



Web development CSS

Visual Formatting model

Het visual formatting model is meer dan wat we reeds behandelde.

De eigenschap `display` heeft veel meer eigenschappen dan `block`, `inline` en `none`.

We gaan er nog 2 behandelen:

➤ `display:inline-block;`

Hiermee kan je een element (net zoals een inline element) in de regel plaatsen, maar krijg je ook de mogelijkheid om het (net zoals een block-element) een hoogte en breedte mee te geven.

➤ `display:flex;`

Een flexible box layout maakt het veel eenvoudiger om een flexibele responsive layout te maken in vergelijking met verouderde methodes zoals een tabel in HTML of `display:table;`, `display:inline;`, `display:inline-block;`, floating, positioning, ...



Web development CSS

Flex container – Flex items

Wanneer we met een flex layout beginnen moet je weten dat er een onderscheid is tussen

**de flexcontainer
= de parent**

en

**de flex-items
= de children**



- Als we items flexibel t.o.v. elkaar willen plaatsen moet er altijd een container rond staan
- Je hebt eigenschappen voor de flexcontainer en eigenschappen voor de flex-items

Bron: <https://css-tricks.com/snippets/css/a-guide-to-flexbox>



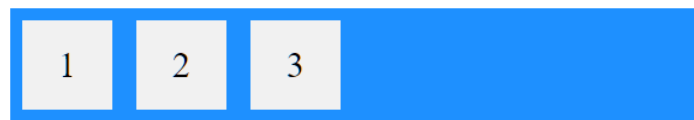
Web development CSS

display: flex;

Het eerste dat we toepassen om de flex-items flexibel te plaatsen, is de eigenschap `display: flex;` plaatsen bij de flex-container:

```
<div class="flex-container">
  <div></div>
  <div></div>
  <div></div>
</div>

.flex-container {
  display: flex;
}
```



Bron: https://www.w3schools.com/css/css3_flexbox.asp

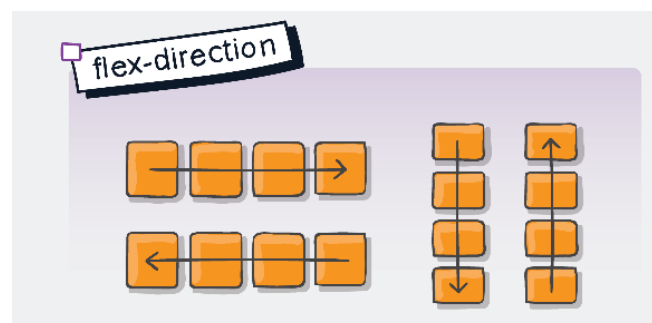


Web development CSS

flex-direction

Standaard komen de items van links naar rechts naast elkaar. Met `flex-direction` kunnen we dit aanpassen:

```
.flex-container {
  flex-direction: row | row-reverse | column | column-reverse;
}
```



Bron: <https://css-tricks.com/snippets/css/a-guide-to-flexbox>



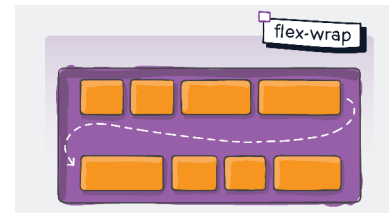
Web development CSS

flex-wrap

Standaard worden alle items op één rij (kolom) geplaatst en gekrompen om op de rij (kolom) te passen.

Met `flex-wrap: wrap;` kunnen de grote ongemoeid laten en laten overlopen naar de volgende regel. Met `flex-wrap: wrap-reverse;` is de volgorde van de rijen (kolommen) omgekeerd.

```
.flex-container {
  flex-wrap: nowrap | wrap | wrap-reverse;
}
```



