



CSS Grid & Flexbox - Vdemy

2) Tools

Emmet

- 1) !
- 2) nav → `1tab`
- 3) nav → a * 5 `center`
- 4) ctrl+D, `[E]` / `[R]`
- 5) Live Server extension
- 6) link `center` `href="css/styles.css" >`




3) Flexbox ?

- ↳ 1-D layout model, a method that offer space distribution between items
- ↳ A replacement for floats, `display: table`, `display: table-cell`;
- ↳ Specially designed for building
- ↳ Keywords:
 - ↳ Flex Container ^(FC) → 'wrapper'; CSS properties specific to FC
 - ↳ Flex Direction → `flex-direction`: ; `row`, `column`, `-reverse`
default
 - ↳ Flex Items → children of FC
 - ↳ Alignment → `justify-content`: ; `left`, `flex-end`, `center`

4) Flex Container

`display: inline-flex`; // rarely use.

5) Flex Direction

- ↳ `flex-direction`: ; `row`, `column`, `-reverse`
- ↳ Too many element → overflow?
 - ↳ `flex-wrap`: ; `nowrap`, `wrap`, `wrap-reverse`,
ss
- ↳ If container too small, overflow the container?  implicit grid

flex-flow: $\frac{\text{column}}{\text{row}}$ $\frac{\text{nowrap}}{\text{wrap}}$; // short-cut
 ↑ ↑
 Direction Wrap

7) Order → pass to flex items * .same-class: nth-child (1) { }

order: — ; -1, 2, 3. * only apply for flex items within flex container

8) Flex Grow, Flex Shrink, Flex Basis & The Shorthand Flex (For Flex Items)

flex-grow: — ; 1, 2 ≈ fr

flex-shrink: — ; 1, 2, 3 ≈ fr (inverse)

flex-basis: — ; 600px

+ flex-grow: 0; to fix the size to flex-basis.

flex: $\frac{0}{\text{flex-grow}}$ $\frac{0}{\text{flex-shrink}}$ $\frac{600px}{\text{flex-basis}}$;

9) The Flex Property in depth

eg. ~~same~~ flex-items {

flex: 0 0 33.3%; if 50%, 3rd item overflow.

flex-container {
 display: flex;
 flex-wrap: wrap;
 }

flex-items: nth-child (3) {

flex: 0 0 100%;

}

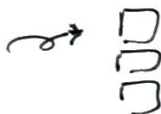


} @media (min-width: 768px) { }

flex-items {

flex: 1 0 100%;

}



10) Flex Alignment with justify-content & align-items

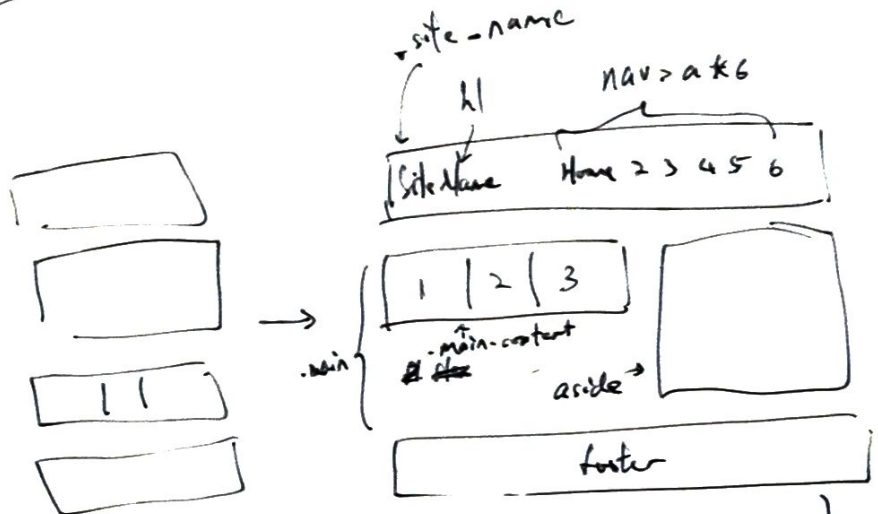
└ justify-content: — ; flex-start, flex-end, center, space-around, space-between
(main-axis)

└ align-items: — ; stretch, flex-start, flex-end, baseline, center
(sub-axis)

To flex-items:

└ align-self: — ;

11) Building a Layout



```
.site-name {
  display: flex;
  flex-direction: column;
}
```

```
@media screen and (min-width: 768px) {
  .site-name {
    flex-direction: row;
    justify-content: space-between;
    align-items: center;
  }
}
```

display: flex;
jc, ai: center;

