

Uwaga: w tym pliku zostały przedstawione jedynie ogólne informacje odnośnie flexboxa i grida. Osoby, które po raz pierwszy stykają się z tymi zagadnieniami zachęcam do zapoznania się prezentacjami dostępnymi na YT (poniżej wkleiłem linki) oraz prezentacjami i przykładami na stronie css-ticks (linki również znajdują się w tym pliku). Następnie proszę spróbować stworzyć prosty układ w oparciu o flexbox i/lub grid.

W razie jakichkolwiek pytań proszę o kontakt wojcikpa@kul.pl jestem również codziennie dostępny na messenger.

Flexbox

Przy domyślnych stawieniach elementy blokowe układają się jeden pod drugim. Chcąc ułożyć elementy blokowe obok siebie mogliśmy skorzystać z właściwości `float` z wartością `left` lub `right` lub właściwości `display` ustawionej na wartość `inline-block`.

Kolejnym sposobem na ustawienie elementów obok siebie jest wartość `flex`. Reguła

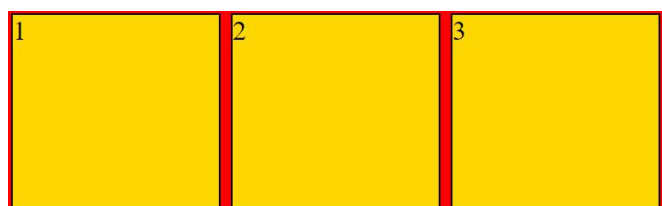
```
display: flex;
```

ustawiona dla rodzica spowoduje, że „dzieci” ułożą się poziomo (obok siebie).

```
#flex{
  background-color: red;
  display: flex;
  height: 120px;
}
```

```
<div id="flex">
  <p></p>
  <p></p>
  <p></p>
</div>
```

```
#flex>p{
  background-color: gold;
  width: 120px;
  border: 1px solid black;
  margin: 3px;
}
```



Element nadrzędny nazywamy **flex-container (container)** a jego „dzieci” **flex-items (items)**.

Aby wykorzystać możliwości „flexbox” ustawiamy właściwości zarówno dla elementu container jak również dla elementów items

Właściwości dla elementu container

```
flex-direction
flex-wrap
justify-content
align-items
align-content
```

Właściwości elementów items

```
order
flex-grow
flex-shrink
align-self
```

Dokładny opis tych właściwości oraz przykłady można znaleźć na stronie

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Również na YT można znaleźć wiele prezentacji tworzenia stron za pomocą flexbox

np. <https://www.youtube.com/watch?v=yFHgPOMF3V4&t=522s>

Grid

Jeszcze większe możliwości do flexbox-a daje grid. Pozwala on stworzyć siatkę na kształt tabeli i umieszczać poszczególne elementy HTMLa w odpowiednich „komórkach tabeli”. Zaletą tego rozwiązania jest to, że tworzymy je po stronie CSS i łatwo możemy przygotować kilka układów dla różnych rozdzielczości nie zmieniając struktury HTML.

Podobnie jak przy flexbox-ie ustawiamy właściwości dla elementu container oraz elementów items

Dokładny opis właściwości dla container oraz items można znaleźć na stronie

<https://css-tricks.com/snippets/css/complete-guide-grid/>

Mini kursy grida można znaleźć też na YT

np.

<https://www.youtube.com/playlist?list=PLAZsLPvWI5XDGiUFrWpfsWxf4snAwq7u9>