Bron: <https://css-tricks.com/snippets/css/a-guide-to-flexbox>

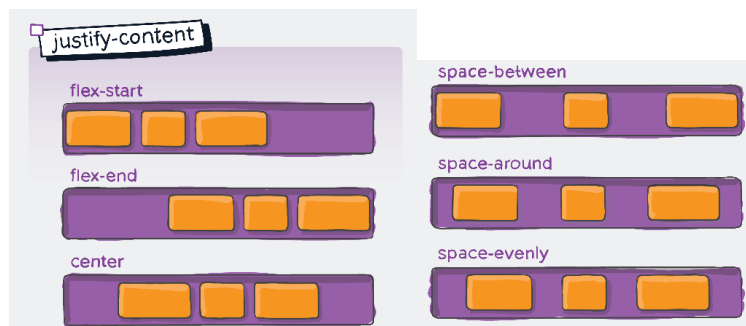


Web development CSS

justify-content

Standaard worden de items links (bovenaan) geplaatst, dit kunnen we aanpassen met `justify-content`.

```
.flex-container {
  justify-content: flex-start | flex-end | center |
  space-between | space-around | space-evenly;
}
```



Bron: <https://css-tricks.com/snippets/css/a-guide-to-flexbox>



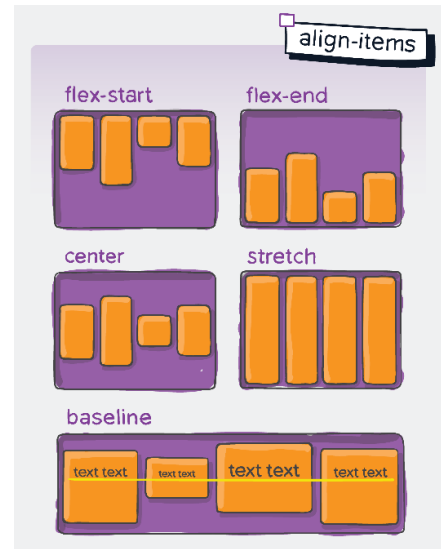
Web development CSS

align-items

Standaard worden items in de hoogte uitgetrokken om allemaal even hoog te zijn (of in de breedte om even breed te zijn bij flex-kolommen). Dit kunnen we aanpassen met `align-items`.

Wanneer we dit aanpassen zal niet alle hoogte van de rij (of breedte van de kolom) benut worden.

```
.flex-container {
  align-items: stretch | flex-start |
  flex-end | center | baseline;
}
```



Bron: <https://css-tricks.com/snippets/css/a-guide-to-flexbox>



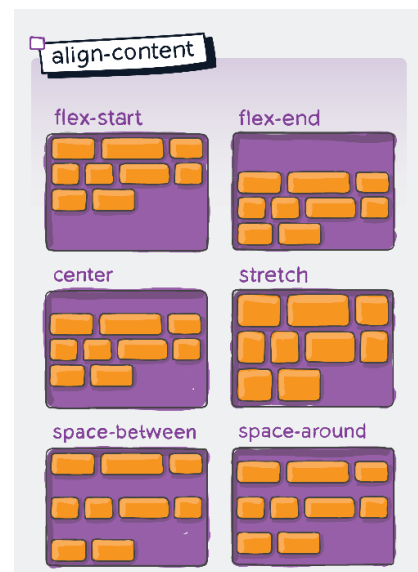
Web development CSS

align-content

Standaard worden de items in de hoogte uitgetrokken (of in de breedte bij flex-kolommen) om binnen de container te passen, dit kunnen we aanpassen met `align-content`.

Dit kunnen we vergelijken met de `justify-content` die werkt volgens de hoofdas, terwijl `align-items` werkt volgens de gekruiste as. Buiten het feit dat `justify-content` default `flex-start` is en `align-content` default `stretch` is.

```
.flex-container {
  align-content: stretch | flex-start |
  flex-end | center | space-between |
  space-around | space-evenly;
}
```



Bron: <https://css-tricks.com/snippets/css/a-guide-to-flexbox>



Web development CSS

eigenschappen voor flex-items

Verder zijn er nog verschillende eigenschappen voor de flex-items: `order`, `flex-shrink`, `flex-grow`, `flex-basis`, `align-self`.

order

```
.item {  
  order: 5; /* default is 0 */  
}
```

flex-shrink, flex-grow, flex-basis

```
.item {  
  flex-shrink: 1; /* default is 1 */  
  flex-grow: 2; /* default is 0 */  
  flex-basis: 50px; /* default auto */  
}
```

align-self

```
.item {  
  align-self: auto | flex-start |  
  flex-end | center | baseline | stretch;  
}
```

Bron: <https://css-tricks.com/snippets/css/a-guide-to-flexbox>