

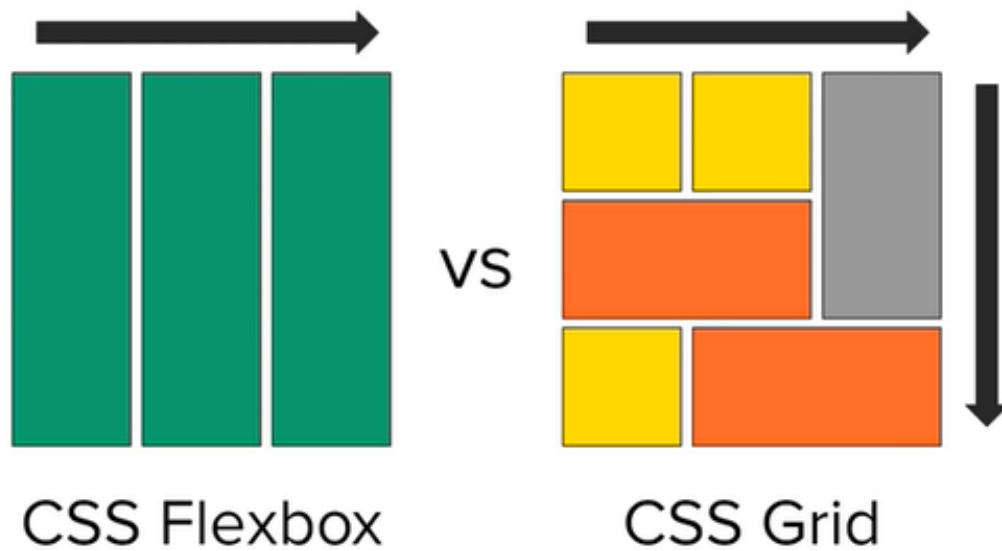
GRID



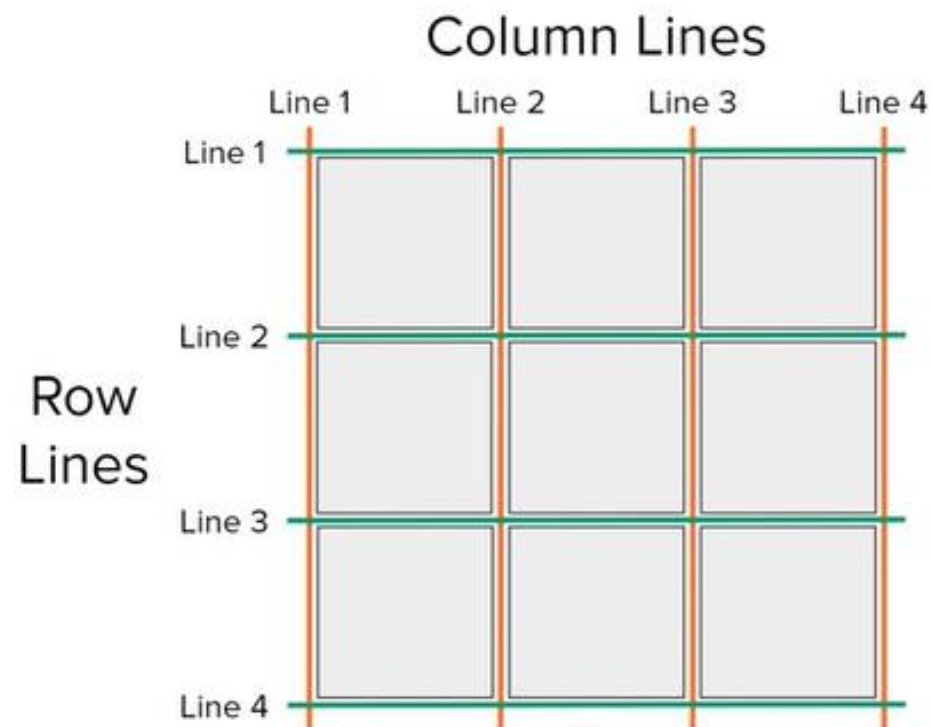
CSS Grid Layout to układu oparty na siatce, z wierszami i kolumnami, co ułatwia projektowanie stron internetowych..

CSS Grid Layout jest najlepszą metodą aby podzielić stronę na części, definiowanie relacji takich jak rozmiar, pozycja i warstwa, pomiędzy podstawowymi elementami HTML.

