



## CSS **Grid**

[dimon.work/kurs.html](http://dimon.work/kurs.html)

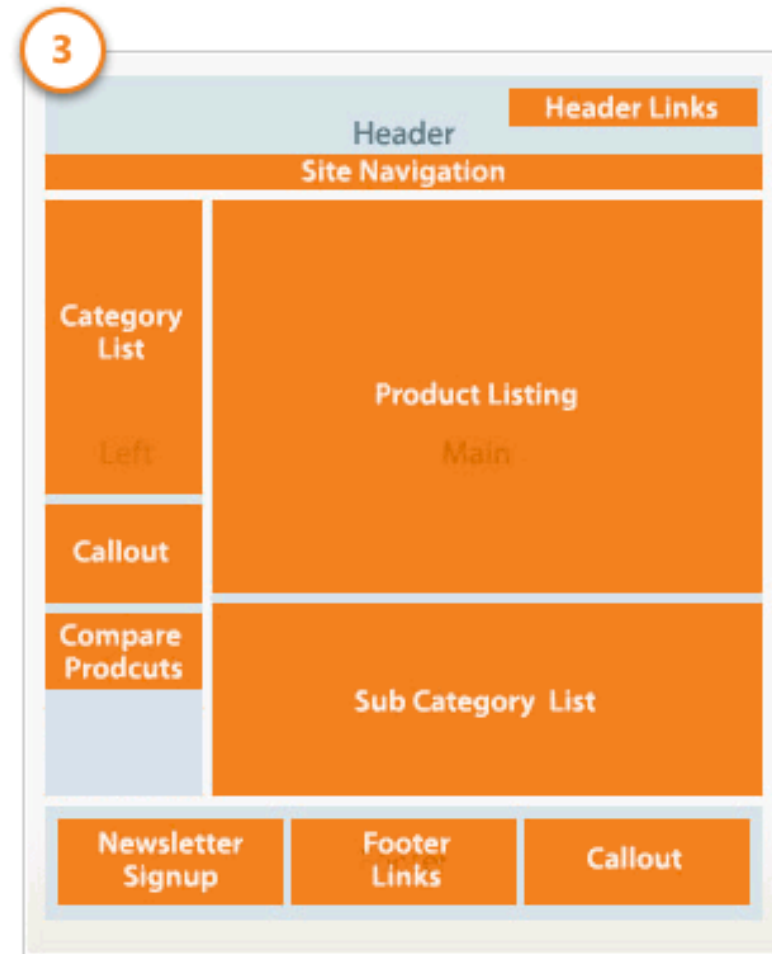
