

HTML: HyperText Markup Language

HyperText: Text Contains Links (النص الفائق)

Markup: Using Tags (وسوم أو علامات)

وضع علامة أو وسم

لغة توصيف (وسم) النص الفائق

HTML describes the structure of a web page.

HTML consists of a series of elements that tell the browser how to display the content.

<!DOCTYPE html>

يخبر المتصفح إن ده مستند HTML5 حديث

<html lang="en"> Open Tag

Consists of head and body

</html> Close Tag

Root Element الحاوية الأساسية لمحتويات الصفحة

<head>

metadata about the page like:

<title>

عنوان الصفحة في التاب بتاع المتصفح

</title>

</head>

`<body>`

الجسم يحتوي على المحتوى الرئيسي للمستخدم

نصوص، صور، روابط، أزرار

`</body>`

We have only six levels of headings

`<h1>... <h6>`

هيكل هرمي للمحتوى داخل الصفحة

`</h1>... </h6>`

`<h7>Hello</h7>` هيتعامل نص عادي مثل عنوان

`<!--Comments-->`

shortcut: ctrl + /

`<p>` paragraph

lorem10 (VS code generates 10 words paragraph)

`</p>`

`
` Empty Tag (self-closing tag) break

ينزل سطر تحت

نص وهمي لملء الفراغات lorem ipsum

ordered list

```
<ol>
```

```
<li>listItem1</li>
```

```
<li>listItem2</li>
```

```
<li>listItem3</li>
```

```
</ol>
```

unordered list

```
<ul>
```

```
<li>Ali</li>
```

```
<li>Saif</li>
```

```
<li>Mhmd</li>
```

```
</ul>
```

nested list

```
<ol>
```

```
<li>Computers:</li>
```

```
<ul>
```

`HP`

`Dell`

`Asus`

``

``

Images:

- Local images
- Remote images

``

`<!--`

it's a self-closing tag

alt means alternative text

src means source link or location

`-->`

``

``

رابط ضمن نص Hyperlink

اختصاراً Link

href = HyperText Reference

 Press this link

About Page

لينك لصفحة html ثانية

Forms

نموذج إدخال أو استمارة

<form>

<label>Email</label>

<input type="text"> <!--self-closing tag-->

<label>Password</label>

<input type="text">

</form>

يعني عنوان أو بطاقة label

Input types:

حقل نصي عادي type="text"

حقل نصي ولكن يخفي الحروف type="password"

@ مخصص لإدخال بريد إلكتروني type="email"

مخصص للأرقام فقط type="number"

مخصص لأرقام الهواتف type="tel"

<label for="UserEmail">Email</label>

<input type="email" id="UserEmail"

name="Email">

id مطابق ل for لأنه يعرف المتصفح إن حقل الإدخال ده مرتبط بالبطاقة ده فلما تدوس على كلمة Email مربع الإدخال بيتفعل.

name الاسم اللي بيتعرف بيه server على البيانات المدخلة.

أدخلت test@example.com وضغطت submit ما سيصل لل خادم هو: Email=test@example.com

<input type="submit" value="Log In">

إرسال نموذج البيانات للسيرفر أو الجهة المحددة في:

<form action="/login" method="post">

Log In

قيمة تظهر مكان كلمة submit

<input type="button" value="Learn More">

زر يحتاج إلى جافا سكربت لتحديد وظيفته

<button>Download</button> زر عادي

<button type="reset">Reset</button>

زر بيحتوي أن قيمة مكان النص Reset سواء صور أو أيقونات

Container Element

زر داخل الفورم لإعادة تعيين الحقول إلى القيم الأولية

<label>I have read all license agreements</label>

<input type="checkbox" name="agreement"
value="agreed">

value القيمة التي سترسل للخادم إذا تم تحديد المربع

agreement = agreed

<input type="color" />

self-closing tag <... />

<input type="date" />

<input type="file" />

<input type="radio" />

<input type="search" />

Audio & Video

<video controls src="Assets/Drums.mp4"></video>

<audio controls src="Assets/Audio.mp3"></audio>

HTML Semantics: وسوم دلالية تصف المحتوى بداخلها

تقسم الكود إلى هياكل وظيفية

Non-semantic elements like <div> and tells nothing about its content.

