

CSS

CASCADING
STYLE
SHEETS

CSS .





CSS

CSS È UNO DEI PILASTRI FONDAMENTALI DEL WEB MODERNO. CON CSS, POSSIAMO SEPARARE LA STRUTTURA E IL CONTENUTO DEL NOSTRO SITO WEB DAL SUO ASPETTO VISIVO.





CSS

È UN **LINGUAGGIO DI STILE** CHE DESCRIVE COME GLI ELEMENTI HTML DEVONO ESSERE VISUALIZZATI SULLO SCHERMO, SU STAMPA O IN ALTRI DISPOSITIVI. UTILIZZANDO REGOLE CSS, POSSIAMO SPECIFICARE COLORI, FONT, DIMENSIONI, POSIZIONI, BORDI E MOLTO ALTRO ANCORA PER QUALSIASI ELEMENTO DELLA PAGINA WEB.

È STATO SVILUPPATO PER RISOLVERE PROBLEMI DI DESIGN COMPLESSI E RIPETITIVI, COME LA CREAZIONE DI STILI COERENTI PER TESTI, MENU DI NAVIGAZIONE, TABELLE E LAYOUT A GRIGLIA. GRAZIE A CSS, POSSIAMO CREARE PAGINE WEB MODERNE, RESPONSIVE E ACCESSIBILI, CHE SI ADATTANO A QUALSIASI DISPOSITIVO E UTENTE.





BREVI CENNI STORICI

CSS È STATO INTRODOTTO PER LA PRIMA VOLTA NEL DICEMBRE 1996 DAL **WORLD WIDE WEB CONSORTIUM (W3C)** COME UN MODO PER SEPARARE LA PRESENTAZIONE VISIVA DEL CONTENUTO HTML.

INIZIALMENTE, CSS ERA LIMITATO A POCHE PROPRIETÀ, COME COLORI, FONT E ALLINEAMENTO DEL TESTO. NEL CORSO DEGLI ANNI, TUTTAVIA, IL CSS È STATO ESTESO CON NUOVE PROPRIETÀ E FUNZIONALITÀ, RENDENDOLO UNO STRUMENTO ESSENZIALE PER LA CREAZIONE DI PAGINE WEB MODERNE.





BREVI CENNI STORICI: CSS2 & CSS3

LA VERSIONE **CSS2** È STATA RILASCIATA NEL 1998 E HA AGGIUNTO NUOVE FUNZIONALITÀ, COME LAYOUT A GRIGLIA, POSIZIONAMENTO DEI LIVELLI, EFFETTI DI TRANSIZIONE E ANIMAZIONE.

NEL 2001 È STATO RILASCIATO IL PRIMO STANDARD **CSS3**, CHE HA AGGIUNTO MOLTE ALTRE FUNZIONALITÀ AVANZATE, COME SFUMATURE, BORDI ARROTONDATI, TRASFORMAZIONI, MEDIA QUERY E SELEZIONI AVANZATE.



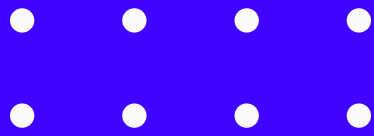


BREVI CENNI STORICI: OGGI

OGGI, CSS È UNO DEGLI STANDARD PIÙ IMPORTANTI DEL WEB MODERNO E VIENE UTILIZZATO DA TUTTI I PRINCIPALI BROWSER PER RENDERE LE PAGINE WEB VISIVAMENTE ACCATTIVANTI, RESPONSIVE E ACCESSIBILI.

INOLTRE, CSS È DIVENTATO UN PILASTRO FONDAMENTALE DEL DESIGN WEB E VIENE UTILIZZATO DA DESIGNER E SVILUPPATORI IN TUTTO IL MONDO PER CREARE PAGINE WEB SOFISTICATE E DI ALTA QUALITÀ.





SEPARARE CONTENUTO E STILE





SEPARARE CONTENUTO E STILE

UNO DEI PRINCIPI FONDAMENTALI DEL DESIGN WEB MODERNO È LA **SEPARAZIONE DEI CONTENUTI DALLO STILE**. QUESTO SIGNIFICA CHE I CONTENUTI DI UNA PAGINA WEB (HTML) E IL LORO ASPETTO VISIVO (CSS) DEVONO ESSERE SEPARATI, IN MODO CHE LE MODIFICHE APPORTATE ALLO STILE NON INFLUISCANO SUI CONTENUTI E VICEVERSA.

CSS CI CONSENTE DI SEPARARE LO STILE DAI CONTENUTI IN TRE MODI PRINCIPALI:

1. UTILIZZANDO **FILE CSS ESTERNI**
2. UTILIZZANDO **STILI INTERNI**
3. UTILIZZANDO **STILI INLINE**





SEPARARE CONTENUTO E STILE: FILE CSS ESTERNI

CREARE UN FILE CSS SEPARATO E COLLEGARLO ALLA PAGINA HTML UTILIZZANDO UN ELEMENTO `<LINK>`.

IN QUESTO MODO, TUTTO LO STILE VIENE DEFINITO NEL FILE CSS, MENTRE LA PAGINA HTML CONTIENE SOLO IL CONTENUTO.





SEPARARE CONTENUTO E STILE: STILI INTERNI

DEFINIRE I NOSTRI STILI ALL'INTERNO DEL TAG
<HEAD> DELLA PAGINA HTML, UTILIZZANDO
L'ELEMENTO **<STYLE>**.

QUESTO METODO CONSENTE DI DEFINIRE GLI STILI
DIRETTAMENTE ALL'INTERNO DELLA PAGINA HTML, MA
ANCORA SEPARATI DAL CONTENUTO EFFETTIVO.



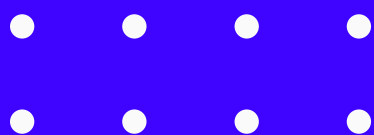


SEPARARE CONTENUTO E STILE: STILI INLINE

DEFINIRE GLI STILI DIRETTAMENTE ALL'INTERNO DEI TAG HTML UTILIZZANDO L'ATTRIBUTO "**STYLE**".

QUESTO METODO È MENO RACCOMANDATO, IN QUANTO GLI STILI SONO MISTI CON I CONTENUTI E RENDONO LA MANUTENZIONE PIÙ DIFFICILE.





SINTASSI





SINTASSI

LA SINTASSI CSS È COMPOSTA DA TRE PARTI
PRINCIPALI:
IL **SELETTORE**, LA **PROPRIETÀ** E IL **VALORE**.





SINTASSI: SELETTORE

IL **SELETTORE** IDENTIFICA GLI ELEMENTI HTML AI QUALI VOGLIAMO APPLICARE UNO STILE. POSSIAMO SELEZIONARE GLI ELEMENTI HTML IN BASE AL LORO TIPO, CLASSE, ID O GERARCHIA.

```
h1 {  
  /* selettore per tutti gli elementi h1 */  
}  
  
.my-class {  
  /* selettore per tutti gli elementi con la classe "my-class" */  
}  
  
#my-id {  
  /* selettore per l'elemento con l'ID "my-id" */  
}  
  
.parent .child {  
  /* selettore per tutti gli elementi con la classe "child" che  
sono figli di un elemento con la classe "parent" */  
}
```





SINTASSI: PROPRIETÀ

LA **PROPRIETÀ** SPECIFICA QUALE ASPETTO VISIVO VOGLIAMO APPLICARE ALL'ELEMENTO SELEZIONATO.