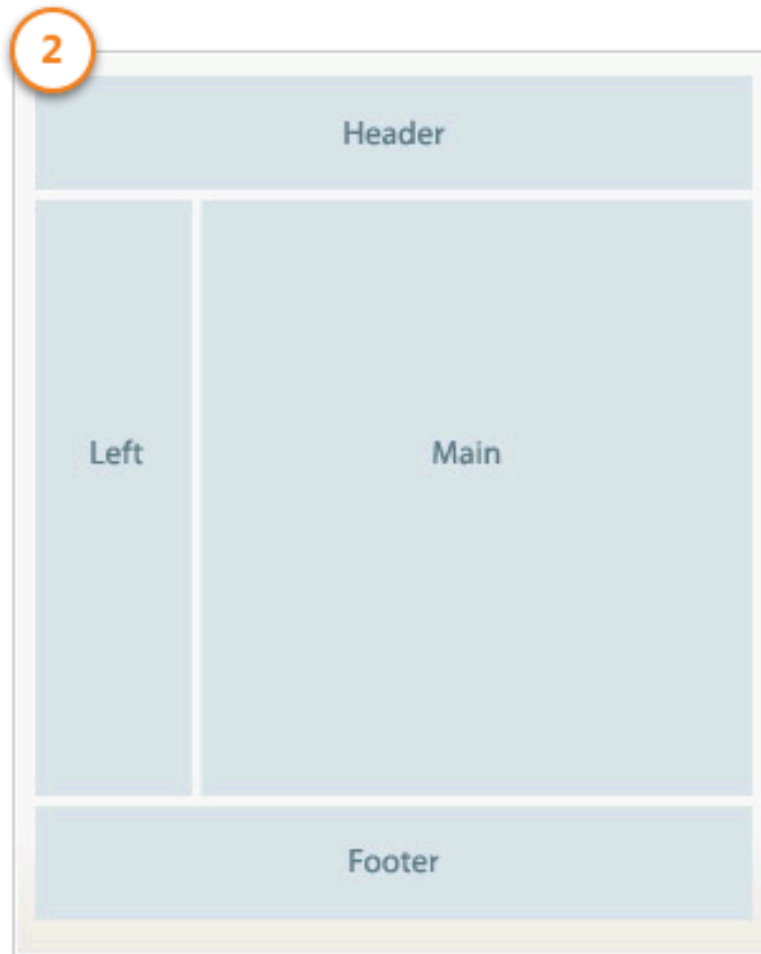
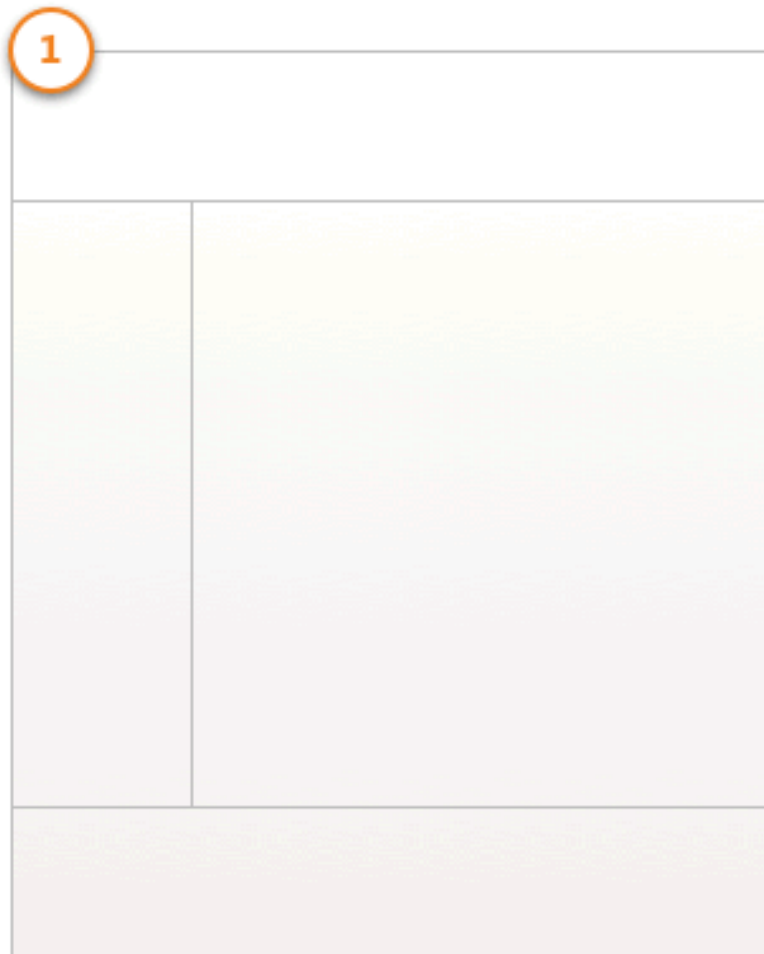
**Trudność w umieszczeniu  
elementów na stronie**



