

Tema: Flexbox-uri în CSS. "Responsive web design" și flexbox

După această lecție:



- vei afla ce este un *flexbox*;
- vei cunoaște proprietățile CSS din modulul FLEX;
- vei învăța cum cu ajutorul proprietăților CSS din modulul FLEX și ale interogărilor Media poți defini conținuturi *responsive*.

La lecția trecută am utilizat elementul **DIV** și proprietatea **float**, combinată cu **media Query**, pentru machetarea paginii web. La această lecție vom învăța și alte proprietăți, ce pot fi utilizate pentru machetarea paginilor web.

Modulul FLEX folosit pentru machetare

Acest modul format din proprietăți CSS, ușurează procesul de proiectare a machetei adaptive a paginii web. În acest caz nu se vor folosi proprietățile **float** și cele de poziționare, ci doar proprietăți din grupul **flex**.

Pentru început, este necesar să se înțeleagă, că pentru a utiliza proprietățile din modulul **flex**, trebuie definit un element *container*, care va stoca mai multe *elemente-urmași*, elemente ce vor fi amplasate în interiorul elementului-container.

Containerul va deveni *responsive*, dacă pentru el se definește proprietatea **display: flex**;

De exemplu, pentru următorul bloc format din alte trei blocuri-urmași, scris în HTML:



```
<div class="cards_prod">
  <div class="card">
    
    <h3>Cappuccino</h3>
  </div>
  <div class="card">
    
    <h3>Latte</h3>
  </div>
  <div class="card">
    
    <h3>Espresso</h3>
  </div>
</div>
```

Voi scrie declarația **display: flex;**:



```
.cards_prod {
  display: flex;
  max-width: 100%;
  padding: 0;
  margin-top: 85px;
}
```

Și atunci, acest bloc va servi drept container pentru toate elementele amplasate în el. Poți plasa în el poze sau elemente DIV. Eu am amplasat alte 3 DIV-uri, cu clasa **.card**. Pentru elementul-container mai trebuie definite, suplimentar, și alte proprietăți de stil. De asemenea, pot fi definite proprietăți de stil și pentru elementele-urmași. Să vedem ce proprietăți mai sunt în modulul **flex**.

Proprietăți din modulul FLEX, folosite pentru elementul-container

Proprietatea **flex-direction** definește direcția în care containerul va amplasa elementele-urmași. Valori posibile pentru această proprietate:

- **column** – elementele-urmași se vor prezenta sub forma unei coloane;
- **column-reverse** – elementele-urmași se vor prezenta sub formă de coloană, dar de jos în sus;
- **row** – elementele-urmași se vor prezenta pe orizontală – de la stânga la dreapta;
- **row-reverse** – elementele-urmași se vor prezenta pe orizontală – de la dreapta la stânga.

Proprietatea **flex-wrap** definește dacă elementele-urmași trebuie să treacă din linie nouă sau nu, atunci când se umple spațiul de afișare, pe orizontală. Valori posibile:

- **wrap** – va trece din linie nouă, la umplerea containerului în lățime;
- **nowrap** – nu va trece din linie nouă. Este valoarea predefinită;
- **wrap-reverse** – afișarea inversă a elementelor, cu trecerea din rând nou.

Proprietatea **flex-flow** reprezintă forma prescurtată pentru proprietățile **flex-direction** și **flex-wrap**.

Proprietatea **justify-content** se utilizează pentru alinierea elementelor din *flexbox*, pe orizontală. Valori posibile:

- **center** – elementele se aliniază pe mijlocul containerului;

- **flex-start** – elementele se aliniază de la începutul containerului – este valoarea predefinită;
- **flex-end** – aliniază elementele de la sfârșitul containerului, spre început;
- **space-around** – afișează elementele cu spațiu în jurul acestora;
- **space-between** – afișează elementele din *flexbox* cu spațiu între linii.

Proprietatea **align-items** se folosește pentru alinierea elementelor pe verticală. Valori posibile:

- **center** – alinierea se realizează pe mijlocul containerului;
- **stretch** – întinde elementele pentru a umple containerul. Este valoarea predefinită;
- **flex-start** – aliniază elementele pe partea de sus a containerului;
- **flex-end** – aliniază elementele pe partea de jos a containerului;
- **baseline** – aliniază elementele astfel încât nivelele de la bază să coincidă.

Centrarea elementelor cu ajutorul proprietăților *flex*

Pentru centrarea elementelor în cadrul containerului – pe orizontală și pe verticală – se vor folosi proprietățile **justify-content** și **align-items**.

