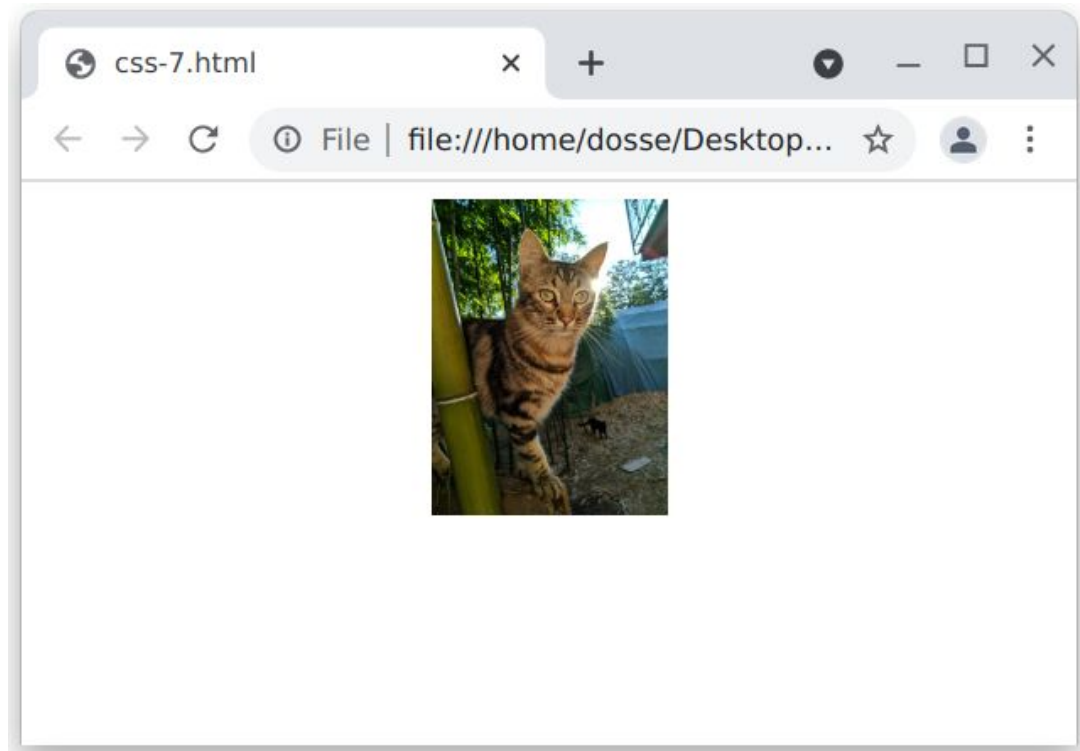
# Esempio

```
img.piccola{  
    width:7rem;  
    height:auto;  
}  
img.centrata{  
    display:block;  
    margin:0 auto;  
}
```

```
<body>
```

```
    
```


```
</body>
```



# Vi siete accorti?

- All'immagine ho applicato sia la classe piccola che la classe centrata
- Posso applicare tutte le classi che voglio, separandole con uno spazio all'interno dell'attributo class

```
<body>  
    
</body>
```

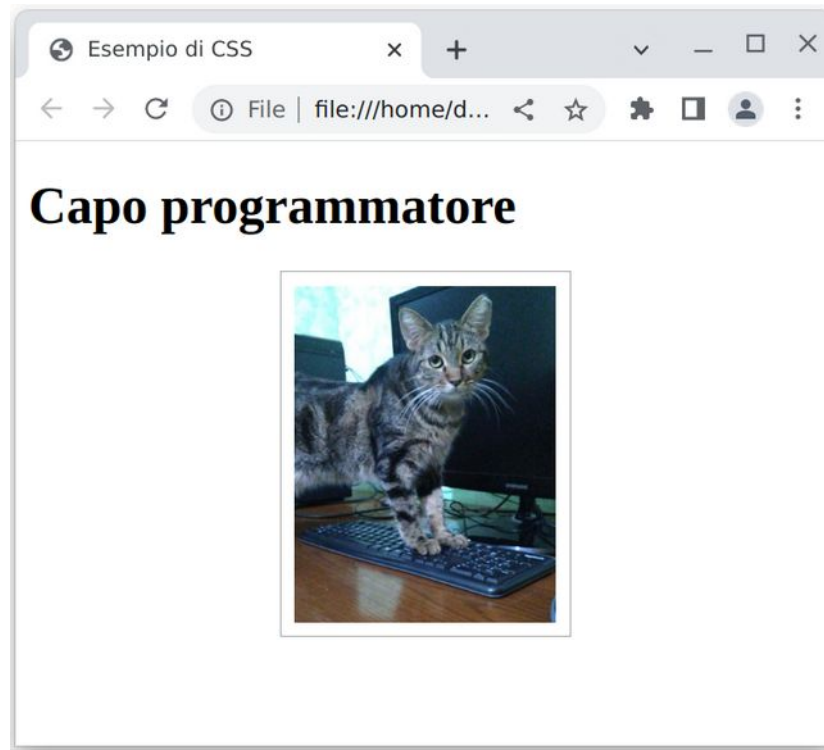


# Esempio

- Usando le proprietà border e padding (che vedremo meglio nella sezione sul box model) possiamo inserire una semplice cornice attorno alle immagini

```
img.centrata{  
  display:block;  
  margin:0 auto;  
}  
img.piccola{  
  width:11rem;  
  height:auto;  
}  
img.incorniciata{  
  border:1px solid darkgray;  
  padding:0.5rem;  
  box-sizing:border-box;  
}
```

```
<body>  
  <h1>Capo programmatore</h1>  
    
</body>
```