## Układanie elementów

Umieszczanie elementów na stronie, przede wszystkim w poziomie (ale także w pionie), zawsze było zadaniem nietrywialnym, ponieważ... do niedawna nie było narzędzia zaprojektowanego, aby rozwiązać ten problem...

# Tradycyjne rozmieszczenie elementów strony



**Flexbox nie jest**  
**rozwiązaniem**

# Flexbox ma problemy ze „złożonymi” układami komponentów



**Sebastien Saunier**

CTO at Le Wagon

Get in touch

**3,536**

contributions

**758**

followers

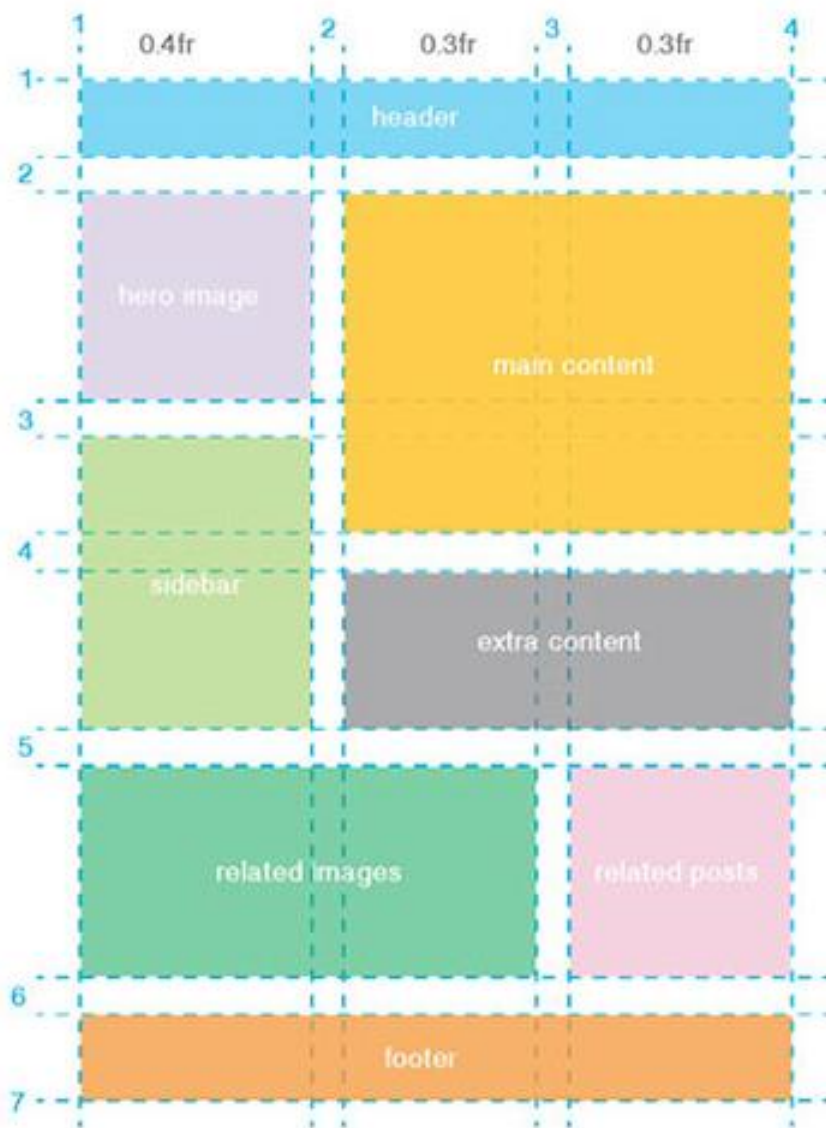
**119**

repositories

## More about Sebastien

I am a full-stack developer with 9 years of experience working in startups. Scratching your own itch? Working on a startup project? Lots of ideas but frustrated that you can't code them? Confused with agencies' or freelances' quotes? No clue when a developer is talking? Time to learn to code with Le Wagon with a 9-week intensive and immersive bootcamp!

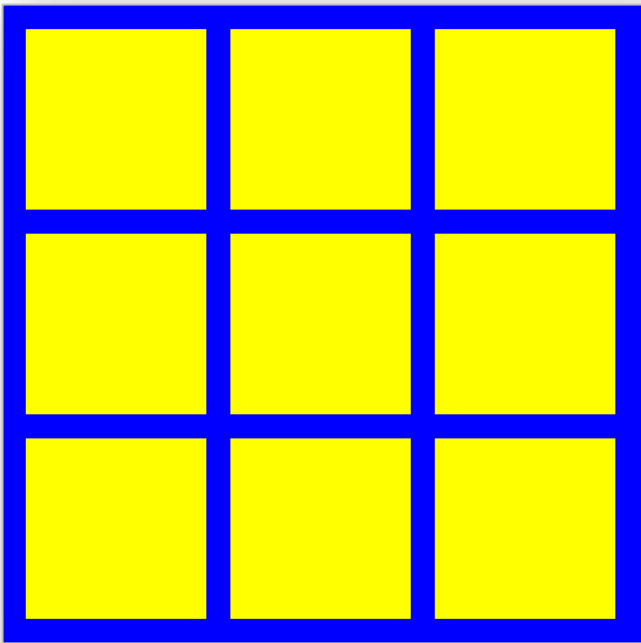
# CSS **Grid** – konstruktor dwuwymiarowych maketów