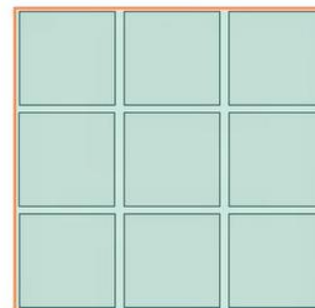
Podobnie do tabela rozmieszczenie grid pozwala układać elementy w kolumny i wiersze. Jednak grid dopuszcza więcej ułożeń niż tabela. Na przykład kontener grid potrafi spozycjonować elementy wewnątrz siebie w taki sposób, aby na siebie nachodziły oraz używa warstw.



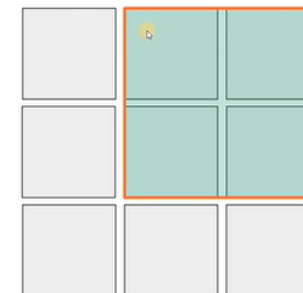
Podstawowe pojęcia



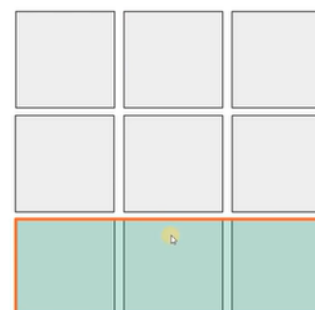
Container



Area



Track



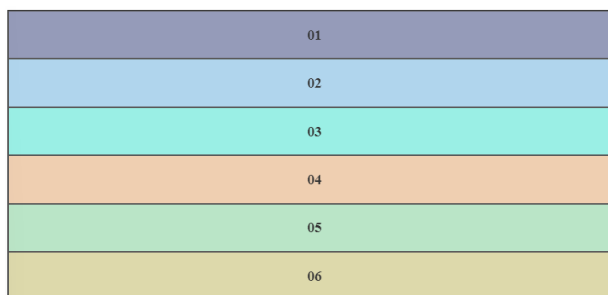
CSS Grid Layout

Kod na którym będą bazować
poniższe przykłady



```
<section class="grid_container">
  <article class="item_1">
    <h3>01</h3>
  </article>
  <article class="item_2">
    <h3>02</h3>
  </article>
  <article class="item_3">
    <h3>03</h3>
  </article>
  <article class="item_4">
    <h3>04</h3>
  </article>
  <article class="item_5">
    <h3>05</h3>
  </article>
  <article class="item_6">
    <h3>06</h3>
  </article>
</section>
```

```
section {
  margin: 10px;
  border: 1px solid rgb(47, 47, 47);
}
article {
  display: flex;
  align-items: center;
  justify-content: center;
  border: 1px solid rgb(47, 47, 47);
}
article {
  filter: opacity(80%);
}
```



```
.grid_container {
}
.item_1 {
  background-color: #7a82a7;
}
.item_2 {
  background-color: #9ccae9;
}
.item_3 {
  background-color: #81ebde;
}
.item_4 {
  background-color: #ebc39d;
}
.item_5 {
  background-color: #a9dfb9;
}
.item_6 {
  background-color: #d5cf96;
}
```

Definiowanie wierszy i kolumn

```
.grid_container {  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
}
```

01	02
03	04
05	06

Definiowanie wierszy i kolumn

```
.grid_container {  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
}
```

02	00
03	04
01	05

```
.grid_container {  
  display: grid;  
  grid-template-columns: 50px 1fr;  
}
```

02	00
03	04
01	05

```
.grid_container {  
  display: grid;  
  grid-template-columns:  
  1fr 3fr .5fr;  
}
```

04	02	00
01	05	03

Definiowanie wierszy i kolumn

```
.grid_container {  
  display: grid;  
  grid-template-columns: repeat(4,1fr);  
}
```

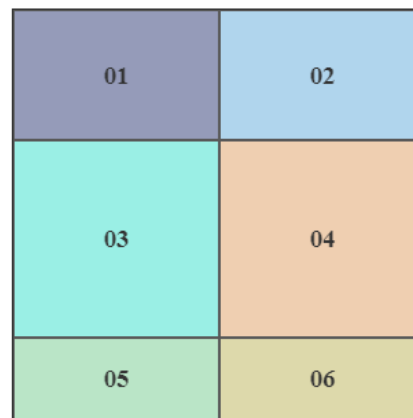
01	02	03	04
05	06		

```
.grid_container {  
  display: grid;  
  grid-template-columns: repeat(4,  
auto);  
}
```

