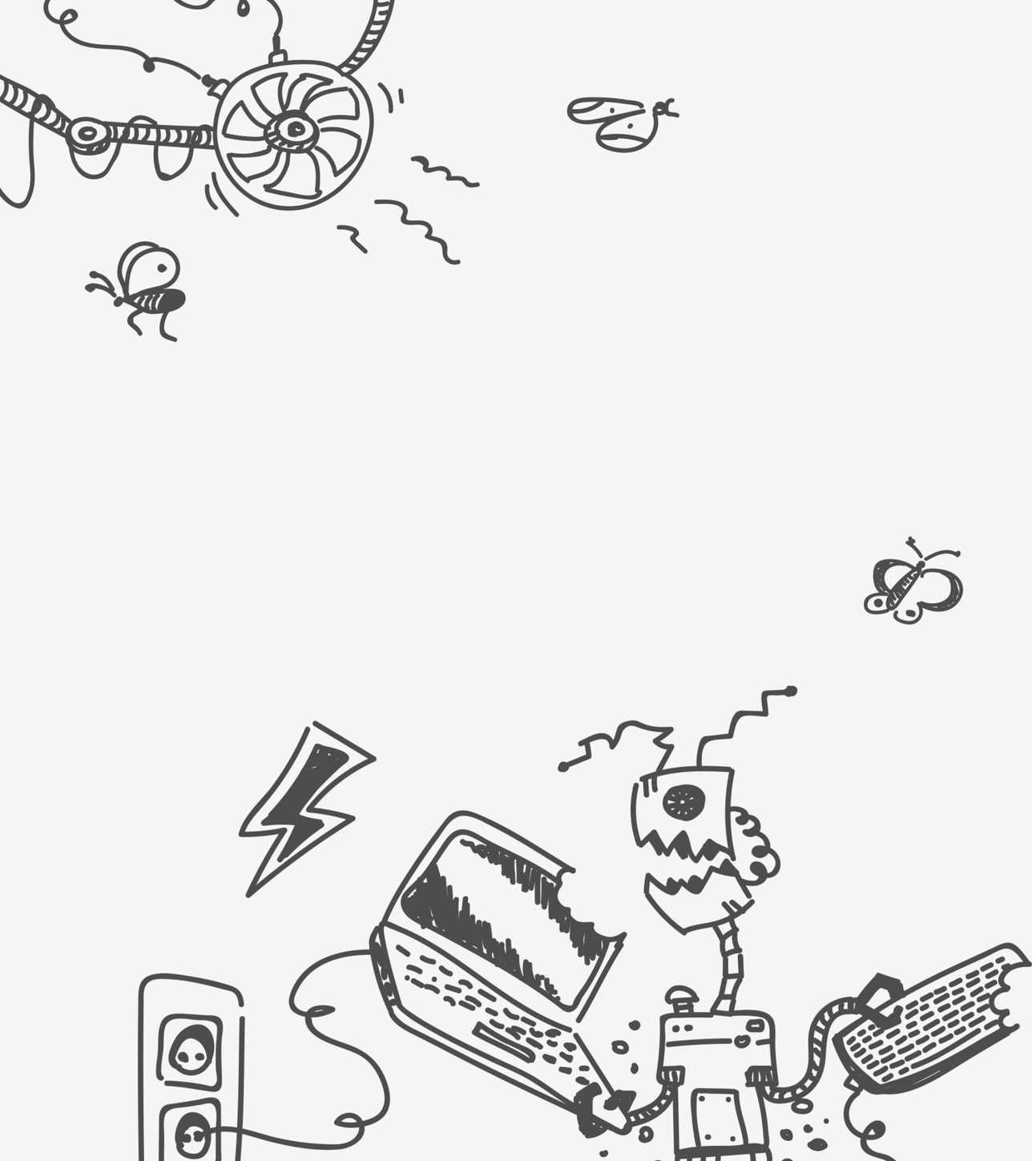


RWD Flexbox

Wprowadzenie

Flexbox

Korzystając z flexboxa do układania elementów na stronie doświadczaliśmy sytuacji, że elementy na stronie działały bardzo elastycznie. Za pomocą takich właściwości jak flex-direction lub flex-wrap, mogliśmy manewrować układaniem elementów w pionie i poziomie, oraz ustawiać warunki, w których elementy mają spadać pod siebie.

Teraz mając wiedzę o media queries oraz o technikach RWD możemy nauczyć się jak połączyć naszą wiedzę w całość, aby tworzyć całościowe układy stron internetowych, dostosowane zarówno do małych telefonów jak i dużych ekranów komputera.