12) Building a layout (Part 2)

```
.nav {
  display: flex;
  justify-content: center;
}
```

```
nav a {
  margin-right: 1rem; // ≈ column gap
```

```
flex: 1;
display: flex;
flex-wrap: wrap; justify-content: space-between;
```

```
main {
```

```
flex: 0 0 calc(33.33% - 1rem);
```

```
flex: 0 0 calc(66.66% - 1rem); flex: 0 0 40%;
```



```
flex: 0 0 100%;
```



```
display: flex;
flex-direction: column;
align-items: center;
margin-bottom: 1rem;
```

13) Card Project

```
.entry .content {
  background-color: white;
}
```

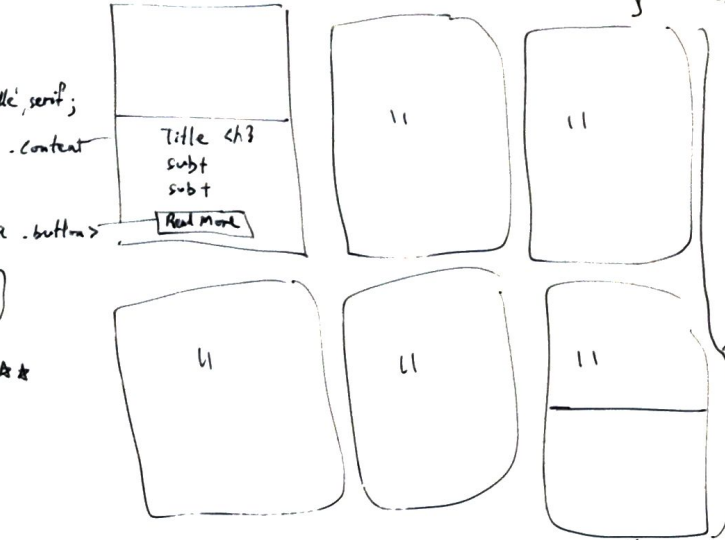
```
h3 {
  margin: 0;
  font-weight: 700;
  font-family: 'Libre Baskerville', serif;
}
```

```
.button {
  background-color: #008000;
  text-decoration: none;
  padding: .6rem 2rem;
}
```

* <a> is inline element,
margin doesn't work

```
margin-top: 1rem;
display: inline-block; ***
color: white
```

latest Entries - h2



```
body {
  background-color: #f2f2f2;
}
```

Box-sizing initial (Paul Irish)

```
html {
  box-sizing: border-box;
}
*, *::before, *::after {
  box-sizing: inherit;
}
```

```
.container {
  max-width: 1200px;
  margin: 0 auto;
}
```

```
h2 {
  text-align: center;
  font-size: 1.6rem;
}
```

```
.main-content {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-between;
  @media (min-width: 480px) {
    .entry {
      flex: 0 0 calc(50% - 1rem);
    }
  }
}
```

```
img {
  max-width: 100%;
}
@media (min-width: 768px) {
  .entry {
    flex: 0 0 calc(33.3% - 1rem);
  }
}
```

15) Project (Freelance web Developer).

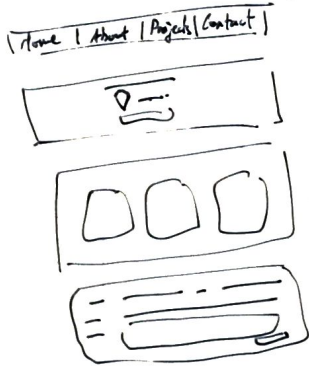
<link fontawesome.com/start -->

<link normalize.css (github) -->

NAV > a * 4

```
<div.hero>
<div.hero-content>
  <h1>
  <p>
  <a>
```

```
.container
  <main>
  <h2>
  <service>
  <h3>
  <icons>
  <p>
```



<link> font-family : Krub (400, 700)

```
:root {
  --mainFont: 'Krub', sans-serif;
  --white: #fff;
  --primary-color: #ff6600;
  --gray: #888888;
  --dark: #222222;
  body {
    font-family: var(--mainFont);
  }
}
```

```
html {
  box-sizing: border-box;
}
*, *:before, *:after {
  box-sizing: inherit;
}
```

→ applying padding/margin without affecting size of the element.

```
1. nav {
  padding: 1rem 0;
  max-width: 1200px;
  margin: 0 auto;
  display: flex;
  flex-direction: column;
  align-items: center;
}
2. @media (min-width: 768px) {
  .nav {
    flex-direction: row;
    justify-content: space-between;
  }
  .nav a {
    margin-bottom: 0;
  }
}
```

```
nav a {
  color: var(--dark);
  text-decoration: none;
  font-size: 1.4rem;
  margin-bottom: 1.5rem;
}
```

* above the text
→ dark overlay effect.

