

# Flexbox

- do tworzenia layoutów (interfejs użytkownika)
- pozycjonowanie elementu (elementów) w innym elemencie

*Do tego samego co np. CSS Grid, którym zajmiemy się później*

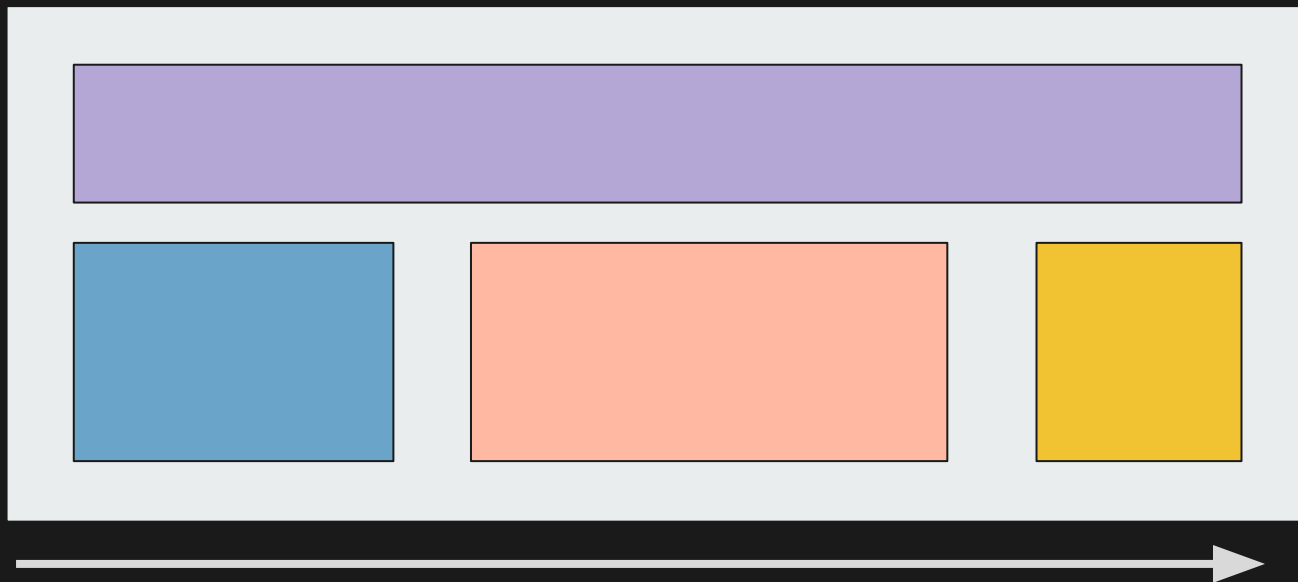


# O co chodzi z flexbox

- układ jednowymiarowy (kontrola w pionie, albo poziomie) - oczywiście mamy pewien wpływ na zachowanie w osi prostopadłej do osi głównej, ale jest to bardzo ograniczone.
- zarządzanie wolną przestrzenią kontenera. To jest kluczowe we flexie mamy przestrzeń, jeśli jest w niej wolne miejsce, to wchodzi flex.
- Zastępuje (nie tylko) float przy tworzeniu layoutów.



# Kontener, elementy elastyczne i oś główna



# Właściwości kontenera

`display: flex; /* inline-flex */`

`flex-direction: row; /* column, row-reverse, column-reverse */`

`flex-wrap: nowrap; /* wrap, wrap-reverse */`

`flex-flow: row nowrap; /* skrót dla flex-direction i flex-wrap */`

`justify-content: flex-start; /* flex-end, center, space-around, space-between, space-evenly */`

`align-items: stretch; /* flex-start, flex-end, center, baseline */`

`align-content: stretch; /* flex-start, flex-end, center, space-between, space-around */`



# Właściwości elementu

`order: 0;` /\* liczba całkowita (wartości dodatnie i ujemne) \*/

`flex-grow: 0;` /\* wartość bezwzględna (wartość od 0 w górę) \*/

`flex-shrink: 1;` /\* wartość bezwzględna (wartość od 0 w górę) \*/

`flex-basis: auto;` /\* wielkości z różnymi jednostkami \*/

`flex: 0 1 auto;` /\* skrót dla flex-grow, flex-shrink i flex-basis \*/

`align-self: auto;` /\* stretch, center, flex-start, flex-end, baseline \*/