# Tabelle

- Lo stile delle tabelle può essere personalizzato manipolando l'aspetto dei tag `table`, `tr`, `td` e `th`
- Ad esempio, per inserire i bordi (quelli che in HTML4 si facevano con `border="1"`), possiamo aggiungere questa regola:

```
table, tr, td, th{  
    border: 1px solid black;  
}
```

# Esempio

```
table, tr, td, th{  
    border:1px solid black;  
}
```

|               |          |          |          |          |
|---------------|----------|----------|----------|----------|
| <b>Riga A</b> | Cella 1  | Cella 2  | Cella 3  | Cella 4  |
| <b>Riga B</b> | Cella 5  | Cella 6  | Cella 7  | Cella 8  |
| <b>Riga C</b> | Cella 9  | Cella 10 | Cella 11 | Cella 12 |
| <b>Riga D</b> | Cella 13 | Cella 14 | Cella 15 | Cella 16 |
| <b>Riga E</b> | Cella 17 | Cella 18 | Cella 19 | Cella 20 |

# Esempio

```
table, tr, td, th{  
    border: 1px solid gray;  
}  
tr:nth-child(2n){  
    background: rgb(200, 200, 200);  
}  
tr:nth-child(2n+1){  
    background: rgb(240, 240, 240);  
}  
tr:last-child{  
    background: yellow;  
}
```

|               |          |          |          |          |
|---------------|----------|----------|----------|----------|
| <b>Riga A</b> | Cella 1  | Cella 2  | Cella 3  | Cella 4  |
| <b>Riga B</b> | Cella 5  | Cella 6  | Cella 7  | Cella 8  |
| <b>Riga C</b> | Cella 9  | Cella 10 | Cella 11 | Cella 12 |
| <b>Riga D</b> | Cella 13 | Cella 14 | Cella 15 | Cella 16 |
| <b>Riga E</b> | Cella 17 | Cella 18 | Cella 19 | Cella 20 |



# Esempio

```
tr,td,th{
    border:1px solid red;
}
td,th{
    padding:0.25em;
}
table{
    border-spacing:0;
    border:0.1em solid black;
}
tr:nth-child(2n){
    background:rgb(200,200,200);
}
tr:nth-child(2n+1){
    background:rgb(240,240,240);
}
```

|               |          |          |          |          |
|---------------|----------|----------|----------|----------|
| <b>Riga A</b> | Cella 1  | Cella 2  | Cella 3  | Cella 4  |
| <b>Riga B</b> | Cella 5  | Cella 6  | Cella 7  | Cella 8  |
| <b>Riga C</b> | Cella 9  | Cella 10 | Cella 11 | Cella 12 |
| <b>Riga D</b> | Cella 13 | Cella 14 | Cella 15 | Cella 16 |
| <b>Riga E</b> | Cella 17 | Cella 18 | Cella 19 | Cella 20 |

# Esempio

```
table, tr, td, th{  
    border: 1px solid darkgray;  
}  
table{  
    border-spacing: 0;  
}  
td, th{  
    padding: 0.25em;  
}  
th{  
    background: black;  
    color: white;  
}  
tr:first-child th{  
    background: yellow;  
    color: black;  
}  
tr:last-child{  
    background: lightblue;  
}  
tr:hover{  
    background: pink;  
}
```

|               |          |          |          |          |
|---------------|----------|----------|----------|----------|
| <b>Riga A</b> | Cella 1  | Cella 2  | Cella 3  | Cella 4  |
| <b>Riga B</b> | Cella 5  | Cella 6  | Cella 7  | Cella 8  |
| <b>Riga C</b> | Cella 9  | Cella 10 | Cella 11 | Cella 12 |
| <b>Riga D</b> | Cella 13 | Cella 14 | Cella 15 | Cella 16 |
| <b>Riga E</b> | Cella 17 | Cella 18 | Cella 19 | Cella 20 |

# Esercizio

La tabella deve essere:

- Senza spazio tra le celle
- Bordo skyblue di 1px
- Celle th nere con testo bianco
- Celle td pari di colore lightgray

| <b>Riga 1</b> | Cella 1 | Cella 2 | Cella 3 | Cella 4 |
|---------------|---------|---------|---------|---------|
| <b>Riga 2</b> | Cella 1 | Cella 2 | Cella 3 | Cella 4 |
| <b>Riga 3</b> | Cella 1 | Cella 2 | Cella 3 | Cella 4 |
| <b>Riga 4</b> | Cella 1 | Cella 2 | Cella 3 | Cella 4 |
| <b>Riga 5</b> | Cella 1 | Cella 2 | Cella 3 | Cella 4 |

# Esercizio

```
table, tr, td, th{
    border:1px solid skyblue;
}
table{
    border-spacing:0;
}
td:nth-child(2n){
    background:lightgray;
}
th{
    background:black;
    color:white;
}
```

|               |         |         |         |         |
|---------------|---------|---------|---------|---------|
| <b>Riga 1</b> | Cella 1 | Cella 2 | Cella 3 | Cella 4 |
| <b>Riga 2</b> | Cella 1 | Cella 2 | Cella 3 | Cella 4 |
| <b>Riga 3</b> | Cella 1 | Cella 2 | Cella 3 | Cella 4 |
| <b>Riga 4</b> | Cella 1 | Cella 2 | Cella 3 | Cella 4 |
| <b>Riga 5</b> | Cella 1 | Cella 2 | Cella 3 | Cella 4 |

# Tabelle

- Le tabelle possono essere centrate un „trucco“ simile a quello che usiamo per le immagini

```
table{  
    margin:0 auto;  
}
```

- **Attenzione: non usare display:block sulle tabelle o sulle celle**

# To be continued

- Scarica il terzo pacco di slide per continuare