Esempio

```
color: red;  
font-size: 16px;  
background-color: yellow;
```





SINTASSI: VALORE

IL **VALORE** SPECIFICA IL VALORE DELL'ASPETTO VISIVO CHE VOGLIAMO APPLICARE ALLA PROPRIETÀ.
AD ESEMPIO:



Esempio

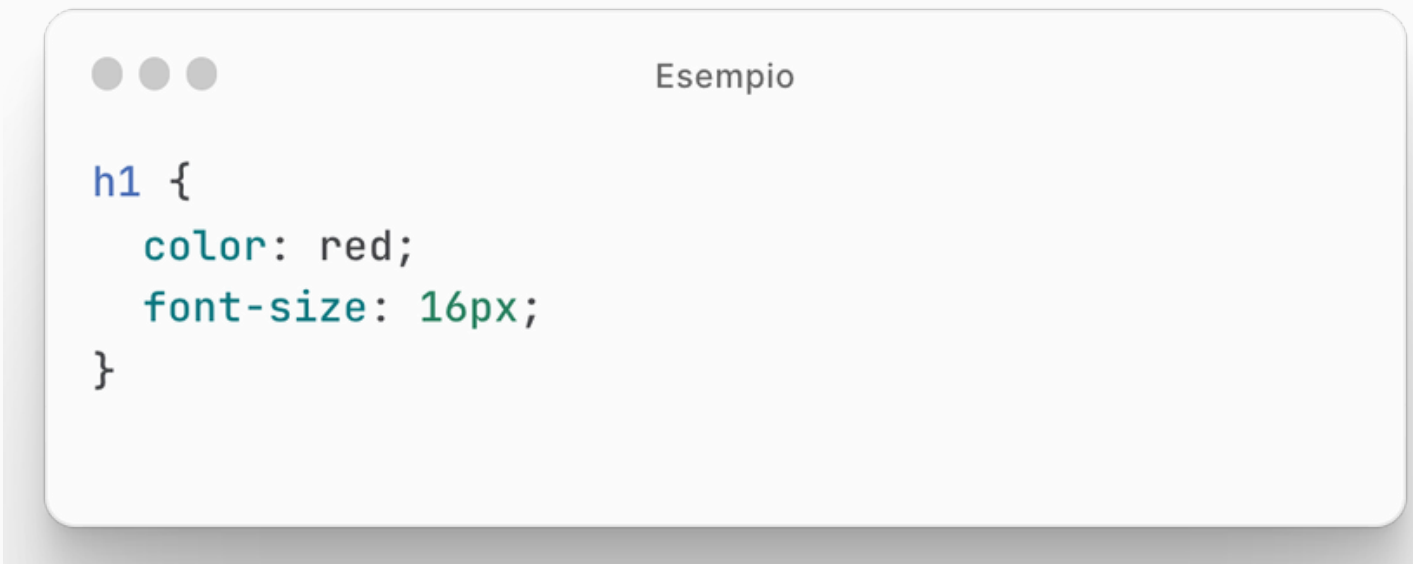
```
color: red; /* il valore è "red" */  
font-size: 16px; /* il valore è "16px" */  
background-color: yellow; /* il valore è "yellow" */
```





SINTASSI: PARENTESI GRAFFE

INOLTRE, LE REGOLE CSS SONO DELIMITATE DALLE PARENTESI GRAFFE. AD ESEMPIO:

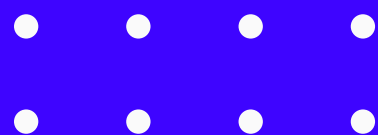


```
h1 {  
  color: red;  
  font-size: 16px;  
}
```

IL SELETTORE È "H1", LE PROPRIETÀ SONO "COLOR" E "FONT-SIZE" E I VALORI SONO "RED" E "16PX".

LE PROPRIETÀ SONO SEPARATE DAI VALORI DA DUE PUNTI E OGNI COPPIA PROPRIETÀ-VALORE È SEPARATA DA UN PUNTO E VIRGOLA.





ANALISI DEGLI STILI PRINCIPALI





ANALISI DEGLI STILI PRINCIPALI

GLI **STILI DEL TESTO** E GLI **STILI PER LO SFONDO** SONO DUE DEI PRINCIPALI TIPI DI STILI CSS CHE POSSIAMO UTILIZZARE PER PERSONALIZZARE L'ASPETTO VISIVO DEL NOSTRO SITO WEB.





ANALISI DEGLI STILI PRINCIPALI: STILI DEL TESTO

GLI STILI DEL TESTO CI PERMETTONO DI PERSONALIZZARE L'ASPETTO DEL TESTO ALL'INTERNO DEGLI ELEMENTI HTML. ALCUNE DELLE PROPRIETÀ PRINCIPALI DEGLI STILI DEL TESTO INCLUDONO:

- **FONT-FAMILY**: DEFINISCE IL TIPO DI CARATTERE DEL TESTO
- **FONT-SIZE**: DEFINISCE LA DIMENSIONE DEL CARATTERE
- **FONT-WEIGHT**: DEFINISCE LO SPESSORE DEL CARATTERE
- **COLOR**: DEFINISCE IL COLORE DEL TESTO
- **TEXT-ALIGN**: DEFINISCE L'ALLINEAMENTO DEL TESTO

QUESTE SONO SOLO ALCUNE DELLE PROPRIETÀ DEGLI STILI DEL TESTO CHE POSSIAMO UTILIZZARE. POSSIAMO ANCHE PERSONALIZZARE L'ASPETTO DEL TESTO CON PROPRIETÀ COME TEXT-DECORATION, TEXT-TRANSFORM E LETTER-SPACING, TRA GLI ALTRI.



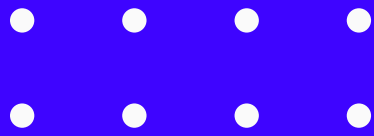


ANALISI DEGLI STILI PRINCIPALI: STILI PER LO SFONDO

GLI STILI PER LO SFONDO CI PERMETTONO DI PERSONALIZZARE LO SFONDO DEGLI ELEMENTI HTML. ALCUNE DELLE PROPRIETÀ PRINCIPALI DEGLI STILI PER LO SFONDO INCLUDONO:

- **BACKGROUND-COLOR**: DEFINISCE IL COLORE DELLO SFONDO
- **BACKGROUND-IMAGE**: DEFINISCE UN'IMMAGINE DA UTILIZZARE COME SFONDO
- **BACKGROUND-REPEAT**: DEFINISCE SE L'IMMAGINE DI SFONDO DEVE ESSERE RIPETUTA ORIZZONTALMENTE E/O VERTICALMENTE
- **BACKGROUND-SIZE**: DEFINISCE LA DIMENSIONE DELL'IMMAGINE DI SFONDO
- **BACKGROUND-POSITION**: DEFINISCE LA POSIZIONE DELL'IMMAGINE DI SFONDO ALL'INTERNO DELL'ELEMENTO HTML





CLASSI E PSEUDOCASSI





CLASSI E PSEUDOCASSI

LE **CLASSI** E LE **PSEUDOCASSI** SONO DUE STRUMENTI IMPORTANTI CHE CI CONSENTONO DI APPLICARE STILI PERSONALIZZATI AI NOSTRI ELEMENTI HTML IN MODO PRECISO E SELETTIVO.





CLASSI E PSEUDOCCLASSI: CLASSI

LE **CLASSI CSS** SONO IDENTIFICATORI PERSONALIZZATI CHE POSSIAMO UTILIZZARE PER APPLICARE UNO STILE SPECIFICO A UNO O PIÙ ELEMENTI HTML. PER DEFINIRE UNA CLASSE CSS, DOBBIAMO UTILIZZARE IL SELETTORE DI CLASSE "." SEGUITO DAL NOME DELLA CLASSE.

IN QUESTO ESEMPIO, ABBIAMO DEFINITO UNA CLASSE CSS CHIAMATA "**MY-CLASS**" CHE APPLICA IL COLORE ROSSO E LA DIMENSIONE DEL FONT DI 16PX A TUTTI GLI ELEMENTI HTML CHE UTILIZZANO QUESTA CLASSE.

● ● ● Eempio

```
.my-class {  
  color: red;  
  font-size: 16px;  
}
```





CLASSI E PSEUDOCASSI: PSEUDOCASSI

LE **PSEUDOCASSI CSS** SONO INVECE UTILIZZATE PER SELEZIONARE UN ELEMENTO HTML IN UN DETERMINATO STATO O POSIZIONE. AD ESEMPIO, LA PSEUDOCASSE ":HOVER" VIENE UTILIZZATA PER APPLICARE UNO STILE A UN ELEMENTO HTML QUANDO IL CURSORE DEL MOUSE È SOPRA DI ESSO. ALTRE PSEUDOCASSI COMUNI INCLUDONO ":ACTIVE", ":FOCUS" E ":VISITED".

