

CSS: template delle griglie con valori nominali

CSS Grid ci permette di definire la posizione degli elementi anche usando riferimenti nominali.

Supponiamo di avere un layout a due colonne con header e footer. La struttura HTML è la seguente:

```
<section id="site">
  <header id="site-header"></header>
  <section id="content"></section>
  <aside id="navigation"></aside>
  <footer id="site-info"></footer>
</section>
```

Definiamo subito gli stili di base del contenitore:

```
#site {
  display: grid;
  grid-template-columns: 1fr 1fr;
  grid-template-rows: 1fr 1fr 1fr 1fr;
  grid-template-areas:
    "header header"
    "navigation navigation"
    "content content"
    "site-info site-info"
  ;
```

```
}  
  
@media screen and (min-width: 768px) {  
  #site {  
    grid-template-areas:  
      "header header"  
      "content navigation"  
      "site-info site-info"  
    ;  
    max-width: 960px;  
    margin: 0 auto;  
  }  
}
```

`grid-template-columns` stabilisce che questa griglia avrà due colonne. `1fr` usa la nuova unità di misura *fraction* che da sola esegue i calcoli sulla ripartizione dello spazio.

`grid-template-rows` stabilisce che questa griglia avrà quattro righe. `grid-template-areas` definisce la disposizione degli elementi sulla griglia usando i riferimenti nominali definiti su ciascun elemento con la proprietà `grid-area`.

Tenendo a mente che abbiamo due colonne, `"header header"` vuol dire che l'header occuperà entrambe le colonne in orizzontale, mentre `"content navigation"`, significa che il contenuto andrà sulla colonna di sinistra e la navigazione su quella di destra. L'ordine in cui appaiono questi valori è rilevante. Ecco come vengono definiti sugli elementi.

```
#site-header {
```

```
    grid-area: header;
    background-color: green;
}

#content {
    grid-area: content;
    background-color: silver;
}

#navigation {
    grid-area: navigation;
    background-color: yellow;
}

#site-info {
    grid-area: site-info;
    background-color: orange;
}
```

 Griglia CSS a due colonne con header, footer, contenuto e navigazione

Come si può notare, questa caratteristica rende estremamente facile la costruzione di layout con i CSS.