

Notatki Flexbox

Notatnik: Coding

Utworzono: 18.02.2019 09:12

Zaktualizo... 18.02.2019 14:08

Autor: danuta.cybulska.biz@gmail.com

Znaczniki: Flexbox

Podstawy

Zanim zaczniemy z *Flexem*... Kilka spraw podstawowych. Aby w ogóle zacząć stylować za pomocą Flexboxa musimy go "włączyć". Robi się to poprzez deklarację:

display: flex; (lub inline-flex)

Przypisujemy ją "rodzicowi", czyli kontenerowi, który ma wewnątrz siebie elementy ("dzieci")

I jeszcze dwie sprawy, które będą istotne. Osie według, których będziemy pozycjonować.

- [main-axis](#) - oś główna;
- [cross-axis](#) - oś przeciwna;

Oś pionowa (OY) biegnie z góry na dół, a oś pozioma (OX) od lewej do prawej. Trzeba mieć to na uwadze, kiedy będziemy przedstawiać główną oś na przeciwną i odwrotnie.

"Rodzice"

Na "rodzicu" deklarujemy inne własności niż na "dziecku". Ale spokojnie, zaraz je wszystkie wypiszę dla lepszego zapamiętania.

- [flex-direction](#) - ustawiamy kierunek względem main-axis, dostępne wartości to: row (wzdłuż rzędu), column (wzdłuż kolumn), row-reverse (odwrotnie, wzdłuż rzędu), column-reverse (odwrotnie, wzdłuż kolumn)
- [flex-wrap](#) - zwijanie linii, dostępne wartości: no-wrap (nie przenosi do kolejnego wiersza), wrap (przenosi do kolejnego wiersza), wrap-reverse (odwrócony kierunek osi przeciwną cross-axis);
- [justify-content](#) - porządkuje elementy względem osi głównej, czyli main-axis, dostępne wartości: flex-start, flex-end, center, space-around, space-between, space-evenly. Pod linkiem znajdziesz różnice, jak się układają dane elementy;
- [align-content](#) (musi być więcej niż jedna linia, aby zauważyć działanie tej właściwości) - porządkuje elementy wzdłuż osi przeciwną, czyli cross-axis, dostępne wartości: flex-start, flex-end, stretch (domyślnie), space-around, space-between, center;
- [align-items](#) - porządkuje elementy wzdłuż cross-axis, dostępne wartości: flex-start, flex-end, stretch (domyślnie), center, baseline - wyrównanie tak, aby linia tekstu była równa pomiędzy elementami w danej linii, a pierwszy tekst w elemencie to może być np. nagłówek;

W tych dwóch ostatnich przypadkach warto ustawić na dzieciach max-width i max-height.

"Dzieci"

A teraz deklaracje ustawiane na "dzieciach", czyli elementach wewnątrz kontenera - "rodzica".

- **[align-self](#)** - przy tej własności możemy wypozycjonować każdy element niezależnie od pozostałych. Pod uwagę bierzemy cross-axis (czyli oś przeciwną). Wartości, jakie może przyjmować to: flex-start, flex-end, center, baseline, stretch, auto - domyślnie układ rodzica.
- **[order](#)** - czyli kolejność elementów. "0" jest wartością domyślną i ustawia elementy zgodnie z kodem html. Uporządkowanie następuje od najmniejszej liczby do największej np. -3, 0, 2, 5. Jeśli występują dwie takie same wartości np. -3, 0, 2, 2; wtedy o miejscu tych dwóch elementów z taką samą wartością decyduje kolejność w kodzie html.
- **[flex-shrink](#)** - to współczynnik kurczenia elastycznego elementu-dziecka. Jeśli rozmiar elementów elastycznych jest większy niż kontener-rodzic elastyczny, elementy-dzieci kurczą się, aby się dopasować. Pod uwagę bierze się oś przeciwną, wartość przyjmuje zero i więcej. Domyślną wartością jest jedynka "1". "0" oznacza wyłączenie kurczenia się elementów. Można ustawiać również części ułamkowe (nie tylko liczby całkowite) np. 1.3.
- **[flex-grow](#)** - jest przeciwnością **[flex-shrink](#)** i określa, jaka część dostępnego miejsca w kontenerze-rodzicu powinna zostać przypisana do tego elementu (współczynnik wzrostu). Wartości to liczby całkowite, domyślna wartość to "0". Jeśli jest włączona ta wartość to nie zadziała **[justify-content](#)**. Pod uwagę bierzemy oś główną.
- **[flex-basis](#)** - ustala rozmiar elementu w kontekście osi głównej. Przyjmuje jednostki: %, px, vh, vw, rem lub auto (domyślnie). Nie zadziała, jeśli jest włączona **[position: absolute](#)**; na elemencie.

Wszelkie max-height/max-width i min-height/min-width respektowane są przez właściwości flex-shrink oraz flex-grow.

Skróty (shorthands)

Zawsze musimy szukać sposobu, aby nie pisać zbyt wiele kodu. Dlatego wymyślono skrócone notacje. Jeśli nie czujesz się jeszcze pewnie w Flexboxie to możesz rozpisywać deklaracje krok po kroku, ale ... warto wyrobić sobie nawyki pisania skrótowo.

Flex-flow

To połączenie dwóch deklaracji: **[flex-direction](#)** oraz **[flex-wrap](#)**. Ustawiamy ją na rodzicu, a jej wartość domyślna to Row (kierunek rząd) Nowrap (nie zwiija).

flex-flow: row nowrap;

Flex

Połączenie trzech deklaracji: **[flex-grow](#)**, **[flex-shrink](#)** oraz **[flex-basis](#)**. Ustawiamy na dziecku. I mamy tu kilka opcji. Może występować jako jedna wartość, dwie lub trzy. W zależności ile ich jest i czy są jakieś jednostki, wtedy przypisywane są odpowiednie deklaracje.

- Jedna wartość: bez jednostek - wtedy oznacza to, że jest to deklaracja **[flex-grow](#)**; z jednostką - **[flex-basis](#)**.
- Dwie wartości: bez jednostek: pierwsza będzie **[flex-grow](#)**, a druga - **[flex-shrink](#)**; jeśli druga ma jednostkę to będzie to **[flex-basis](#)**.
- Trzy wartości - deklaracje układają się po kolei: **[flex-grow](#)**, **[flex-shrink](#)** i **[flex-basis](#)**.

flex: 1 1 30px;