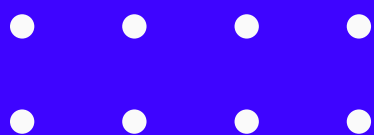


Esempio

```
a:hover {  
  color: blue;  
}
```

IN QUESTO ESEMPIO, ABBIAMO UTILIZZATO LA PSEUDOCASSE **":HOVER"** PER APPLICARE IL COLORE BLU AI LINK HTML QUANDO IL CURSORE DEL MOUSE È SOPRA DI ESSI.





ID UNIVOCI





ID UNIVOCI

GLI **ID UNIVOCI** SONO IDENTIFICATORI CHE VENGONO UTILIZZATI PER IDENTIFICARE UN ELEMENTO HTML IN MODO UNIVOCO ALL'INTERNO DI UNA PAGINA WEB. A DIFFERENZA DELLE CLASSI CSS, GLI ID UNIVOCI POSSONO ESSERE UTILIZZATI SOLO SU UN ELEMENTO HTML SPECIFICO E NON POSSONO ESSERE RIUTILIZZATI SU ALTRI ELEMENTI HTML.





ID UNIVOCI

PER DEFINIRE UN ID UNIVOCO IN CSS, DOBBIAMO UTILIZZARE IL SELETTORE DI ID "**#**" SEGUITO DAL NOME DELL'ID. AD ESEMPIO:

IN QUESTO ESEMPIO, ABBIAMO DEFINITO UN ID UNIVOCO CHIAMATO "MY-ID" CHE APPLICA IL COLORE DI SFONDO GIALLO E IL COLORE DEL TESTO NERO A UN ELEMENTO HTML SPECIFICO CHE UTILIZZA QUESTO ID.

Esempio

```
#my-id {  
  background-color: yellow;  
  color: black;  
}
```



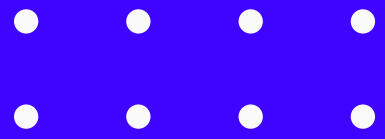


ID UNIVOCI

GLI ID UNIVOCI SONO UTILI QUANDO DOBBIAMO APPLICARE UNO STILE PERSONALIZZATO A UN ELEMENTO HTML SPECIFICO CHE NON UTILIZZIAMO ALTROVE NELLA PAGINA WEB.

TUTTAVIA, È IMPORTANTE UTILIZZARE GLI ID UNIVOCI CON PARSIMONIA, POICHÉ UN ID NON PUÒ ESSERE RIUTILIZZATO SU ALTRI ELEMENTI HTML E PUÒ PORTARE A UNA MAGGIOR COMPLESSITÀ DEL CODICE.





CASO PRATICO:

REALIZZAZIONE DI UNA BARRA DI NAVIGAZIONE COMPLESSA

OBIETTIVO: CREARE UNA BARRA DI NAVIGAZIONE COMPLESSA PER UN SITO WEB CHE CONTIENE DIVERSE VOCI DI MENU, UTILIZZANDO CSS PER LA FORMATTAZIONE E LA DISPOSIZIONE DEGLI ELEMENTI.





CASO PRATICO

IMMAGINA DI DOVER CREARE UNA BARRA DI NAVIGAZIONE COMPLESSA PER IL SITO WEB DI UN'AGENZIA DI VIAGGI. QUESTA BARRA DEVE ESSERE PERSONALIZZATA E INCLUDERE DIVERSE SEZIONI, TRA CUI:

- HOME
- DESTINAZIONI
- OFFERTE SPECIALI
- PRENOTAZIONI
- CONTATTI





CASO PRATICO

LA BARRA DI NAVIGAZIONE DEVE ESSERE POSIZIONATA NELLA PARTE SUPERIORE DEL SITO WEB. LA STRUTTURA HTML DELLA BARRA DI NAVIGAZIONE POTREBBE ESSERE LA SEGUENTE:



Esempio

```
<nav class="navbar">
  <ul class="navbar-list">
    <li class="navbar-item"><a href="#">Home</a></li>
    <li class="navbar-item"><a href="#">Destinazioni</a></li>
    <li class="navbar-item"><a href="#">Offerte speciali</a></li>
    <li class="navbar-item"><a href="#">Prenotazioni</a></li>
    <li class="navbar-item"><a href="#">Contatti</a></li>
  </ul>
</nav>
```





CASO PRATICO

ORA, UTILIZZANDO IL CSS, POSSIAMO PERSONALIZZARE LA BARRA DI NAVIGAZIONE AGGIUNGENDO UN BACKGROUND COLORATO, UNA TRANSIZIONE DI SFONDO QUANDO SI PASSA IL MOUSE SOPRA UN ELEMENTO DELLA LISTA E UNA FONT PERSONALIZZATA:





CASO PRATICO

```
Esempio

.navbar {
  background-color: #333;
  height: 50px;
}

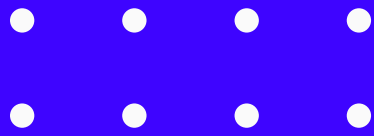
.navbar-list {
  margin: 0;
  padding: 0;
  list-style: none;
}

.navbar-item {
  float: left;
}

.navbar-item a {
  display: block;
  color: #fff;
  font-size: 18px;
  font-weight: bold;
  text-decoration: none;
  padding: 15px;
  font-family: 'Roboto', sans-serif;
  transition: background 0.3s ease-in-out;
}

.navbar-item a:hover {
  background-color: #555;
}
```





IMMAGINI IN BACKGROUND





IMMAGINI IN BACKGROUND

LE IMMAGINI IN BACKGROUND POSSONO ESSERE UTILIZZATE PER PERSONALIZZARE IL DESIGN DI UN SITO WEB, AGGIUNGENDO UN TOCCO VISIVO UNICO ALLA PAGINA. IN CSS, L'ATTRIBUTO **BACKGROUND-IMAGE** PUÒ ESSERE UTILIZZATO PER AGGIUNGERE UN'IMMAGINE DI SFONDO A UN ELEMENTO HTML.





IMMAGINI IN BACKGROUND: SINTASSI

L'ATTRIBUTO CSS BACKGROUND-IMAGE PUÒ ESSERE UTILIZZATO PER SPECIFICARE UN'IMMAGINE DI SFONDO PER UN ELEMENTO.

LA SINTASSI È LA SEGUENTE:

● ● ●

Esempio

```
background-image: url('percorso/immagine.jpg');
```

L'URL DEVE ESSERE IL PERCORSO DELL'IMMAGINE DI SFONDO, CHE PUÒ ESSERE UN'IMMAGINE LOCALE O UN'IMMAGINE ONLINE.





IMMAGINI IN BACKGROUND: PROPRIETÀ CSS

PER PERSONALIZZARE ULTERIORMENTE L'IMMAGINE DI SFONDO, È POSSIBILE UTILIZZARE ALTRE PROPRIETÀ CSS COME:

- **BACKGROUND-REPEAT**: SPECIFICA COME L'IMMAGINE DI SFONDO DEVE ESSERE RIPETUTA
- **BACKGROUND-SIZE**: SPECIFICA LE DIMENSIONI DELL'IMMAGINE DI SFONDO
- **BACKGROUND-POSITION**: SPECIFICA LA POSIZIONE DELL'IMMAGINE DI SFONDO ALL'INTERNO DELL'ELEMENTO
- **BACKGROUND-ATTACHMENT**: SPECIFICA SE L'IMMAGINE DI SFONDO DEVE SCORRERE CON LA PAGINA O RIMANERE FISSA





IMMAGINI IN BACKGROUND: ESEMPIO PRATICO

HTML

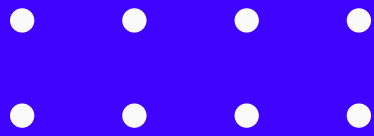
```
● ● ● Eempio
<section class="hero">
  <h1>Benvenuti sul mio sito web</h1>
</section>
```

CSS

```
● ● ● Eempio

.hero {
  background-image: url('sfondo.jpg');
  background-repeat: no-repeat;
  background-size: cover;
  background-position: center;
  height: 500px;
  text-align: center;
  color: #fff;
  display: flex;
  justify-content: center;
  align-items: center;
}
```





BOX MODEL





BOX MODEL: COMPONENTI

IL BOX MODEL CSS È UN CONCETTO FONDAMENTALE NELLA PROGETTAZIONE DI LAYOUT CON CSS.