Pentru exemplul prezentat mai sus, voi scrie stilul, utilizând proprietăți din **flex**, pentru a alinia elementele pe mijlocul containerului:



```
.cards_prod {
  display: flex;
  flex-flow: row wrap;
  justify-content: center;
  align-items: center;
  max-width: 100%;
  padding: 0;
  margin-top: 85px;
}
```

Suplimentar, voi scrie stilul pentru carduri, care sunt formate dintr-o imagine și denumirea produsului. Cardurile vor avea lățimea de 27%, atunci când vor fi prezentate pe un ecran mare:



```
.card {
  width: 27%;
  margin: 3%;
}
```

```
padding: 5px;
}
```

Rezultatul aplicării acestor stiluri va fi:



Observă cum au fost aranjate elementele pe mijlocul containerului – și pe orizontală și pe verticală.

Dar, la începutul lecției am afirmat că proprietățile din grupul **flex**, pot fi utilizate și pentru a realiza un design *responsive* pentru acest element-container. Pentru aceasta voi scrie un **media query**:



```
@media screen and (max-width: 700px) {
  .cards_prod {
    display: flex;
    flex-flow: column wrap;
    justify-content: center;
    align-items: center;
    padding: 0;
    margin-left: 7%;
    margin-right: 5%;
  }
  .cards_prod .card {
    width: 80%;
  }
}
```

Să "tălmăcim" acest **media query**... Deci, atunci când lățimea ferestrei browserului devine mai mică decât 700px, blocul-container va prezenta elementele pe verticală, centrate în

container, elementele trecând unul sub altul. Blocurile cu clasa **.card**, în acest caz, vor ocupa 80% din lăţimea containerului. Vezi aici rezultatul:



Voi adăuga şi un filmuleţ, ca să vezi efectul de trecere şi de adaptare a elementelor la lăţimea ecranului – filmuleţul: **flex_responsive.mp4**.

Poţi încerca să aplici şi alte valori proprietăţilor, ca să vezi cum se vor aranja elementele în container.

Suplimentar, în grupul **flex**, sunt proprietăţi de stil ce pot fi setate elementelor-urmaşi. Să vedem unele din ele.

Proprietăţile modului FLEX pentru elementele-urmaşi

Proprietatea **order** – defineşte ordinea elementelor-urmaşi în containerul flexibil.

Proprietatea **flex-grow** – defineşte cum va creşte un element în raport cu alte elemente din containerul flexibil. Valoarea este una numerică. Valoarea implicită este 0.

Proprietatea **flex-shrink** – defineşte cum se va micşora un anumit element în raport cu

restul conținutului containerului. Valoarea este una numerică. Valoarea implicită este 1. Proprietatea **flex-basis** – definește valoarea inițială a dimensiunii elementului din *flexbox*.

Doar dacă voi adăuga stilul *inline* celui de-al doilea card, setându-i `order:3`; astfel:



```
<div class="cards_prod">
  <div class="card">
    
    <h3>Cappuccino</h3>
  </div>
  <div class="card" style="order:3;">
    
    <h3>Latte</h3>
  </div>
  <div class="card">
    
    <h3>Espresso</h3>
  </div>
</div>
```

...prezentarea elementelor în container se va schimba:



Observă cum cardul cu produsul "Latte", care în codul HTML era al doilea, a migrat în a treia poziție.

Bine...să continuăm...

Proprietatea **flex** – reprezintă forma prescurtată pentru proprietățile **flex-grow**, **flex-shrink**, **flex-basis**. De exemplu, **flex: 0 0 300px**.

Proprietatea **align-self** – definește alinierea pentru elementul-urmas ales, înăuntrul containerului flexibil. Valori posibile: *auto*, *stretch*, *center*, *flex-start*, *flex-end*, *baseline*.

Încheiem aici lecția. Dacă ești curios poți accesa următorul link: https://www.w3schools.com/css/css_grid.asp, pentru a cerceta cum poate fi utilizat un alt modul de proprietăți pentru machetarea paginilor web – modulul **grid**.

Să repetăm ce am învățat în cadrul acestei lecții:



- pentru a utiliza flexbox-urile la definirea conținuturilor *responsive* este necesar să fie creat un element HTML de tip container, în interiorul căruia vor fi definite mai multe elemente urmași;
- pe ecrane mici elementele urmași pot fi plasate unul sub altul, pe verticală, iar pe ecranele mari – elementele urmași pot fi plasate unul lângă altul, pe orizontală;
- în cazul în care se dorește prezentarea elementelor-urmași și pe alte tipuri de ecrane, de dimensiuni medii de exemplu – se vor găsi mai multe puncte de control și va fi definită prezentarea cu ajutorul mediaQuery-lor suplimentare.

Noi însă, încheiem aici studierea limbajului formal CSS. Să știi, că acest limbaj mai are și alte multe secrete, iar pentru a realiza pagini web frumos stilizate, va trebui mereu să urmărești ce a mai apărut nou în CSS...