**CSS Grid** to najnowszy i najbardziej zaawansowany sposób umieszczania elementów na stronie. W przeciwieństwie do wszystkich wcześniej wymienionych metod, pozwala kontrolować rozmieszczenie elementów jednocześnie zarówno w poziomie, jak i w pionie.



```
1 <main>
2   <div></div>
3   <div></div>
4   <div></div>
5   <div></div>
6   <div></div>
7   <div></div>
8   <div></div>
9   <div></div>
10  <div></div>
11 </main>
12 <style>
13   div {
14     background: yellow;
15   }
16
17   main {
18     background: blue;
19
20     display: grid;
21     grid-template-columns: 150px 150px 150px;
22     grid-template-rows: 150px 150px 150px;
23     grid-gap: 20px;
24     padding: 20px;
25   }
26 </style>
```

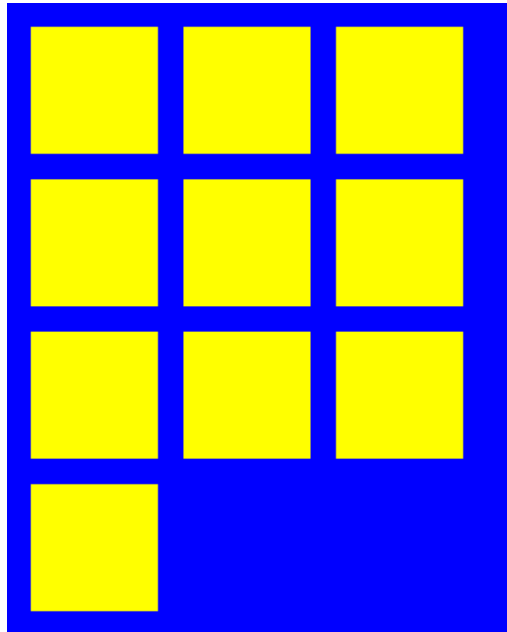


## Siatka CSS

**Grid** możemy włączyć dla dowolnego tagu, używając **display:grid** , a następnie używając właściwości **grid-template-columns** i **grid-template-rows**, aby odpowiednio ustawić: liczbę i szerokość kolumn i wierszy, które będą w utworzonej siatce. Wszystkie elementy podrzędne zostaną automatycznie umieszczone w komórkach siatki. Aby ustawić rozmiar kolumn i wierszy , oprócz znanych już jednostek miary, możesz użyć **fr** ( na przykład: **2,5 fr** ) jednostka ta określa część wolnej przestrzeni. Odstęp między komórkami można ustawić za pomocą właściwości **grid-gap** .

# CSS Grid

```
main {  
  background: blue;  
  
  display: grid;  
  grid-template-columns: repeat(3, 100px);  
  grid-auto-rows: 100px;  
  grid-gap: 20px;  
  padding: 20px;  
}
```



Jeżeli na początku nie wiemy ile będzie elementów( i co za tym idzie wierszy ),

możemy ustawić wysokość dla

wszystkich możliwych wierszy za pomocą

jednej właściwości:

**grid-auto-rows: 200px**

;