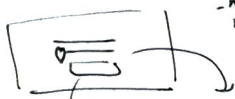
```
2. hero {
  height: 450px;
  background-image: url(../img/hero.jpg);
  position: relative;
  display: flex;
  align-items: center;
}
```

```
.hero:after {
  content: '';
  background-color: rgba(0,0,0,.7);
  display: block;
  top: 0; right: 0;
  left: 0; bottom: 0;
  position: absolute;
}
```

} cover the container

```
<section id="contact".contact>
  <h2> Contact
  <form action="#">
    .form-container
      .field
        <label> Name:
        <input type="text" placeholder="name">
      .field
        <label> Message:
        <textarea name="" id="" cols="30" rows="3">
    </div> <!-- form container -->
    .submit-form
      <input type="submit" class="button" value="send">
```



⑤



- hero
 L hero-content → color: var(--white);
 position: relative;
 z-index: 1; ★ :: dark overlay
 text-align: center;

<a .button> <p><i>m</i></p>

③



service
 " "

- container
 L <h2>
 L (<div .service> x 3) ← div .service-container

.container {
 margin: 0 auto; 2rem auto 3rem auto
 max-width: 1200px;
 padding: 2rem;
 background-color: var(--white);
 box-shadow: 0 4px 12px -1px rgba(0,0,0,0.95);
 = 2 ★ www.cssmatic.com/box-shadow ★ ★
 border-radius: 1rem;
 }

★ html {
 min-height: 100%;
 }

★ body {
 min-height: 100%;
 background: linear-gradient(to top, #dfe9f3 0%, white 100%);

h2, h3 {
 text-align: center;

}
 h3 {
 color: var(--secondary);
 font-weight: 400;
 font-size: 1.6rem;

}
 p {
 line-height: 2;

@media (min-width: 768px) {

.service-container {
 display: flex;
 justify-content: space-between;

}
 .service {
 flex: 0 0 calc(33.3% - 1rem); ≈ 1fr 1fr 1fr.

}
 .service {
 display: flex;
 flex-direction: column;
 align-items: center;

.hero-content {
 font-size: 3rem;
 color: var(--primary);

.button {
 background-color: var(--secondary);
 padding: .5rem 3rem;
 border-radius: 5px;

text-decoration: none;
 margin-top: .5rem;
 font-weight: 700;
 font-size: 1.2rem
 border: none;
 color: var(--white);
 text-transform: uppercase;
 cursor: pointer;

}
 .button:hover {
 background-color: var(--second...);

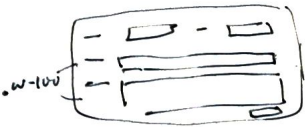
- <h3>
 - <div .icons>
 L <i>
 - <p>

.icons {
 height: 100px;
 width: 100px;
 display: flex;
 background-color: var(--primary);
 border-radius: 50%; ★ circle
 align-items: center;
 justify-content: center;

}
 .icons {
 font-size: 2rem;
 margin-right: 1rem;

④ Contact Form

Contact



```
<section .contact
  <h2
  <form>
    <.form-container
      <.field x4
      <.submit-form
    <input btn -->
```



```
form {
  background-color: var(--gray);
  padding: 2rem;
  border-radius: 1rem;
}
```

@media (min-width: 768px){

```
form {
  max-width: 600px;
  margin: 0 auto;
}
```

```
.form-container {
  display: flex;
  justify-content: space-between;
  flex-wrap: wrap;
}
```

```
.field {
  flex: 00 calc(50% - 1rem);
}
```

}

```
.field {
  display: flex;
  margin-bottom: 1rem;
  align-items: center;
}
```

```
.field input [type="text"],
  " tel,
  " mail,
  " textarea }
```

flex: 1; → to take the available space
height: 2.4rem

```
.w-100 {
  flex: 0 0 100%;
}
```