HTML Semantics: وظيفتها دلالية وليست شكلية

تعطى معنى للمطور ولمحرك البحث

HTML Formatting

bold text

StrongText

<p>I am Ahmed</p>

<p>I am <i>Ahmed</i></p>

italic = emphasis = خط مائل للتشديد والتركيز

<small>Hello</small>

<p>100\$ - 200\$</p>

<ins>Inserted Text</ins>

<!--نص مدرج تحته خط-->

<p>X₂</p>

<p>X²</p>

<p>Hello in this <mark>course</mark></p>

<p>My Quote: <q>"Don't be afraid"</q></p>

<abbr title="Hyper Text Markup Language">HTML</abbr>

abbr = abbreviation (اختصار)

`<nav> </nav>` (semantic tag)

يحتوي على روابط التنقل بين الأقسام يفهمه محرك البحث والمطور ويسهل إمكانية الوصول.

`<header>Site Logo & Title</header>` (semantic tag)

ده الرأس بتاع ال body.

`<section>قسم لمحتوى مترابط</section>`

`<form>`

`<label>Email</label>`

`<input type="email" placeholder="Email" />`

`</form>`

CSS

CSS: Cascading Style Sheets

أوراق التنسيق المتتالية

CSS is the language we use to style an HTML document.

Best way to connect CSS with HTML:

External Style Sheet: **style.css**

we use the `<link>` tag inside the `<head>` tag of html.

`<link rel="stylesheet" href="style.css">`

Internal Style Sheet

طريقة أخرى لما تكون عايز تنسيق مختلف لصفحة محددة.

We write CSS code inside HTML file

```
<head>
```

```
<style>CSSCode</style>
```

```
</head>
```

Comment in CSS:

```
/*...*/
```

Basic Syntax of CSS:

```
selector {property: value;}
```

selector: المحدد

يعني: عنصر HTML الذي ستطبق عليه الخاصية.

```
<style>
```

```
h1 {
```

```
color: blueviolet;
```

```
}
```

```
p {
```

```
color: rgb(13, 51, 187);
```

```
font-size: larger;
```

```
}
```

```
</style>
```

style.css content example

```
h2 {  
color: chocolate;  
font-style: italic;  
}
```

Link in html file <head>:

```
<link rel="stylesheet" href="style.css" />
```

self-closing tag

rel: relationship نوع العلاقة ملف نمط

href: Hypertext reference مسار الملف

Colors

Color:

- rgb red – green – blue (intensity of each color from 0 to 255) color: rgb(255, 0, 0)
- hex color: #ff0000;
- name color: chocolate;

hexadecimal form: #rrggbb (00:ff)=(0:255)

color: rgba(255, 0, 0, 0.5)

a: alpha = transparency (0.5 = 50%)

HSL: hue – saturation – lightness

(0:360 ColorWheel – 0:100% - 0:100%)

background-color: teal; لون خلفية العنصر

Units

Absolute Length – Relative Length

Absolute: cm – mm – in – px

```
div {
```

```
background-color: red;
```

```
width: 500px;
```

```
height: 500px;
```

```
}
```

Relative:

?: relative to the parent elements

```
section { /* CSS File */
```

```
width: 1000px;
```

```
height: 1000px;
```

```
background-color: aqua;
```

```
}
```

```
div {
```

```
background-color: red;
```

```
width: 100%;
```

```
height: 100%;
```

```
}
```

<section> <!--HTML File-->

<div></div>

</section>

الابن يغطي على الأب يعني div covers section

em = font-size of the parent

font-size: 2em; (ضعف خط الأب)

rem: root em (relative to the root element
<html>)

font-size: 2rem; (<html>خط ضعف خط)