È COMPOSTO DA QUATTRO PARTI PRINCIPALI:

- IL **CONTENUTO**: LA PARTE DEL BOX CHE CONTIENE IL CONTENUTO, AD ESEMPIO IL TESTO O LE IMMAGINI
- IL **BORDO**: LA PARTE DEL BOX CHE DELIMITA IL CONTENUTO E IL PADDING
- IL **PADDING**: LA PARTE DEL BOX CHE SEPARA IL CONTENUTO DAL BORDO
- IL **MARGINE**: LA PARTE DEL BOX CHE SEPARA IL BOX DA ALTRI ELEMENTI SULLA PAGINA





BOX MODEL: SINTASSI

IN CSS, POSSIAMO UTILIZZARE LE PROPRIETÀ PER DEFINIRE LE DIMENSIONI DI OGNI COMPONENTE DEL BOX MODEL. LA SINTASSI È LA SEGUENTE:

● ● ● Eempio

```
box-sizing: border-box;  
width: 100%;  
padding: 20px;  
border: 1px solid #ccc;  
margin: 10px;
```

NELL'ESEMPIO SOPRA, ABBIAMO UTILIZZATO LE PROPRIETÀ CSS WIDTH, PADDING, BORDER E MARGIN PER DEFINIRE LA DIMENSIONE DEL BOX E DEI SUOI COMPONENTI.





BOX MODEL: BOX SIZING

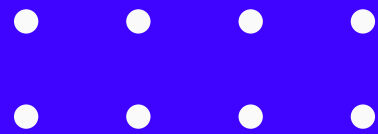
PER GESTIRE LA DIMENSIONE DEL BOX, ESISTE LA PROPRIETÀ CSS BOX-SIZING, CHE PUÒ ESSERE IMPOSTATA SU CONTENT-BOX O BORDER-BOX.

- **CONTENT-BOX**: LA DIMENSIONE DEL BOX INCLUDE SOLO IL CONTENUTO
- **BORDER-BOX**: LA DIMENSIONE DEL BOX INCLUDE IL CONTENUTO, IL PADDING E IL BORDO

```
Esempio

.box {
  box-sizing: border-box;
  width: 50%;
  padding: 20px;
  border: 1px solid #ccc;
  margin: 10px;
}
```





IL LAYOUT ELASTICO PER ADATTARE IL PROPRIO SITO A DIFFERENTI RISOLUZIONI E DIMENSIONI DEL MONITOR





LAYOUT ELASTICO

IL **LAYOUT ELASTICO** È UNA TECNICA DI DESIGN PER CREARE SITI WEB CHE SI ADATTANO A DIFFERENTI RISOLUZIONI E DIMENSIONI DEL MONITOR.





LAYOUT ELASTICO: IL CONCETTO DI FLUIDITÀ

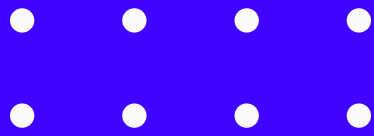
IL LAYOUT ELASTICO SI BASA SUL CONCETTO DI FLUIDITÀ. GLI ELEMENTI ALL'INTERNO DELLA PAGINA VENGONO DEFINITI CON UNITÀ DI MISURA *RELATIVE*, COME **PERCENTUALI**, **EM** O **REM**, INVECE DI UNITÀ DI MISURA FISSE, COME PIXEL.

IN QUESTO MODO, GLI ELEMENTI DELLA PAGINA SI ADATTANO IN MODO FLUIDO ALLE DIMENSIONI DEL MONITOR, SENZA DEFORMAZIONI O TAGLI.

EM: RELATIVO ALLA DIMENSIONE DEL CARATTERE DELL'ELEMENTO

REM: RELATIVO ALLA DIMENSIONE DEL CARATTERE DELL'ELEMENTO RADICE





BOX-SIZING





BOX-SIZING

LA PROPRIETÀ BOX-SIZING È UNA PROPRIETÀ DI CSS CHE PERMETTE DI MODIFICARE IL MODO IN CUI VIENE CALCOLATA LA DIMENSIONE DI UN ELEMENTO HTML, INCLUSI IL BORDO E IL PADDING.

VALORI DELLA PROPRIETÀ BOX-SIZING:

- **CONTENT-BOX** (VALORE PREDEFINITO): LA DIMENSIONE DELL'ELEMENTO HTML VIENE CALCOLATA INCLUDENDO SOLO IL CONTENUTO, ESCLUDENDO IL BORDO E IL PADDING.
- **BORDER-BOX**: LA DIMENSIONE DELL'ELEMENTO HTML VIENE CALCOLATA INCLUDENDO IL CONTENUTO, IL BORDO E IL PADDING.



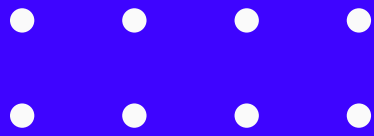


BOX-SIZING

LA PROPRIETÀ BOX-SIZING È PARTICOLARMENTE UTILE PER MANTENERE LA COERENZA DELLE DIMENSIONI DEGLI ELEMENTI HTML IN UN LAYOUT, GARANTENDO CHE LA LARGHEZZA E L'ALTEZZA SPECIFICATE SIANO SEMPRE RISPETTATE, INDIPENDENTEMENTE DALL'AGGIUNTA DI BORDI E PADDING.

TUTTAVIA, È IMPORTANTE CONSIDERARE LE IMPLICAZIONI DI QUESTA PROPRIETÀ QUANDO SI CREA IL DESIGN DI UN SITO WEB, POICHÉ POTREBBE INFLUIRE SULLA POSIZIONE DEGLI ELEMENTI E SULLA LORO DISPOSIZIONE ALL'INTERNO DELLA PAGINA.





CSS RESET





CSS RESET: COSA È?

IL CSS RESET È UNA TECNICA UTILIZZATA PER AZZERARE I VALORI DI DEFAULT DEI DIVERSI ELEMENTI HTML, IN MODO DA AVERE UN CONTROLLO MAGGIORE SULLE PROPRIETÀ CSS.

SPESSO I BROWSER HANNO STILI PREDEFINITI PER GLI ELEMENTI HTML, COME AD ESEMPIO MARGINI, PADDING E DIMENSIONI DEI CARATTERI, CHE POSSONO VARIARE DA UN BROWSER ALL'ALTRO E CHE POSSONO CREARE PROBLEMI DI COMPATIBILITÀ.

IL CSS RESET RISOLVE QUESTI PROBLEMI, CREANDO UNA SORTA DI "BASE NEUTRA" SU CUI POI APPLICARE I PROPRI STILI PERSONALIZZATI





CSS RESET: COME SI IMPLEMENTA

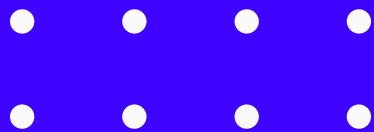
QUESTO È UN ESEMPIO DI CSS RESET MOLTO COMUNE, CHE IMPOSTA IL MARGIN E IL PADDING DI TUTTI GLI ELEMENTI A 0 E UTILIZZA IL BOX-SIZING "BORDER-BOX" PER CALCOLARE LE DIMENSIONI DEGLI ELEMENTI.