W tym rozdziale poznamy kilka strategii i zasad jak korzystać z technologii flexbox przy mniejszych urządzeniach.

Kiedy używać flexbox?

Jak używać techniki flexbox?

Flexbox przyda się w następujących sytuacjach:

- do ustawiania elementów obok siebie
- do ustawiania maksymalnych odstępów między elementami (np. wertykalnie lub horyzontalnie względem kontenera). Do tego potrzebujemy skorzystać z właściwości justify-content i align-items)
- do układania elementów pod sobą (flex-wrap)

Oznacza to, że flexbox będzie potrzebny raczej do kontenerów (np. całej nawigacji, czy całych boksów) niż do poszczególnych elementów.

Warto przyjąć zasadę, że używamy techniki flexbox wtedy kiedy nie możemy osiągnąć pożądanego efektu za pomocą elementów inline - blokowych (np. gdy potrzebujemy ustawić dwa elementy obok siebie, nadać im paddingi, lub gdy potrzebujemy aby elementy były w idealnym odstępie między sobą)

Kiedy nie używać flexboxa?

Jak nie używać flexboxa

Flexboxa nie należy stosować w poniższych sytuacjach:

- kiedy możemy osiągnąć ten sam efekt za pomocą właściwości - display block, lub inline-block
- kiedy element musi być sztywnie ustawiony względem jakiegoś elementu (np. ozdobniki do list). Wtedy lepiej użyć pozycji relatywnej/absolutnej

Warto pamiętać, że flexbox zmienia zachowanie dzieci. Zbyt częste używanie tej właściwości powoduje, że nasz kod przestaje być czytelny i zrozumiały.

Przy tworzeniu CSS warto stosować zasadę, że im mniej kodu tym lepiej. Oznacza to, że powinniśmy używać tylko tych właściwości, które nam są niezbędne.

Strategia - warunkowe renderowanie

Jeśli elementy powinny znajdować się obok siebie tylko na dużym ekranie, a na mniejszych ekranach chcemy, aby wszystko było jedno pod drugim - wtedy warto ustawić display: flex tylko dla dużych ekranów.

```
.container {
  display: block;
  /* domyślna wartość */

  @media screen and (min-width: 640px) {
    display: flex;
  }
}
```

Desktop:

Mobile:

Strategia - spadanie elementów

Pierwszą strategią RWD przy budowaniu układów stron jest wyśrodkowanie zawartości kontenera dla **ekranów mobilnych** za pomocą **justify-content**. Na dużych ekranach zazwyczaj możemy pozwolić sobie na układanie elementów obok siebie, natomiast na telefonach komórkowych musimy elementy ułożyć jeden pod drugim.

Na następnym slajdzie znajdują się zdjęcia opisujące tę sytuacje.

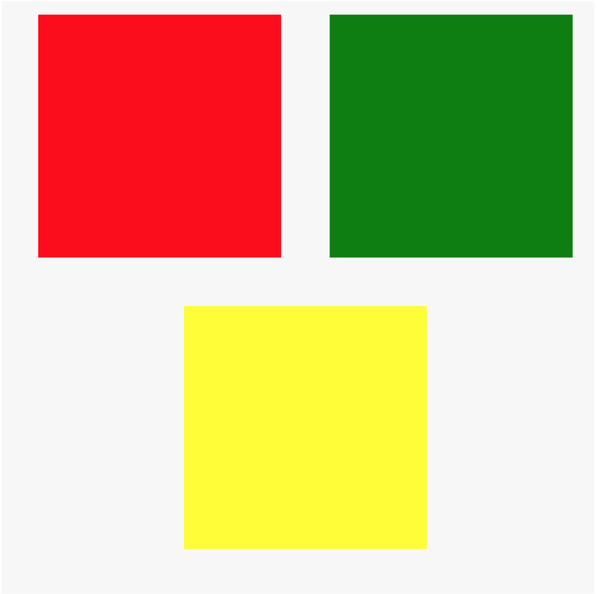
```
.container {
   display: flex;
   flex-wrap: wrap;
   justify-content: center;

   @media screen and (min-width: 640px) {
     justify-content: flex-start;
   }
 }
```

Strategia - spadanie elementów c.d.

Desktop:

Tablet:



Mobile:

Podsumowanie

Jak ćwiczyć układanie stron?

Nic nie jest w stanie zastąpić praktyki przy tworzeniu układów stron internetowych.

Tutaj znajduje się zbiór kilkudziesięciu układów stron internetowych, które w ramach treningu możemy zakodować przy pomocy HTML/CSS.
https://colorlib.com/wp/free-psd-website-templates/