حجم الخط الافتراضي في المتصفح: 16px

rem is the best practice

عشان بتحل مشكلة التراكم أو التضاعف المتكرر

vw: viewport المساحة المرئية من شاشة المتصفح

width: 100vw; (عرض الشاشة 100%)

height: 100vh; (ارتفاع الشاشة 100%)

Typography

font-family: sans-serif;

font-family: "Segoe UI", sans-serif;

اسم الخط يوضع في علامة التنصيص لو أكثر من كلمة بيشوف الأول لو
مش موجود اللان بعده وفي الآخر اسم عائلة الخطوط يشوف خط منها

text-transform: uppercase;

text-transform: lowercase;

text-transform: capitalize; (أول حرف من كل كلمة)

p {

font-family: sans-serif;

font-size: 50px;

line-height: 100px;

letter-spacing: 25px;

word-spacing: 20px;

text-align: center;

text-decoration: line-through;

cursor: pointer;

font-weight: bolder;

font-style: italic;

}

How to import a font from google:

```
@import url("https://fonts.googleapis.com/css2?
family=Momo+Signature&display=swap");
```

```
/* style.css file */
```

```
h1 {
```

```
font-family: "Momo Signature", cursive;
```

```
font-weight: normal;
```

```
font-style: italic;
```

```
}
```

You can create fonts folder inside project then download a font and import it from folder:

```
@font-face {  
font-family: "Impact";  
src: url("fonts/Impact.ttf") format("truetype");  
}  
/* style.css file */  
h1 {  
font-family: Impact;  
font-size: 100px;  
font-style: normal;  
}
```

Box

Everything on the website is made up of a box and the box is made up of four things:

- المحتوى – content
- الهامش الداخلي – padding
- الإطار – border
- الهامش الخارجي – margin

واستكشف inspect

```
div {  
/* border-width: 5px;
```

```
border-style: solid;
border-color: red; */
/* border: 10px dotted teal; */
border-top: 2px solid black;
border-bottom: 2px solid fuchsia;
}
```

```
div {
border: 7px solid red;
/* padding: 50px; */
/* padding-top: 50px;
text-align: right; */
padding: 25px 50px;
/* first value for top and bottom
second value for left and right */
}
```

```
div {
border: 7px solid red;
/* margin-top: 20px;
margin-left: 20px; */
}
```



```
/* margin: 50px; */  
margin: 20px 50px;  
}
```

BACKGROUND

```
* { /* (*{properties}) means select all */  
margin: 0;  
}  
div {  
height: 100vh;  
background-color: aquamarine;  
}
```

```
div {  
height: 100vh;  
background-image: url("");  
background-repeat: no-repeat;  
background-size: cover;  
background-position: top;  
background-attachment: fixed;  
}
```

```
div {  
height: 98vh; /* div content is empty so height: 0  
auto */  
/* background-image: linear-gradient(blue, green);  
*/  
/* background-image: radial-gradient(yellow,  
orange); */  
background-image: linear-gradient(to right, blue,  
green, red);  
background-size: cover;  
}
```

```
div {  
/* div content is empty so height: 0 auto */  
/* background-image: linear-gradient(blue,  
green); */  
/* background-image: radial-gradient(yellow,  
orange); */  
background-image: linear-gradient(  
to bottom,  
rgba(58, 58, 211, 0.4),  
rgba(108, 238, 108, 0.3),  
rgba(229, 66, 66, 0.9)
```

```
),  
url("https://images.unsplash.com/  
photo-1761839257469-96c78a7c2dd3?  
q=80&w=1169&auto=format&fit=crop&ixlib=r  
b-4.1.0&ixid=M3wxMjA3fDF8MHxwaG90by1w  
YWdlfHx8fGVufDB8fHx8fA%3D%3D");  
height: 98vh;  
background-size: cover;  
}
```

```
-----  
filter: blur(7px) brightness(70%) contrast(125%);  
filter: grayscale(90%);  
filter: saturate(50%);  
filter: opacity(75%);  
-----
```

Gradient Website

```
background-image: linear-gradient(to top, #a18cd1 20%,  
#fbc2eb 50%);
```

النسبة بتعبر عن المساحة اللّون بيغطيها من الشاشة
يعني:

1. مرحلة الثبات الأولى (0% - 20%): المتصفح يبدأ باللون البنفسجي (#a18cd1) ويفضل "محتفظ" بيه زي ما هو (Solid Color) لحد ما يوصل لخط ال 20%.