È IMPORTANTE TENERE PRESENTE CHE IL CSS RESET NON È L'UNICO MODO PER GESTIRE LA COMPATIBILITÀ TRA I DIVERSI BROWSER, E CHE PUÒ AVERE ANCHE ALCUNI EFFETTI COLLATERALI INDESIDERATI. È QUINDI CONSIGLIABILE UTILIZZARLO SOLO SE SI HA UNA BUONA CONOSCENZA DELLE IMPLICAZIONI CHE COMPORTA.

```
Esempio

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
```





@FONT-FACE





@FONT-FACE

LA PROPRIETÀ @FONT-FACE È UTILIZZATA IN CSS PER DEFINIRE FONT PERSONALIZZATI DA UTILIZZARE ALL'INTERNO DI UN SITO WEB. IN PRATICA, @FONT-FACE PERMETTE DI INCORPORARE UN FONT DIRETTAMENTE NEL SITO WEB, IN MODO CHE IL BROWSER POSSA VISUALIZZARLO CORRETTAMENTE ANCHE SE NON È INSTALLATO SUL COMPUTER DELL'UTENTE.





@FONT-FACE

IN QUESTO ESEMPIO, VIENE DEFINITO UN NUOVO FONT CHIAMATO "OPEN SANS", CHE VIENE CARICATO DAL FILE OPENSANS-REGULAR.TTF. IL FORMATO "TRUETYPE" SPECIFICATO CON LA PROPRIETÀ "FORMAT" INDICA IL TIPO DI FILE DEL FONT.



Esempio

```
@font-face {  
  font-family: 'Open Sans';  
  src: url('fonts/OpenSans-Regular.ttf') format('truetype');  
}
```





@FONT-FACE

UNA VOLTA DEFINITO IL FONT CON @FONT-FACE, È POSSIBILE UTILIZZARLO IN CSS APPLICANDO LA PROPRIETÀ "FONT-FAMILY" AL TESTO CHE SI DESIDERA FORMATTARE:



Esempio

```
body {  
  font-family: 'Open Sans', sans-serif;  
}
```



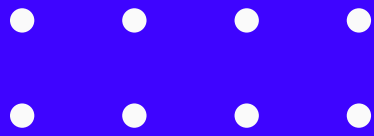


@FONT-FACE

IN QUESTO ESEMPIO, IL FONT "OPEN SANS" VIENE UTILIZZATO COME FONT PREDEFINITO PER TUTTO IL TESTO DEL SITO. LA PROPRIETÀ "SANS-SERIF" SPECIFICATA DOPO IL NOME DEL FONT INDICA UN FONT DI FALLBACK DA UTILIZZARE NEL CASO IN CUI "OPEN SANS" NON SIA DISPONIBILE.

È IMPORTANTE TENERE PRESENTE CHE L'UTILIZZO DI FONT PERSONALIZZATI TRAMITE @FONT-FACE PUÒ AUMENTARE IL TEMPO DI CARICAMENTO DEL SITO, E CHE È NECESSARIO ASSICURARSI DI AVERE I DIRITTI PER UTILIZZARE IL FONT SCELTO.





WOFF E WOFF2, SVG





WOFF E WOFF2, SVG

ESISTONO DIVERSI FORMATI PER I FONT UTILIZZATI NEI SITI WEB,
OGNUNO CON LE SUE CARATTERISTICHE E VANTAGGI.

DUE FORMATI COMUNI PER I FONT OPENTYPE SONO WOFF E WOFF2,
MENTRE UN FORMATO VETTORIALE SPESSO UTILIZZATO PER I LOGHI E LE
ICONE È SVG.





WOFF E WOFF2, SVG

I FORMATI WOFF E WOFF2 SONO PROGETTATI PER RIDURRE LA DIMENSIONE DEI FILE DEI FONT, MIGLIORANDO COSÌ IL TEMPO DI CARICAMENTO DEL SITO. WOFF (WEB OPEN FONT FORMAT) È STATO INTRODOTTO NEL 2009, MENTRE WOFF2 È UNA VERSIONE SUCCESSIVA, INTRODOTTA NEL 2013. QUESTI FORMATI SONO COMPATIBILI CON LA MAGGIOR PARTE DEI MODERNI BROWSER E CONSENTONO UNA MIGLIORE COMPRESSIONE RISPETTO AD ALTRI FORMATI, COME AD ESEMPIO TTF O OTF.



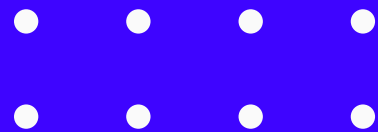


WOFF E WOFF2, SVG

IL FORMATO SVG (SCALABLE VECTOR GRAPHICS) È UN FORMATO VETTORIALE CHE PERMETTE DI CREARE GRAFICHE SCALABILI DI ALTA QUALITÀ.

IL FORMATO SVG È PARTICOLARMENTE UTILE PER LE ICONE E I LOGHI, PERCHÉ PUÒ ESSERE RIDIMENSIONATO SENZA PERDITA DI QUALITÀ E PERCHÉ È COMPATIBILE CON I DISPOSITIVI A SCHERMO AD ALTA RISOLUZIONE (COME I DISPLAY RETINA).





COLORI RGBA E LA PROPRIETÀ OPACITY





COLORI RGBA E LA PROPRIETÀ OPACITY

CSS OFFRE DIVERSE OPZIONI PER MODIFICARE I COLORI DI UN SITO WEB. UNA DI QUESTE OPZIONI È L'UTILIZZO DI COLORI RGBA E LA PROPRIETÀ OPACITY, CHE PERMETTONO DI CREARE SFUMATURE TRASPARENTI DI QUALSIASI COLORE.





COLORI RGBA E LA PROPRIETÀ OPACITY

IL VALORE RGBA (RED, GREEN, BLUE, ALPHA) È UN SISTEMA DI COLORI CHE UTILIZZA QUATTRO VALORI: IL VALORE ROSSO, IL VALORE VERDE E IL VALORE BLU (RGB) E IL VALORE ALPHA, CHE RAPPRESENTA LA TRASPARENZA DEL COLORE.



Esempio

```
background-color: rgba(255, 255, 255, 0.5);
```

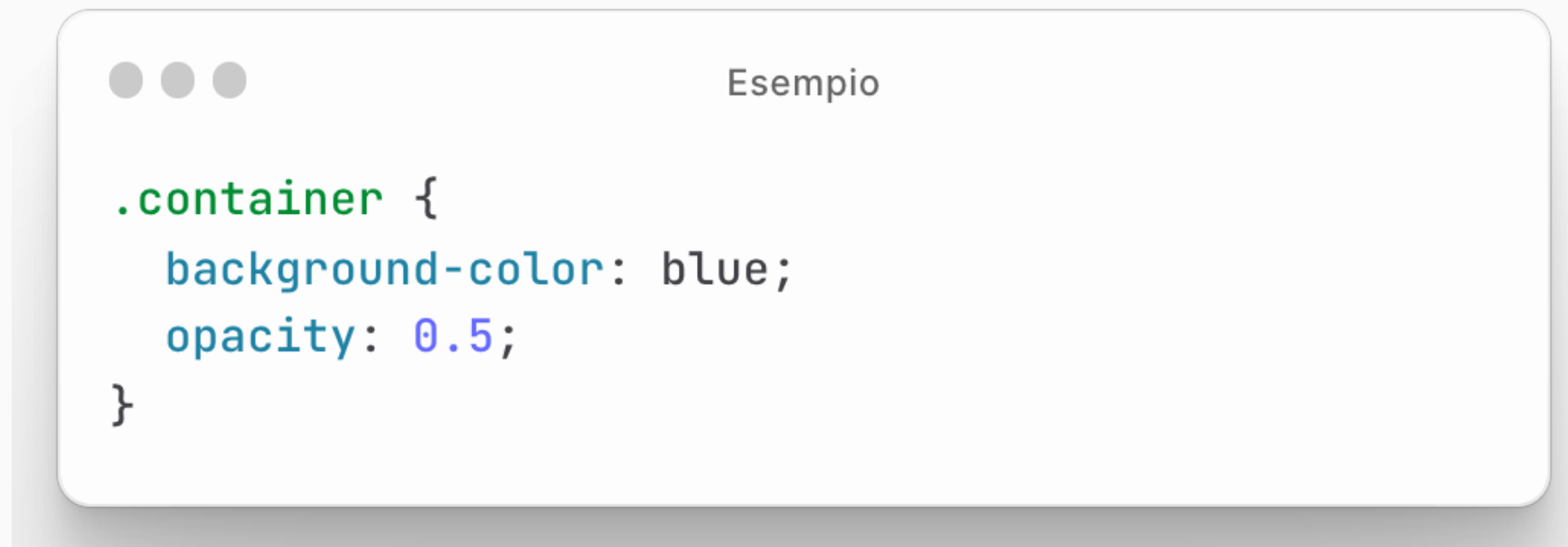
N QUESTO ESEMPIO, VIENE IMPOSTATO IL COLORE DI SFONDO A BIANCO, MA CON UN'OPACITÀ DEL 50% (0.5).