<https://xsltdev.ru/css/grid-auto-rows/>

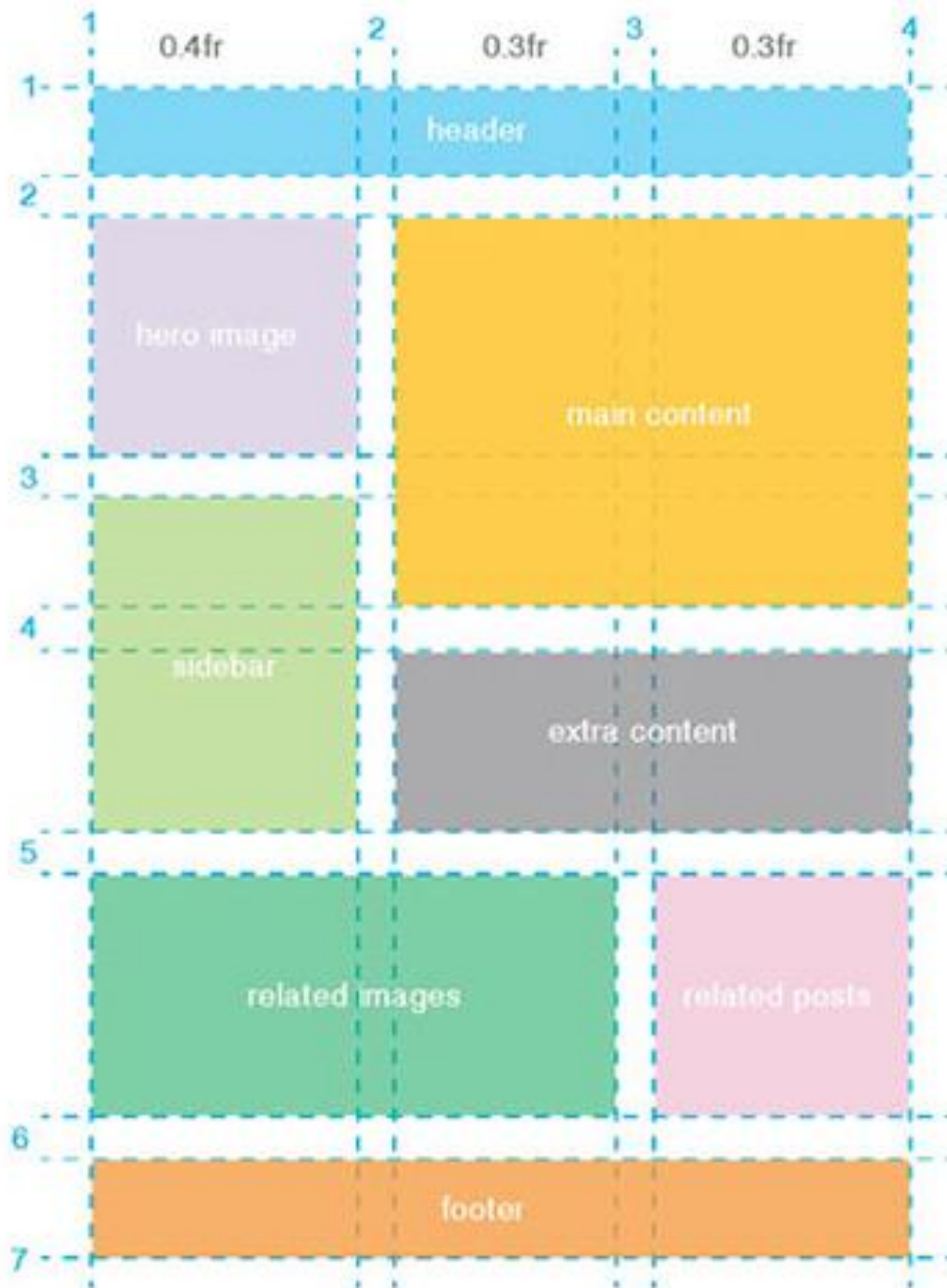
## CSS Grid – minmax()

funkcja **minmax()**. służy do ustawiania rozmiaru wierszy i/lub kolumn oraz umożliwia określenie granic, pomiędzy którymi przeglądarka może wybrać rozmiar ( przeglądarka będzie dążyć do ustawienia wartości maksymalnej , ale będzie respektować istniejące ograniczenia siatki jako całości ).

## CSS Grid – auto-fill и auto-fit

wartości **auto-fit** i **auto-fill** używany podczas umieszczania elementów w rzędzie. Uwzględniając szerokość elementów i dostępną przestrzeń w **grid-kontenerze** .