2. مرحلة التدرج (20% - 50%): ديف الـ "Mixing Zone" (منطقة الخلط). هنا المتصفح يبدأ بغير اللون تدريجيًا بنعومة من البنفسجي للوردي في مسافة الـ 30% ديف.

3. مرحلة الثبات الثانية (50% - 100%): بمجرد ما يوصل الـ 50% ويوصل للون الوردي (#fbc2eb) بالكامل، يكمل باقي الشاشة بنفس اللون ده لحد الآخر.

Selectors

We have seen selectors `div {}`, `h1 {}` and `p {}`.

الميزة	id (الرقم القومى)	class (السم العائلة)
التكرار	لازم يكون مميز (واحد بس في الصفحة كلها).	عادي يتكرر (ممك 100 عنصر ياخدوه).
الاستخدام	بيستخدم لعنصر واحد فريد (زي الهيدر الرئيسى) أو عشان الـ JavaScript.	بيستخدم لتجميع عناصر لها نفس الشكل (زي الأزرار).
الرمز في CSS	# (هاش)	. (نقطة)

* select all

```
<body> <!--HTML File-->
```

```
<h1 class="head">Head One</h1>
```

```
<h2 class="head">Head Two</h2>
```

```
<h3 class="head">Head Three</h3>
```

```
</body>
```

```
.head { /* CSS */  
font-size: 2em;  
}
```

```
<!--HTML File-->
```

```
<body>
```

```
<h1 id="one">Head One</h1>
```

```
<h2 id="two">Head Two</h2>
```

```
</body>
```

```
#one {
```

```
/* CSS */
```

```
color: blue;
```

```
}
```

```
#two {
```

```
color: bisque;
```

```
}
```

```
<h1 id="priority">Hello</h1>
```

```
h1 {
```

```
font-size: 5px;
```

```
}  
#priority {  
font-size: 200px;  
color: blueviolet;  
}
```

id له أولوية عن ال class وال class له أولوية عن ال tag العادي:

```
<!--HTML File-->
```

```
<body>
```

```
<main>
```

Main Container

```
<section>Section of related  
content</section>
```

```
<br />
```

```
<article>Independent Article can be moved  
anywhere</article>
```

```
</main>
```

```
</body>
```

```
/* main { /* Everything inside main*/  
border: 5px solid red;
```

```
} */  
main * { /* All tags inside main */  
border: 5px solid brown;  
}
```

```
<!--HTML File-->
```

```
<body>
```

```
<div>
```

```
<p>Hello</p>
```

```
</div>
```

```
<p>World</p>
```

```
</body>
```

```
/*CSS*/
```

```
div p {
```

```
/* only paragraph inside div tag */
```

```
color: crimson;
```

```
}
```

```
p {
```

```
/* Every Paragraph */
```

```
font-style: italic;
```

```
}
```

div p: selects all <p> elements inside <div> elements.

div > p: selects all <p> elements where the parent is a <div> element.

div + p: selects every <p> element immediately after <div> element.

```
<a href="https://www.google.com"
target="_blank" rel="noopener noreferrer">
GoogleLink
</a>
```

لينك يفتح في نافذة جديدة آمنة ويمكن تخصيصه
عن طريق class = " "

```
input[type="email"] {
border: 2px solid orange;
}
```

المحدد الوظيفي: attribute selector
تخصيص بناءً على الوظيفة أو البيانات الداخلية
تخصيص جميع حقول الإيميل