COLORI RGBA E LA PROPRIETÀ OPACITY

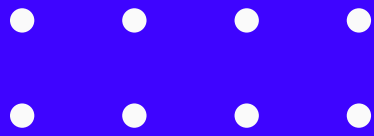
LA PROPRIETÀ OPACITY CONSENTE DI IMPOSTARE LA TRASPARENZA DI UN ELEMENTO HTML, COMPRESI TUTTI I CONTENUTI ALL'INTERNO DI QUELL'ELEMENTO.



```
.container {  
  background-color: blue;  
  opacity: 0.5;  
}
```

IN QUESTO ESEMPIO, VIENE IMPOSTATA L'OPACITÀ DEL CONTENITORE A 50%, RENDENDO IL COLORE DI SFONDO BLU TRASPARENTE.





ARROTONDARE I BORDI DEI BOX





ARROTONDARE I BORDI DEI BOX

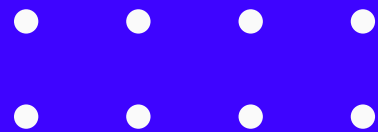
LA PROPRIETÀ **BORDER-RADIUS** È UNA PROPRIETÀ DI CSS CHE PERMETTE DI ARROTONDARE I BORDI DI UN ELEMENTO HTML, COME AD ESEMPIO UN DIV

VALORI DELLA PROPRIETÀ **BORDER-RADIUS**:

- **VALORE SINGOLO:** CONSENTE DI ARROTONDARE TUTTI I BORDI DELL'ELEMENTO ALLO STESSO MODO. AD ESEMPIO, **BORDER-RADIUS: 10PX** ARROTONDERÀ TUTTI I BORDI DI UN ELEMENTO DI 10PX.
- **VALORI SEPARATI:** CONSENTE DI ARROTONDARE OGNI BORDO DELL'ELEMENTO IN MODO DIVERSO. AD ESEMPIO, **BORDER-RADIUS: 10PX 20PX 30PX 40PX** ARROTONDERÀ I BORDI IN SENSO ORARIO A PARTIRE DALL'ANGOLO IN ALTO A SINISTRA CON 10PX, 20PX, 30PX E 40PX.
- **VALORI PERCENTUALI:** CONSENTE DI ARROTONDARE I BORDI IN BASE ALLA PERCENTUALE SPECIFICATA RISPETTO ALLE DIMENSIONI DELL'ELEMENTO. AD ESEMPIO, **BORDER-RADIUS: 50%** ARROTONDERÀ TUTTI I BORDI DELL'ELEMENTO AL 50% DELLE SUE DIMENSIONI.

.





LE PROPRIETÀ CSS3 PER OMBREGGIARE BOX E TESTI





BOX-SHADOW

LE PROPRIETÀ CSS3 **BOX-SHADOW** E **TEXT-SHADOW** PERMETTONO DI AGGIUNGERE OMBREGGIATURE A BOX E TESTI, RISPETTIVAMENTE, CREANDO UN EFFETTO TRIDIMENSIONALE SUL TUO SITO WEB.

LA PROPRIETÀ BOX-SHADOW CONSENTE DI AGGIUNGERE UN'OMBRA A UN ELEMENTO HTML, COME AD ESEMPIO UN DIV. LA SINTASSI DELLA PROPRIETÀ È LA SEGUENTE:



Esempio

```
box-shadow: h-shadow v-shadow blur spread color;
```





BOX-SHADOW

- **H-SHADOW:** VALORE OBBLIGATORIO PER LA LUNGHEZZA DELL'OMBRA SULL'ASSE ORIZZONTALE.
- **V-SHADOW:** VALORE OBBLIGATORIO PER LA LUNGHEZZA DELL'OMBRA SULL'ASSE VERTICALE.
- **BLUR:** VALORE FACOLTATIVO CHE SPECIFICA IL GRADO DI SFOCATURA DELL'OMBRA. SE NON SPECIFICATO, IL VALORE PREDEFINITO È 0.
- **SPREAD:** VALORE FACOLTATIVO CHE SPECIFICA LA DIMENSIONE DELL'OMBRA. SE NON SPECIFICATO, L'OMBRA SI ESTENDERÀ SU TUTTA LA SUPERFICIE DELL'ELEMENTO HTML.
- **COLOR:** VALORE FACOLTATIVO CHE SPECIFICA IL COLORE DELL'OMBRA. SE NON SPECIFICATO, IL VALORE PREDEFINITO È IL VALORE DI COLORE DEL TESTO.



Esempio

```
box-shadow: h-shadow v-shadow blur spread color;
```





TEXT-SHADOW

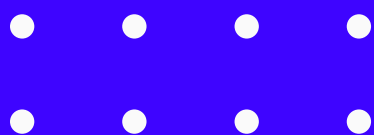
L'ORIZZONTALE E IL VERTICALE SPECIFICANO LA POSIZIONE DELL'OMBRA RISPETTO AL TESTO. LA SFUMATURA (BLUR) SPECIFICA LA SFOCATURA DELL'OMBRA, MENTRE IL COLORE (COLOR) SPECIFICA IL COLORE DELL'OMBRA.



Esempio

```
text-shadow: h-shadow v-shadow blur color;
```





PSEUDOCASSI





PSEUDOCCLASSI

CON L'INTRODUZIONE DEI CSS3, SONO STATE AGGIUNTE MOLTE NUOVE PSEUDOCCLASSI CHE CONSENTONO DI SELEZIONARE ELEMENTI IN BASE A CRITERI PIÙ SPECIFICI E SOFISTICATI.

ECCO ALCUNE DELLE PSEUDOCCLASSI PIÙ UTILI E INTERESSANTI INTRODOTTE DAI CSS3:

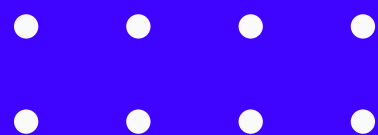




PSEUDOCCLASSI: ALCUNI ESEMPI

- **:NTH-CHILD()**: CONSENTE DI SELEZIONARE UN ELEMENTO FIGLIO IN BASE ALLA SUA POSIZIONE ALL'INTERNO DEL GENITORE. AD ESEMPIO, :NTH-CHILD(2N) SELEZIONA TUTTI GLI ELEMENTI FIGLIO PARI.
- **:NOT()**: QUESTA PSEUDOCCLASSE CONSENTE DI SELEZIONARE TUTTI GLI ELEMENTI CHE NON CORRISPONDONO A UN DETERMINATO SELETTORE. AD ESEMPIO, :NOT(P) SELEZIONA TUTTI GLI ELEMENTI CHE NON SONO PARAGRAFI.
- **:CHECKED**: QUESTA PSEUDOCCLASSE CONSENTE DI SELEZIONARE GLI ELEMENTI DI INPUT CHE SONO STATI SELEZIONATI O CONTROLLATI DALL'UTENTE. AD ESEMPIO, SE HAI UNA SERIE DI CASELLE DI CONTROLLO, PUOI UTILIZZARE :CHECKED PER SELEZIONARE SOLO QUELLE CHE SONO STATE SELEZIONATE.
- **:EMPTY**: QUESTA PSEUDOCCLASSE CONSENTE DI SELEZIONARE GLI ELEMENTI CHE NON CONTENGONO ALCUN CONTENUTO. AD ESEMPIO, :EMPTY SELEZIONA TUTTI GLI ELEMENTI VUOTI, COME I TAG <P> </P>.





