# Flexbox – Wytłumaczone po ludzku (dla Adama)

## 1. justify-content – rozstawianie elementów POZIOMO

Ustawia jak dzieci (np. divy) są rozłożone w poziomie – czyli czy mają być do lewej, na środku, czy porozstawiane równo.

**flex-start:** Elementy do lewej strony (domyślnie).

**flex-end:** Elementy do prawej strony kontenera.

**center:** Elementy są wyśrodkowane.

**space-between:** Pierwszy przyklejony do lewej, ostatni do prawej – równe odstępy między nimi.

**space-around:** Równe odstępy dookoła każdego elementu (czyli po bokach też).

**space-evenly:** Wszędzie równe odstępy – pomiędzy i po bokach.

## 2. align-items – ustawianie elementów w PIONIE

Dotyczy jednego rzędu. Ustawia, gdzie w pionie mają być dzieci (np. czy przy górze, na środku itd.).

**stretch:** Domyślnie – wszystkie dzieci rozciągają się na pełną wysokość kontenera.

**flex-start:** Wyrównane do góry kontenera.

**flex-end:** Wyrównane do dołu kontenera.

**center:** Wyśrodkowane w pionie.

**baseline:** Wyrównane do linii bazowej tekstu.

## 3. flex-direction – kierunek układu elementów

Mówi, czy elementy mają się układać poziomo czy pionowo.

**row:** Elementy w poziomie od lewej do prawej (domyślnie).

**row-reverse:** Elementy w poziomie, ale od prawej do lewej.

**column:** Elementy ustawione pionowo – jeden pod drugim.

**column-reverse:** Pionowo, ale od dołu do góry.

## 4. flex-wrap – czy zawijać elementy, jeśli się nie mieszczą

Gdy elementów jest za dużo, decyduje, czy mają się zawinąć do nowego wiersza.

**nowrap:** Wszystko w jednej linii, nawet jak się nie mieści (domyślnie).

**wrap:** Elementy zawijają się do nowej linii, jak braknie miejsca.

**wrap-reverse:** Jak wrap, ale w odwrotnym kierunku (na górę zamiast w dół).

## 5. align-content – układ wielu wierszy w pionie

Ustawia jak są rozmieszczone linie (rzędy) elementów – tylko jeśli jest więcej niż jeden rząd.

**stretch:** Rozciąga wszystkie rzędy, żeby zajęły całą wysokość kontenera.

**flex-start:** Rzędy do góry.

**flex-end:** Rzędy do dołu.

**center:** Rzędy na środku.

**space-between:** Równe odstępy między rzędami.

**space-around:** Równe odstępy wokół każdego rzędu.

## 6. align-self – ustawienie jednego elementu w pionie

Pozwala nadpisać align-items tylko dla konkretnego dziecka.

**auto:** Domyślnie – bierze z align-items kontenera.

**stretch:** Rozciąga się na wysokość kontenera.

**flex-start:** Do góry.

**flex-end:** Na dół.

**center:** Na środku.

**baseline:** Do linii tekstu.

## 7. flex-grow, flex-shrink, flex-basis – kontrola rozmiaru dzieci

Te trzy właściwości mówią, jak dziecko ma się rozciągać, kurczyć i jaką ma mieć bazową szerokość.

**flex-grow:** Czy dziecko może się rozciągać, by zająć więcej miejsca. 0 = nie, 1 = może.

**flex-shrink:** Czy może się zmniejszać, gdy brakuje miejsca. 0 = nie, 1 = tak.

**flex-basis:** Domyślny rozmiar przed podziałem miejsca – działa jak width, ale elastyczny.

## 8. flex – skrót do 3 w 1

Skrót do wpisania flex-grow, flex-shrink i flex-basis jednocześnie. Przykład:

flex: 1 0 200px; // rośnie, nie kurczy się, ma bazę 200px