Pseudo Selector

تخصيص العنصر بناءً على حاله أو مكانه

تختار العنصر بناءً على حالته أو موقعه :

تختار جزء معين داخل العنصر ::

الرابط الذي لم يتم النقر عليه بعد link:

الرابط الذي تم زيارته visited:

الرابط عند مرور المؤشر عليه hover:

الرابط لحظة النقر عليه active:

```
<a href="https://www.google.com">google</a>
```

```
<a href="https://www.udemy.com">udemy</a>
```

```
<a href="https://www.wikipedia.com">wikipedia</a>
```

```
<a href="https://www.youtube.com">youtube</a>
```

```
a:link {
```

```
color: burlywood;
```

```
}
```

```
a:visited {
```

```
color: gray;
```

```
}
```

```
a:hover {
```

```
color: blue;
```

```
}
```

```
a:active {
```

```
color: red;
```

```
}
```

<body>

<p>Hello World</p>

</body>

p {

/* تنسيق الفقرة الأصلية */

color: #f14747;

border: 1px solid #615353;

padding: 10px;

}

/* The content property is mandatory with ::before and ::after */

/* 1. ::before (قبل) */

p::before {

content: "Note 💡: ";

font-weight: bold;

color: orange;

}

/* 2. ::after (بعد) */

p::after {

content: " ✓";

color: green;

}

/* 3. ::first-letter (الحرف الأول) */

p::first-letter {

font-size: 2em; /* ضعف حجم الخط */

color: blue;

}

Every HTML element

له خاصية عرض افتراضية تسمى display وقيمتها إما Block or Inline

Block تبدأ دائماً على سطر جديد وتأخذ مساحة عرضها بالكامل.

Inline لا تبدأ سطر جديد بل تجلس بجانب العنصر الآخر في نفس السطر وتأخذ العرض الضروري لمحتواها فقط.

```
<div>Block</div>
```

```
<span>Inline</span>
```

```
<span>Along Side</span>
```

```
div {
```

```
background-color: aqua;
```

```
width: 100px;
```

```
}
```

```
span {
```

```
background-color: blue;
```

```
}
```

عنصر البلوك يفرض نفسه على السطر كامل و يجبر ال span يبدأ من السطر اللان تحته.

لو عنصرين inline ورا بعض في الكود هياجوا بجوار بعضهم.

Inline (like span): any width or height values will have no effect.

Inline-Block: As Inline but you can apply width and height values.

```
span {  
background-color: blue;  
display: inline-block;  
height: 200px;  
}
```

display: none (تخفي العنصر)

Icons

<https://fontawesome.com>

موقع يحتوي على أيقونات عبارة عن خطوط وليست
صور (web fonts)

```
<i class="fa-brands fa-twitter"></i>
```

i: icon (دلالة)

```
<link rel="stylesheet"  
href="https://cdnjs.cloudflare.com/ajax/libs/font-  
awesome/7.0.1/css/all.min.css" integrity="sha512-  
2SwdPD6lNVrV/IHTZbO2nodKhrnDdJK9/kg2XD1r9uGqPo1  
cUbujc+IYdIYdEErWNu69gVcYgdxImVmzTWnetw=="  
crossorigin="anonymous" referrerpolicy="no-referrer" />
```

التاج اللان بيضيف ملف CSS خارجي من قاعدة توزيع محتوى اسمها cdnjs الملف ده بيحتوى على قاعدة الخط بتاع fontawesome عشان يتم عرض الأيقون بشكل صحيح.

fontawesome makes display of icons inline-block

Shadows

```
div {  
background-color: #35024d;  
width: 500px;  
height: 500px;  
box-shadow: 10px 5px 20px orange; /* x y  
blurriness color */  
}
```

```
h1 {  
font-size: 200px;  
text-align: center;  
font-family: Impact;  
letter-spacing: 10px;  
color: orangered;  
text-shadow: 10px 10px 10px rgb(241, 203,  
132);  
}
```