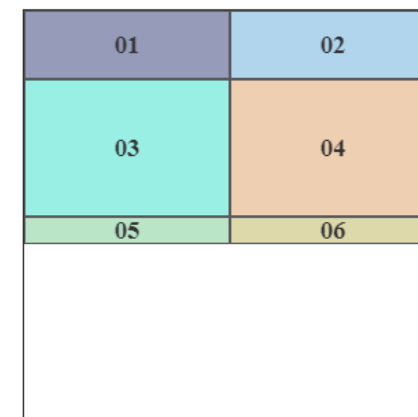
01	02	03	04
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum convallis, quam nec malesuada rhoncus, odio metus vestibulum quam, in laoreet ex odio scelerisque mi. Sed sit amet egestas augue. Proin fermentum sit amet nibh eget facilisis. Suspendisse ultrices et lacus at placerat..	06		

Definiowanie wierszy i kolumn

```
.grid_container {  
  display: grid;  
  height: 300px;  
  width: 300px;  
  grid-template-columns: repeat(2,1fr);  
  grid-template-rows: 2fr 3fr 1fr;  
}
```



```
.grid_container {  
  display: grid;  
  height: 300px;  
  width: 300px;  
  grid-template-columns: repeat(2,1fr);  
  grid-template-rows: 50px 100px 20px;  
}
```

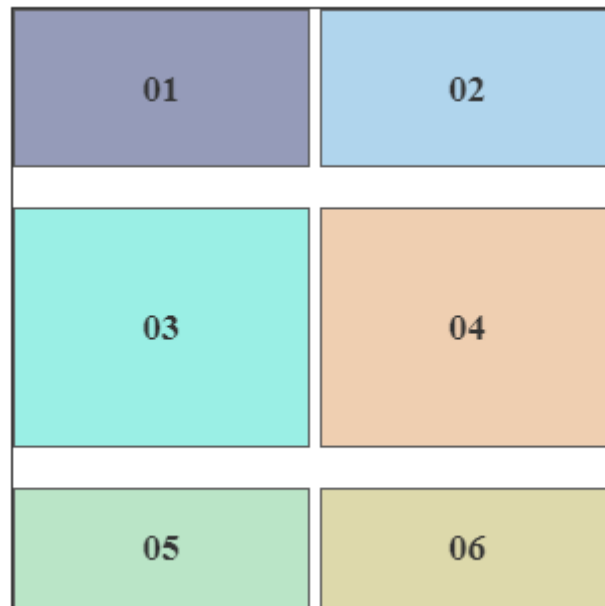


Odstępy pomiędzy kolumnami i wierszami

```
gap: 20px 5px;
```

```
column-gap: 5px;
```

```
row-gap: 20px;
```