TRE PROPRIETÀ SPECIALI: DISPLAY, FLOAT E CLEAR





TRE PROPRIETÀ SPECIALI: DISPLAY, FLOAT E CLEAR

DISPLAY, **FLOAT** E **CLEAR** SONO TRE PROPRIETÀ CSS CHE CONSENTONO DI CONTROLLARE LA POSIZIONE E IL LAYOUT DEGLI ELEMENTI HTML ALL'INTERNO DI UNA PAGINA WEB.





TRE PROPRIETÀ SPECIALI: LA PROPRIETÀ DISPLAY

LA PROPRIETÀ **DISPLAY** SPECIFICA COME UN ELEMENTO HTML DEVE ESSERE VISUALIZZATO SULLO SCHERMO.

ESISTONO DIVERSE OPZIONI TRA CUI SCEGLIERE:

- **BLOCK**: L'ELEMENTO VIENE VISUALIZZATO COME UN BLOCCO E OCCUPA L'INTERA LARGHEZZA DISPONIBILE
- **INLINE**: L'ELEMENTO VIENE VISUALIZZATO ALL'INTERNO DELLA LINEA DEL TESTO E NON OCCUPA L'INTERA LARGHEZZA DISPONIBILE
- **INLINE-BLOCK**: L'ELEMENTO VIENE VISUALIZZATO COME UN BLOCCO ALL'INTERNO DELLA LINEA DEL TESTO
- **NONE**: L'ELEMENTO NON VIENE VISUALIZZATO SULLO SCHERMO
- **FLEX**: L'ELEMENTO VIENE VISUALIZZATO COME UN FLESSIBILE CONTAINER DI CONTENUTO





FOCUS SU FLEXBOX

FLEXBOX È UN MODULO DI LAYOUT CSS CHE CONSENTE DI POSIZIONARE E ALLINEARE GLI ELEMENTI IN UN CONTENITORE IN MODO FLESSIBILE, INTUITIVO E REATTIVO. CON FLEXBOX, È POSSIBILE DEFINIRE IN MODO PRECISO COME GLI ELEMENTI DEVONO ESSERE POSIZIONATI ALL'INTERNO DI UN CONTENITORE, UTILIZZANDO PROPRIETÀ COME "DISPLAY", "FLEX-DIRECTION", "JUSTIFY-CONTENT" E "ALIGN-ITEMS".

IL CONCETTO CHIAVE DI FLEXBOX È IL "CONTENITORE FLESSIBILE" (FLEX CONTAINER) CHE CONTIENE UNO O PIÙ ELEMENTI CHE VENGONO CHIAMATI "ELEMENTI FLESSIBILI" (FLEX ITEMS). IL CONTENITORE FLESSIBILE PUÒ ESSERE CONFIGURATO CON DIVERSE PROPRIETÀ, CHE CONTROLLANO COME GLI ELEMENTI FLESSIBILI VENGONO DISPOSTI ALL'INTERNO DEL CONTENITORE.

AD ESEMPIO, LA PROPRIETÀ "FLEX-DIRECTION" DEFINISCE LA DIREZIONE IN CUI GLI ELEMENTI FLESSIBILI VENGONO DISPOSTI ALL'INTERNO DEL CONTENITORE (DA SINISTRA A DESTRA, DA DESTRA A SINISTRA, DALL'ALTO VERSO IL BASSO O DAL BASSO VERSO L'ALTO). LA PROPRIETÀ "JUSTIFY-CONTENT" DEFINISCE COME GLI ELEMENTI FLESSIBILI VENGONO ALLINEATI LUNGO L'ASSE PRINCIPALE (AD ESEMPIO, ALLINEATI A SINISTRA, A DESTRA, CENTRALIZZATI O DISTRIBUITI EQUAMENTE). LA PROPRIETÀ "ALIGN-ITEMS" DEFINISCE COME GLI ELEMENTI FLESSIBILI VENGONO ALLINEATI LUNGO L'ASSE TRASVERSALE (AD ESEMPIO, ALLINEATI IN ALTO, IN BASSO O CENTRALIZZATI).





TRE PROPRIETÀ SPECIALI: LA PROPRIETÀ FLOAT

ESPLOREREMO UN CONCETTO CHIAVE IN CSS CHIAMATO "**float**", CHE CI CONSENTE DI ORGANIZZARE GLI ELEMENTI NELLA NOSTRA PAGINA WEB IN MODO PARTICOLARE.

PENSATE AL FLOAT COME A UN MODO DI FAR "GALLEGGIARE" GLI ELEMENTI ALL'INTERNO DI UN CONTENITORE, DETERMINANDO SE DEVONO ALLINEARSI A SINISTRA O A DESTRA.

- **left**: L'ELEMENTO VIENE SPOSTATO A SINISTRA
- **right**: L'ELEMENTO VIENE SPOSTATO A DESTRA
- **none**: L'ELEMENTO NON VIENE SPOSTATO





TRE PROPRIETÀ SPECIALI: LA PROPRIETÀ FLOAT

DEFINIZIONE DEL FLOAT:

QUANDO APPLICHIAMO IL FLOAT A UN ELEMENTO, STIAMO DICENDO AL BROWSER DI FARLO "GALLEGGIARE" RISPETTO AGLI ALTRI ELEMENTI NEL SUO CONTESTO.

```
.ELEMENTO {  
  FLOAT: LEFT; /* O RIGHT */  
}
```





TRE PROPRIETÀ SPECIALI: LA PROPRIETÀ FLOAT

IMPATTO SUL LAYOUT:

GLI ELEMENTI FLOAT ESCONO DAL FLUSSO DI LAYOUT NORMALE E CONSENTONO AGLI ALTRI ELEMENTI DI "AVVOLGERLI".

QUESTO È UTILE QUANDO VOGLIAMO ORGANIZZARE GLI ELEMENTI IN ORIZZONTALE, COME AD ESEMPIO UNA FILA DI IMMAGINI O BLOCCHI DI TESTO.





TRE PROPRIETÀ SPECIALI: LA PROPRIETÀ FLOAT

ATTENZIONE AL CONTENITORE:

SE ABBIAMO ELEMENTI FLOAT ALL'INTERNO DI UN CONTENITORE, POTREMMO NOTARE CHE IL CONTENITORE NON SI ADATTA PIÙ AUTOMATICAMENTE ALL'ALTEZZA DEGLI ELEMENTI FLOAT. PER RISOLVERE QUESTO PROBLEMA, DOBBIAMO SPESSO UTILIZZARE LA TECNICA DEL "CLEARFIX" O ALTRI METODI DI PULIZIA DEL FLOAT.



```
.contenitore::after {  
  content: "";  
  display: table;  
  clear: both;  
}
```





TRE PROPRIETÀ SPECIALI: LA PROPRIETÀ CLEAR

PARLIAMO DEL CONCETTO DI **CLEAR** IN CSS, CHE SPESSO È CORRELATO ALL'UTILIZZO DEL FLOAT.

QUANDO UTILIZZIAMO IL FLOAT PER FAR GALLEGGIARE GLI ELEMENTI ALL'INTERNO DI UN CONTENITORE, PUÒ VERIFICARSI UN EFFETTO COLLATERALE INTERESSANTE.

GLI ELEMENTI SUCCESSIVI POTREBBERO NON "RENDERSI CONTO" DELLA PRESENZA DEGLI ELEMENTI FLOAT E FINIRE PER SOVRAPPORSI AD ESSI.

È QUI CHE ENTRA IN GIOCO LA PROPRIETÀ CLEAR.





TRE PROPRIETÀ SPECIALI: LA PROPRIETÀ CLEAR

LA PROPRIETÀ CLEAR VIENE UTILIZZATA PER SPECIFICARE DA QUALE LATO DEGLI ELEMENTI FLOAT SI DESIDERA CHE GLI ELEMENTI SUCCESSIVI NON SIANO INFLUENZATI.

```
.elemento-successivo {  
  clear: left; /* o right o both o none */  
}
```

- LEFT: L'ELEMENTO SUCCESSIVO NON DEVE TOCCARE GLI ELEMENTI FLOAT A SINISTRA.
- RIGHT: L'ELEMENTO SUCCESSIVO NON DEVE TOCCARE GLI ELEMENTI FLOAT A DESTRA.
- BOTH: L'ELEMENTO SUCCESSIVO NON DEVE TOCCARE GLI ELEMENTI FLOAT SU ENTRAMBI I LATI.
- NONE: L'ELEMENTO SUCCESSIVO PUÒ TOCCARE GLI ELEMENTI FLOAT SU ENTRAMBI I LATI.