Inheritance

بشكل عام، الخصائص التي تؤثر على التيبوغرافيا (Typography) والنص هي التي تُورث تلقائيًا، أما الخصائص التي تحدد أبعاد العنصر ومساحته وموقعه فهي لا تُورث.

layout related properties will not be inherited.

```
<body>
```

```
<div class="parent">
```

Hello

```
<div class="child">World</div>
```

```
</div>
```

```
</body>
```

```
.parent {
```

```
color: aquamarine;
```

```
font-family: sans-serif;
```

```
font-size: 50px;
```

```
font-weight: bolder;
```

```
font-style: italic;
```

```
padding: 100px; /* not inherited and below */
```

```
width: 20px;
```

```
height: 20px;  
background-color: blue;  
}  
.child {  
padding: 20px;  
}
```

```
.parent {  
font-size: 100px;  
width: 300px;  
height: 100px;  
background-color: blue;  
}
```

الابن هياخذ نفس الاتساع بتاع الأب بصفته بلوك موجود داخله وليس بالوراثة

width = auto (parent)
height is not the same

inherit keyword

للخصائص التي لا تورث أوتوماتيك

```
.child {  
width: inherit;  
height: inherit;  
background-color: inherit;  
}
```

Position

position: static (Normal Flow)

الوضع الثابت الافتراضي: العنصر يتبع التسلسل الافتراضي للمستند.

top – left – right – bottom

لن يكون لها أي تأثير

position: relative (نسبي)

مثل سابقه ولكن يقبل قيم:

top – left – right – bottom

بالنسبة لموضعه الأصلي.

position: absolute (مطلق / حر)

يتم وضعه بالنسبة لأقرب أب له يكون له خاصية غير static ولو مفيش أب يبقى بالنسبة ل <html>

position: fixed (ثابت)

يتم وضعه بالنسبة لمنفذ العرض viewport ويظل ثابتاً في مكانه حتى لو المستخدم عمل scroll

معلومة: العنصر body دائماً يأخذ اتساع width الشاشة بالكامل أما بالنسبة للارتفاع فهو يأخذ فقط ما يكفي height المحتوى داخله.

ملحوظة مهمة: position = absolute or fixed
بتخرج العنصر من التسلسل الافتراضي للمستند
يعني العنصر يطفو واللي تحته يياخذ مكانه.

لاصق Position: sticky

يتصرف ك relative لحد ما يوصل لنقطة معينة أنت
محددها ويقلب fixed

```
<body>  
<div class="container">  
<div class="d one"></div>  
<div class="d two"></div>  
<div class="d three"></div>  
</div>  
</body>
```

```
.container {  
width: 900px;  
height: 800px;  
background-color: black;
```

```
}  
.d {  
width: 200px;  
height: 200px;  
}  
.one {  
background-color: brown;  
position: relative;  
left: 100px;  
}  
.two {  
background-color: blue;  
position: absolute;  
top: 500px;  
}  
.three {  
background-color: burlywood;  
/* position: fixed; */  
top: 0;  
position: sticky;  
}
```

sticky في البداية يتصرف ك relative ويحل محل
الذي فوقه وبعد كذا مع scroll عند $top = 0$ يعني
لما العنصر يلتصق بأعلى الشاشة يتحول ل fixed
ويثبت في مكانه.

Flexbox

نظام تخطيط أحادي البعد يستخدم لترتيب العناصر
في صف واحد أو عمود واحد.

Main Axis: اتجاه العناصر سواء صف أو عمود

Cross Axis: المحور المتقاطع ده المحور العمودي
على المحور الرئيس

تفعلها للعنصر الأب عشان ترتب جواه `display: flex;`

`flex-direction: Main Axis Direction`

`div>.child*8` (Emmet Abbreviation)

Emmet: built-in plugin in **vs code** enables us
to write long html code with an abbreviation

ده يعمل عنصر `div` بداخله 8 أبناء `class="child"`

`margin: 100px auto;`

بيخلي الهوامش الخارجية 100 من فوق وتحت ويمين ويسار بيخصم عرض
العنصر من عرض الصفحة ويقسم النتيجة على 2 عشان يوزع يمينا ويسار
بالتساوي بشكل ديناميكي فيعمل توسيط إنما لو كتبت قيمة ثابتة
بيوزعها في الأربع نواحي بدون توسيط.