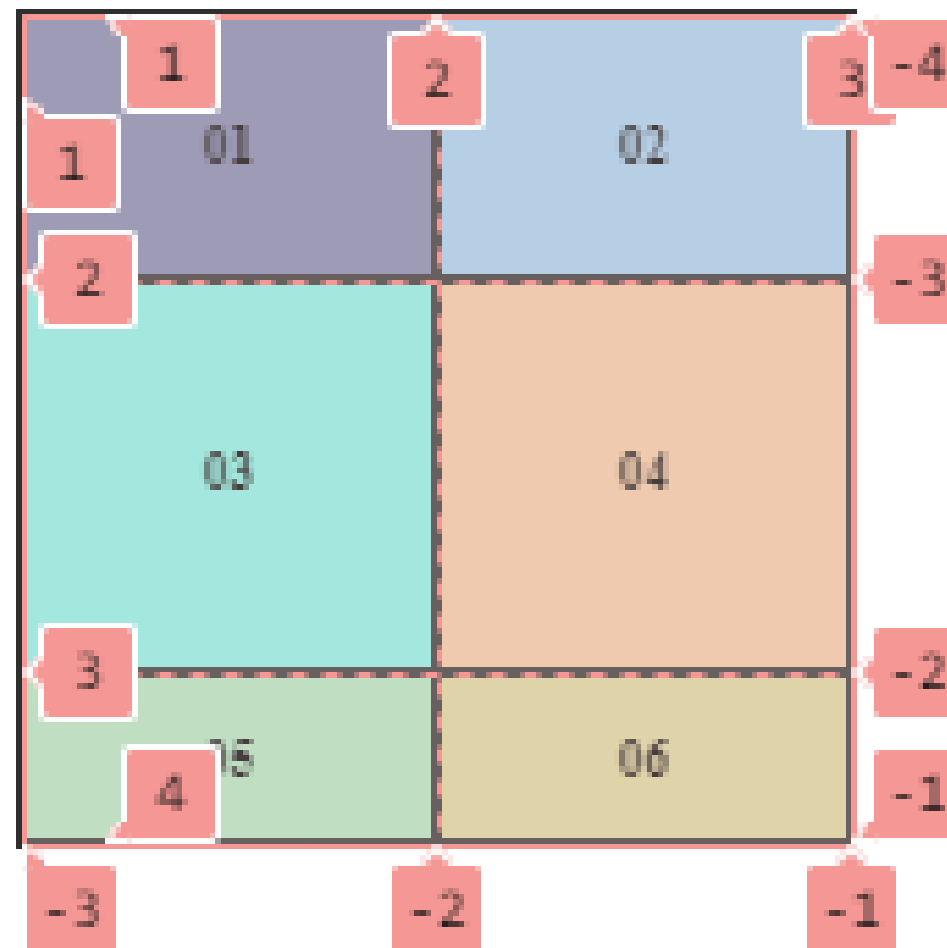


Zwróćmy uwagę, że numeracja linii kolumn i linii wierszy jest dwustronna:

- - od lewej do prawej liczby dodatnie (od jedynki)
- od prawej do lewej liczby ujemne (od -1)

Gdzie -1 oznacza ostatnią linię -2 przedostatnią i tak dalej.

Uwaga! Rozmieszczając elementy odnosimy się nie do kolumn i wierszy lecz bezpośrednio do linii.



Rozmieszczanie elementów sposób pierwszy: względem linii kolumn i linii wierszy

```
<section class="grid_container">
  <article class="item_1">
    <h3>01</h3>
  </article>
  <article class="item_2">
    <h3>02</h3>
  </article>
  <article class="item_3">
    <h3>03</h3>
  </article>
  <article class="item_4">
    <h3>04</h3>
  </article>
  <article class="item_5">
    <h3>05</h3>
  </article>
  <article class="item_6">
    <h3>06</h3>
  </article>
</section>
```

```
section {
  margin: 10px;
  border: 1px solid rgb(47, 47, 47);
}

article {
  display: flex;
  align-items: center;
  justify-content: center;
  border: 1px solid rgb(47, 47, 47);
}

article {
  filter: opacity(50%);
}
```

```
.grid_container {
  display: grid;
  height: 300px;
  width: 300px;
  grid-template-columns: repeat(4,1fr);
  grid-template-rows: repeat( 4, 1fr);
  gap: 10px;
}
```



```
.item_1 {  
  background-color: #7a82a7;  
  grid-column-start: 1;  
  grid-column-end: 3;  
  grid-row-start: 2;  
  grid-row-end: 4;  
}
```

