Dzień dobry wszystkim tak wcześnie rano :)

To moje pierwsze szkolenie i mam nadzieję, że nie ostatnie, ale zobaczymy jak będzie.

Na początek wybrałam sobie temat z CSSów. Może też dlatego, że wydaje mi się, że przywiązujemy do nich zbyt mało uwagi. Z drugiej strony, patrząc na plany szkoleniowe zaczyna się to zmieniać :)

A teraz do celu….

Gdy zaczynałam kodować strony CSSy pisano inlinowo, a wszystko a cały layout strony opierał się na tabelach. Na początku były to tabele o stałej szerokości, bodajże 600px, bo nie było też czegoś takiego jak responsywność, standardowa rozdzielczość to 800x600px. Wszystko było stałe. istniała tylko jedna przeglądarka IE3 :)

Świat był prosty :)

A strony wyglądały tak….. i tak….. i tak. :)

Potem zaczęto używać osobnych plików CSS. Layout opart został na blokach. Zaczęto stosować DISPLAY. Najpierw było to nadal display: table, potem jednak przeło na FLOATy z ich CLEAR i klasą .clearfix, FLASHe ze ich ActionScript, POSITION: absolute, DISPLAY: inline-block, BOOTSTRAP Grid System, FLEX.

Aż w końcu zaczęto używać CSS GRID. Coś na co należy patrzeć jak na tabele :)

Wracamy do korzeni :)

Od października 2017 roku wszystkie PRAWIDŁOWE przeglądarki desktopowe, czyli Chrome, FF, Safari, Edge i większość przeglądarek mobilnych zaczęły całkowicie wspierać CSS GRID. To już prawie 1,5roku.

O IE10 nie mówię, bo nie obsługujemy.

A Jeśli chodzi o IE11 to wsparcie jest ograniczone. Można używać ale przy dodatkowej specyfikacji. Nie jest wspierany grid-gap oraz grid-template-areas. Są na to obejścia, ale polecam już we własnym zakresie znaleźć rozwiązania.

Pomocny może być artykuł z linku.

No dobrze to może do samego CSS grid. Chciałabym go wam przedstawić na podstawie przykładów i pokazać jego możliwości. Zacznijmy może od narysowania sobie takiej prostej tabeli 3/2.

(rysunek)

- display: grid

- grid-template-columns

- grid-gap

- wariacje grid-template-columns

długości, auto, repeat, max-content/min-content, , procenty, flex

FLEXIBLE LENGTH (elastyczna długość) inaczej FLEX reprezentowana jest przez jednostkę FR (fraction).

Jest to długość, która odzwierciedla cześć/proporcje pozostałego obszaru.

minmax(min, max), fit-content (podobne do auto)

- grid-template-rows

Teraz podpiszmy sobie linie każdej z tabel:

(rysunek)

I załóżmy, że chcemy narysować coś takiego:

(rysunek)

- grid-column - <start-line> / <end-line> | <start-line> / span <value>;

- grid-row - <start-line> / <end-line> | <start-line> / span <value>

Layouty gridowe możemy tworzyć jeszcze w jeden sposób używając GRID-AREAS

- grid-template-areas

- grid-area

Grid możemy pozycjonować podobnie jak obiekty Flexowe. Nie będę będę się więc nad tym rozwodzić tera bo i czasu brak na to.

Wspomne jeszcze tylko o :

- grid-auto-columns / grid-auto-rows

- grid-auto-flows