**Linie siatki**



# Grid Lines

W Grid istnieje koncepcja linii - linii oddzielających wiersze i/lub kolumny w siatce. Za pomocą tych linii (a dokładniej ich numerów) możemy umieścić elementy wewnątrz siatki nie w kolejności, ale w potrzebnych komórkach. W tym celu potrzebujemy następujących właściwości: **grid-column-start**, **grid-column-end**, **grid-row-start**, **grid-row-end** - które odpowiednio określają numery początkowej i końcowej linii pionowej oraz początkową i końcową linię poziomą. *Ale to nie jest najwygodniejszy sposób...*

# CSS Grid Areas

```

<main>
  <div class="cat"></div>
  <div class="dog"></div>
  <div class="fish"></div>
  <div class="bird"></div>
</main>

```

```

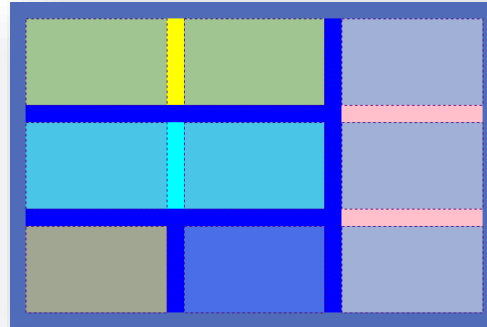
.cat{ background: yellow; grid-area: cat; }
.dog{ background: pink; grid-area: dog; }
.fish{ background: cyan; grid-area: fish; }
.bird{ background: orange; grid-area: bird; }

```

```

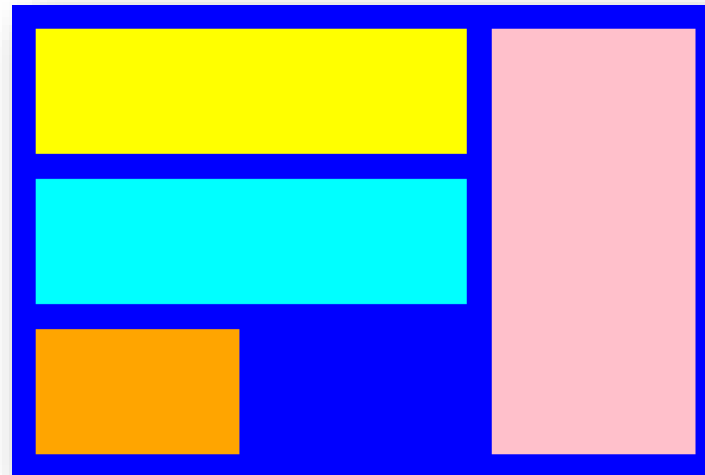
main {
  display: grid;
  background: blue;
  grid-gap: 20px;
  padding: 20px;
  grid-template-columns: 1fr 1fr 1fr;
  grid-template-rows: 100px 100px 100px;
  grid-template-areas: "cat   cat   dog"
                      "fish  fish  dog"
                      "bird  .    dog"
}

```



## CSS Grid Areas

CSS Grid umożliwia utworzenie układu w formie tekstowej. Aby to zrobić, każdemu elementowi siatki należy nadać nazwę za pomocą właściwości **grid-area**. I użycie właściwości **grid-template-areas** rozmieszczenie elementów w siatce określamy za pomocą podanych nazw elementów.





Przyda się

# Zadania

# Ukończ grę przez siatkę CSS !

## GRID GARDEN

Level 25 of 28

Now there is a 75 pixel column of weeds on the left side of your garden. 3/5 of the remaining space is growing carrots, while 2/5 has been overrun with weeds.


Use `grid-template-columns` with a combination of `px` and `fr` units to make the necessary columns.

```
1 #garden {  
2   display: grid;  
3   grid-template-columns: 75px 3fr 2fr  
4   grid-template-rows: 100%;  
5 }  
6  
7  
8  
9  
10  
11  
12  
13  
14
```

Next

Grid Garden is created by [CodePip](#) • [GitHub](#) • [Twitter](#) • [English](#)

Want to learn CSS flexbox? Play [Flexbox Froggy](#).



<https://cssgridgarden.com/>



Lorem ipsum dolor sit amet conse ctetur adipiscing elit

\$159.86

Add to cart

Lorem ipsum dolor ctetur adipiscing elit

\$159.86

Add to cart

Dolor sit amet conse ctetur adipiscing elit

\$159.86

Add to cart

Lorem ipsum dolor sit amet

\$159.86

Add to cart



Conse ctetur adipiscing elit

\$159.86 \$172.98

Add to cart

Lorem ipsum dolor sit amet conse ctetur adipiscing

\$159.86 \$172.98

Add to cart

Set mgan dolor sit amet conse ctetur adipiscing elit

\$159.86 \$172.98

Add to cart

Adipiscing elit

\$159.86 \$172.98

Add to cart

# Zadanie