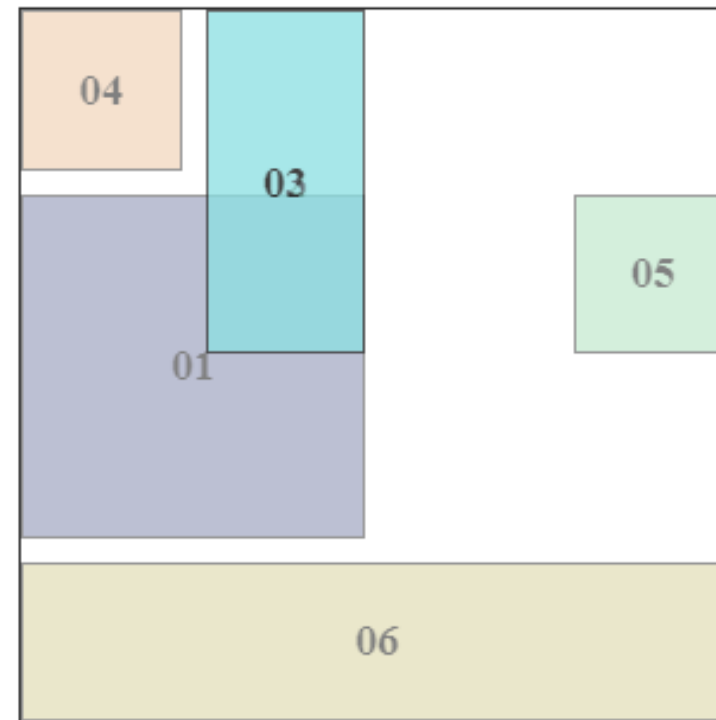
```
.item_2 {  
  background-color: #9ccae9;  
  grid-column-start: 2;  
  grid-column-end: 3;  
  grid-row-start: 1;  
  grid-row-end: 3;  
}
```

```
.item_3 {  
  background-color: #81ebde;  
  grid-column-start: 2;  
  grid-column-end: 3;  
  grid-row-start: 1;  
  grid-row-end: 3;  
}
```

```
.item_4 {  
  background-color: #ebc39d;  
}
```

```
.item_5 {  
  background-color: #a9dfb9;  
  grid-column: -2 / -2;  
  grid-row: 2 / 2;  
}
```

```
.item_6 {  
  background-color: #d5cf96;  
  grid-column: 1 / -1;  
  grid-row: -1 / -2;  
}
```





grid-template-areas Właściwość ta określa nazwane obszary siatki, ustanawiając komórki w siatce.

```
grid-template-areas: "a b";
```

```
grid-template-areas: "a b ."  
                    "a c c";
```

CSS Grid Layout



Przykład:



```
<section id="page">
  <header>Header</header>
  <nav>Navigation</nav>
  <main>Main area</main>
  <footer>Footer</footer>
</section>
```

```
#page {
  display: grid;
  width: 100%;
  height: 250px;
  grid-template-areas:
    "head head"
    "nav main"
    "nav foot";
  grid-template-rows: 50px 1fr 30px;
  grid-template-columns: 150px 1fr;
}

header, nav, main, footer {
  display: flex;
  align-items: center;
  justify-content: center;
}
```

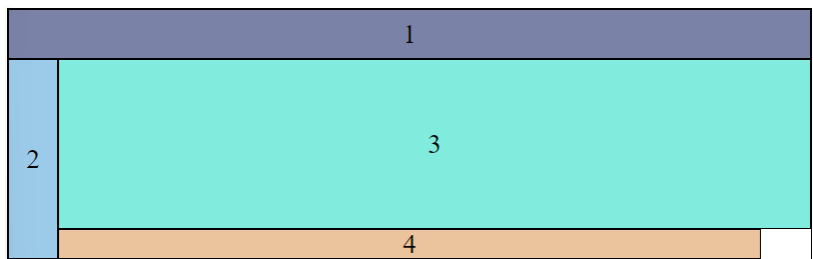
```
header {
  grid-area: head;
  background-color: #7a82a7;
}

nav {
  grid-area: nav;
  background-color: #9ccae9;
}

main {
  grid-area: main;
  background-color: #81ebde;
}

footer {
  grid-area: foot;
  background-color: #ebc39d;
}
```

II wersja:



```
<section class="contener">
  <div class="item_1">1</div>
  <div class="item_2">2</div>
  <div class="item_3">3</div>
  <div class="item_4">4</div>
</section>
```

```
.container {
  display: grid;
  width: 100%;
  height: 250px;
  border: 1px solid black;
  grid-template-areas:
    "head head head"
    "nav  main main"
    "nav  foot .";
  grid-template-rows: 50px 1fr 30px;
  grid-template-columns: 50px 1fr 50px;
}

div {
  display: flex;
  align-items: center;
  justify-content: center;
  font-size: 20pt;
  border: 1px solid black;
}
```

```
.item_1 {
  grid-area: head;
  background-color: #7a82a7;
}

.item_2 {
  grid-area: nav;
  background-color: #9ccae9;
}

.item_3 {
  grid-area: main;
  background-color: #81ebde;
}

.item_4 {
  grid-area: foot;
  background-color: #ebc39d;
}
```




Inne ciekawe opcje:

auto-fit
auto-fill

```
grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
```

```
grid-auto-flow: column;
```

```
align-items: flex-end;
```

```
justify-self: flex-start;
```

```
justify-items: flex-end;
```

```
justify-content: space-around;
```