`flex-direction: row-reverse;`

هو property خاصة بالأب وله أربع قيم:

row | row-reverse | column | column-reverse

justify-content:

property خاصة بالأب وتستخدم لمحاذاة العناصر على المحور الرئيسي.

Values: center | end | space-around

space-between | space-evenly

flex-wrap: nowrap; or wrap or wrap-reverse

خاصية للأب تحدد سلوك الأبناء عند اكتمال المساحة

nowrap: تبقى جميع العناصر على سطح واحد حتى لو اضطرت للانكماش أو تجاوز الحدود

wrap: تنتقل العناصر المتبقية إلى سطر جديد بالترتيب الطبيعي

wrap-reverse: عكس اتجاه المحور المتقاطع

يعني المحور المتقاطع بدل ما كان اتجاهه من الأعلى للأسفل وكل ما تضيف قيمة جديدة تنزل تحت أصبح اتجاهه من الأسفل للأعلى وكل ما تضيف قيمة جديدة بتطلع لفوق

align-items:

محاذاة العناصر على المحور المتقاطع

Values: start | end | center

align-items: center

justify-content: center

يخلين الأبناء في مركز الحاوية بالضبط

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-
width, initial-scale=1.0" />
<title>صفحتي</title>
<link rel="stylesheet" href="style.css" />
</head>
<body>
<div class="container">
<div class="child">1</div>
<div class="child">2</div>
<div class="child">3</div>
<div class="child">4</div>
<div class="child">5</div>
</div>
</body>
</html>
```

```
* {  
margin: 0;  
padding: 0;  
}  
  
body {  
height: 100vh;  
background-color: violet;  
}  
  
.container {  
border: 10px solid red;  
width: 600px;  
height: 600px;  
margin: 100px auto;  
display: flex;  
/* flex-direction: row;  
flex-wrap: nowrap; */  
justify-content: end;  
align-items: end;  
}  
  
.container .child {  
width: 50px;  
height: 50px;
```

```
background-color: cadetblue;
margin: 7px;
text-align: center;
}
```

align-content:

خاصية تتحكم في توزيع الصفوف lines يعني الخطوط كاملة على المحور المتقاطع وبالتالي لازم يكون عندك أكثر من صف يعني:

flex-wrap: wrap (متنفعش nowrap)

Children Properties:

```
.my-item {
flex: 1 0 200px; /* flex-grow: 1, flex-shrink: 0,
flex-basis: 200px */
}
```

معامل الانكماش shrink معامل النمو grow
الحجم الأولي basis

order ترتيبه بغض النظر عن ترتيبه في كود ال html
order default value is zero

ويبدأ من الصفر

ع المحور المتقاطع align-self: start | end | center

الخصائص التي تبدأ ب align تعمل على المحور المتقاطع

يتسمح لعنصر الابن يلغى قواعد المحاذاة بتاع الأب ويحدد محاذاة لنفسه.

شرح grow

تخيل عندك حاوية عرضها 1000px وفيها 3 عناصر:

العنصر	flex-basis (الحجم الأولي)	flex-grow	حصة النمو (من المساحة الفارغة)
A	100px	1	1 / 6
B	100px	2	2 / 6
C	100px	3	3 / 6

النتيجة:

$$A = 700px * 0.6$$

$$B = 700px * (2 / 6)$$

$$C = 700px * (3 / 6)$$

$$A \text{ (final width)} = 100px + 116.7px$$

flex-grow default value is 0 while flex-shrink default value is 1

عشان دائمًا العنصر ينكمش ويأخذ جزء من الحاوية وميخرجش برا بدل ما يأخذ الحجم الأصلي.

<https://flexboxfroggy.com>