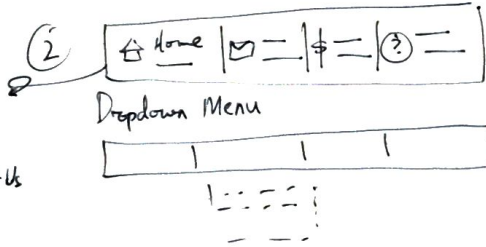
```
.field label {
  flex: 0 0 40px;
  color: var(--white);
}
```

```
.field textarea {
  height: 4rem;
}
```

```
.submit-form {
  display: flex;
  justify-content: flex-end;
}
```

23) Building a ~~Simple~~ Menu

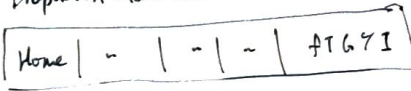
Icons
`nav > ul > (li > a) * 4`
 ↳ `a`
 ↳ `<div.icon>`
 ↳ `<i>`
 ↳ `.text`
 ↳ `Home`
 ↳ `` Learn about Us



Social Menu



Dropdown Menu with Social



sample
`nav > ul > (li * 6 > a) * 6 ?`
 padding: 20px; : hover {}

① - simple ul {
 display: flex;
 justify-content: space-between;
 flex-direction: column;

}
 @media (min-width: 768px) {

 simple ul {
 flex-direction: row;

 }
 simple ul li {
 margin-bottom: 0;

② <Fontawesome .. start using free ... > → →

 icons ul {
 display: flex;
 flex-wrap: wrap;
 justify-content: space-between;

 } .icons ul li {
 flex: 0 0 100%;
 display: flex;
 justify-content: center;
 flex: 0 0 100%;

 } .icons .icon {
 font-size: 2rem;
 flex: 0 0 3rem; 2 gap

 } .icons .text span {
 font-size: .8rem;
 display: block; // flex;
 margin-top: .5rem;

@media (min-width: 480px) {

 icons ul li {
 flex: 0 0 50%;

@media (min-width: 768px) {

 icons ul li {
 flex: 0 0 25%;

(2) dropdown



new_dropdown_out = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100)

dropdown ul {
display: flex;

dropdown ul li {
display: flex;
flex-direction: column;



dropdown ul li ul {
display: none;

dropdown ul li:hover ul {
display: block;
margin-top: 10px;

(width: (min width: 100px)) {
dropdown ul li {
position: relative;
flex: 1 1 100%;

dropdown ul li ul li:hover {
background-color: #f1f3f4;

(width: (min width: 100px)) {

dropdown ul li {
position: relative; flex: 1 1 100%;

dropdown ul li ul {
position: absolute; display: none;
top: 10px; left: 10px; width: 100%; height: 100%;

dropdown ul li:hover ul {
display: flex; flex-wrap: wrap;
background-color: #f1f3f4;
margin-top: 10px;



Social Media

④ Navigation

Social menu

`F L G Y I`

`nav > ul > (li > a) * 5`

* `nav` `ul` `li` `a href="#" > Facebook` * 5

`li` `a href="#" > Facebook` * 5

.social-menu span {
display: none;
}

.social-menu ul {
display: flex;
flex-wrap: wrap;
flex: 1 0 auto;
justify-content: flex-end;
}

.social-menu a::before {
font-family: "Font Awesome 5 Brands";
font-size: 1.2rem;
}

.social-menu a[href*='facebook.com']::before {
content: '\f39e';
width: 50px;
}

.social-menu ul li {
padding: 5px 10px;
}

⑤ Dropdown menu with social media.

combine ③ & ④

.navigation-social-dropdown {
display: flex;
justify-content: space-between;
align-items: center;
background-color: #3f51b5;
}

.navigation-social-dropdown .dropdown ul li {
flex: 1 1 100%;
width: 100%;
}

74) CSS Good

- ↳ 2d
- ↳ \approx table
- ↳ Some thing can be done faster with Flexbox ...

↳ Keywords :

- ↳ Grid Lines
- ↳ Grid Row, Grid Column
- ↳ Grid Track
- ↳ Grid Cell
- ↳ Grid Area

~~25)~~ ★ Firefox Developer Edition

76) Good shorthand & positioning good elements.

grid: rows / columns ; (templates)

grid-column: col-start/col-end (items)

{ grid-row : row-start / row-end (item)

grid-area: row-start/column-start/row-end/col-end (items)

71) grid-auto-flows .row, column, dense

72) grid repeat()

- ↳ span

13) grid-template-areas

↳ fr
 ~ header x 4"
 ~ main x 3 sections
 ~ footer x 4"

grid-gap: — ;

→ grid-area: name ;
header, sidebar, footer.

74) good-template-areas in practice:

75) grid-template-areas with media queries to change the layout order.