TRE PROPRIETÀ SPECIALI: LA PROPRIETÀ CLEAR

ESEMPIO PRATICO:

SUPPONIAMO DI AVERE UN CONTENITORE CON ELEMENTI FLOAT A SINISTRA:

```
● ● ●  
  
.float_a_sinistra {  
  float: left;  
}
```





TRE PROPRIETÀ SPECIALI: LA PROPRIETÀ CLEAR

E VOGLIAMO CHE UN ELEMENTO SUCCESSIVO NON VENGA INFLUENZATO DAGLI ELEMENTI FLOAT:

```
...  
  
.elemento-successivo {  
  clear: left;  
}
```

IN QUESTO MODO, L'ELEMENTO-SUCCESSIVO VERRÀ VISUALIZZATO SOTTO GLI ELEMENTI FLOAT A SINISTRA, EVITANDO SOVRAPPOSIZIONI INDESIDERATE.





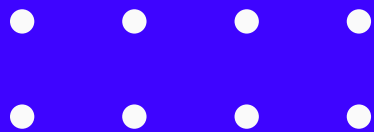
TRE PROPRIETÀ SPECIALI: LA PROPRIETÀ CLEAR

QUANDO UTILIZZARLO????

LA PROPRIETÀ CLEAR È PARTICOLARMENTE UTILE QUANDO SI LAVORA CON ELEMENTI FLOAT E SI DESIDERA GARANTIRE CHE GLI ELEMENTI SUCCESSIVI SIANO POSIZIONATI CORRETTAMENTE RISPETTO A ESSI.

IN CONCLUSIONE, IL CLEAR È UN MODO PER GESTIRE COME GLI ELEMENTI SUCCESSIVI INTERAGISCONO CON GLI ELEMENTI FLOAT SUL LAYOUT DELLA PAGINA WEB.





CSS GRID





CSS GRID

CSS GRID È UNA POTENTE FUNZIONALITÀ DI CSS CHE CI CONSENTE DI CREARE LAYOUT COMPLESSI E FLESSIBILI IN MODO DICHIARATIVO.

INVECE DI AFFIDARSI A FLOAT O ALTRI TRUCCHI, CSS GRID CI OFFRE UN APPROCCIO PIÙ DIRETTO ALLA CREAZIONE DI LAYOUT.





CSS GRID

DEFINIZIONE DELLA GRIGLIA:

PER INIZIARE, DOBBIAMO DEFINIRE UNA GRIGLIA NEL CONTENITORE PADRE.
UTILIZZIAMO LA PROPRIETÀ `DISPLAY: GRID;`.

```
.contenitore {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-gap: 10px;  
}
```

QUESTO ESEMPIO CREA UNA GRIGLIA CON TRE COLONNE DI LARGHEZZA
UGUALE E UNO SPAZIO DI 10PX TRA LE COLONNE.





CSS GRID: GRID-TEMPLATE-COLUMNS

LA PROPRIETÀ GRID-TEMPLATE-COLUMNS DEFINISCE IL NUMERO, LA LARGHEZZA E LA DISPOSIZIONE DELLE COLONNE ALL'INTERNO DELLA GRIGLIA. POSSIAMO SPECIFICARE LE COLONNE IN VARI MODI, AD ESEMPIO USANDO VALORI ASSOLUTI, FRAZIONI, O PAROLE CHIAVE.



Sintassi

```
.contenitore {  
  grid-template-columns: valore_colonna1 valore_colonna2 ...;  
}
```





CSS GRID: GRID-TEMPLATE-COLUMNS

ESEMPIO CON VALORI ASSOLUTI



```
.contenitore {  
  grid-template-columns: 100px 200px 150px;  
}
```

QUESTO CREA TRE COLONNE, RISPETTIVAMENTE DI 100PX, 200PX E 150PX.





CSS GRID: GRID-TEMPLATE-COLUMNS

UTILIZZO DI FRAZIONI



```
.contenitore {  
  grid-template-columns: 1fr 2fr 1fr;  
}
```

QUESTO ASSEGNA IL 1/4 DELLO SPAZIO ALLA PRIMA COLONNA, LA METÀ ALLO SPAZIO ALLA SECONDA COLONNA E UN QUARTO ALLA TERZA COLONNA.





CSS GRID: GRID-TEMPLATE-COLUMNS

UTILIZZO DI PAROLE CHIAVE



```
.contenitore {  
  grid-template-columns: repeat(3, 1fr);  
}
```

LA FUNZIONE REPEAT SEMPLIFICA LA DEFINIZIONE DI COLONNE RIPETITIVE.
IN QUESTO CASO, CREA TRE COLONNE, OGNUNA CON 1FR DI LARGHEZZA.





CSS GRID

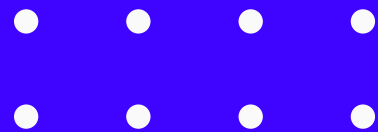
POSIZIONAMENTO DEGLI ELEMENTI:

GLI ELEMENTI FIGLIO POSSONO ESSERE POSIZIONATI NELLA GRIGLIA UTILIZZANDO LE PROPRIETÀ GRID-ROW E GRID-COLUMN.



```
.elemento {  
  grid-row: span 2; /* Occupa due righe */  
  grid-column: 2 / 4; /* Inizia dalla seconda colonna e termina alla quarta colonna */  
}
```





LE MEDIA QUERIES CSS3





MEDIA QUERIES

LE **MEDIA QUERIES** SONO UNO STRUMENTO POTENTE PER CREARE SITI WEB RESPONSIVI, CHE SI ADATTANO AUTOMATICAMENTE ALLE DIVERSE DIMENSIONI E RISOLUZIONI DEI DISPOSITIVI UTILIZZATI PER LA NAVIGAZIONE. LE MEDIA QUERIES CONSENTONO DI IMPOSTARE BREAKPOINT SPECIFICI IN BASE ALLE DIMENSIONI DELLO SCHERMO O AD ALTRE CARATTERISTICHE DEL DISPOSITIVO, COME LA RISOLUZIONE, L'ORIENTAMENTO, LA DENSITÀ DI PIXEL E ALTRO ANCORA.





MEDIA QUERIES

PER IMPOSTARE I BREAKPOINT IN UN FOGLIO DI STILE UTILIZZANDO LE MEDIA QUERIES, PUOI UTILIZZARE LA SEGUENTE SINTASSI:



Esempio

```
@media (breakpoint) {  
    /* stili per il breakpoint */  
}
```





MEDIA QUERIES

PUOI SOSTITUIRE "BREAKPOINT" CON IL VALORE CORRISPONDENTE ALLE DIMENSIONI DELLO SCHERMO O AD ALTRE CARATTERISTICHE DEL DISPOSITIVO, COME AD ESEMPIO:

- **WIDTH**: LA LARGHEZZA DELLO SCHERMO IN PIXEL (AD ESEMPIO, @MEDIA (MIN-WIDTH: 768PX))
- **HEIGHT**: L'ALTEZZA DELLO SCHERMO IN PIXEL (AD ESEMPIO, @MEDIA (MIN-HEIGHT: 500PX))
- **ORIENTATION**: L'ORIENTAMENTO DELLO SCHERMO (AD ESEMPIO, @MEDIA (ORIENTATION: PORTRAIT))
- **RESOLUTION**: LA RISOLUZIONE DELLO SCHERMO IN DPI (AD ESEMPIO, @MEDIA (MIN-RESOLUTION: 300DPI))
- **ASPECT-RATIO**: IL RAPPORTO TRA LARGHEZZA E ALTEZZA DELLO SCHERMO (AD ESEMPIO, @MEDIA (ASPECT-RATIO: 16/9))

INOLTRE, PUOI COMBINARE PIÙ VALORI UTILIZZANDO GLI OPERATORI LOGICI COME AND E OR.

