



CSS

Written By: Ibrahim Abdelaty

الكاتب : ابراهيم عبد العاطي

حسابي علي تويتر: <https://twitter.com/IbrahimElaty>

الفهرس

(الصفحة)	الموضوع
(3).....	قواعد كتابة كود CSS
(3).....	Selectors
(5).....	Variables
(5).....	Global values
(6).....	وحدات القياس في الـ CSS
(6).....	CSS Properties
(6).....	Background
(7).....	Element sizing
(8).....	Fonts & Text
(11).....	Filters
(10).....	Gradient
(10).....	Caret & Cursor
(10).....	Lists
(11).....	Position
(11).....	Float
(12).....	Flex box
(14).....	Grid
(15).....	Transform (2D)
(16).....	Transform (3D)
(17).....	Transition
(17).....	Animation
(18).....	Media queries

قواعد كتابة كود CSS:

```
Selector {  
    Property1: value ;  
    Property2: value  
}
```

لاحظ: المحدد (selector) قد يكون (class/ id) identifier او اسم عنصر ما.

Selectors

1. * // لتحديد كل العناصر
2. element // لتحديد عنصر معين
3. .class // لتحديد عنصر معين من الـ class
4. #id // لتحديد عنصر معين من الـ id
5. div p // لتحديد اي عنصر P داخل اي عنصر div ... بغض النظر عن العناصر الاخرى في الـ div
6. .parent .child // نفس رقم 5
7. .class1.class2 // للبحث عن عنصر فيه "class1 class2" class="
8. .parent .child , div p // كذا او كذا (محددان)
9. p.any-class // لتحديد عنصر p وله الـ class السابق..
10. .parent > .child // هنا يجب ان يكون الابن المباشر للاب.. وليس داخل عنصر داخل اخر
11. div + p // لتحديد اي عنصر P قبله عنصر div ومعه في نفس الاب..
12. div ~ p // لتحديد اي عناصر P قبلها عنصر div ومعه في نفس الاب..
13. [Attribute] // لتحديد عناصر او عناصر عن طريق اسم سمة معينة لها
14. [Attribute = value] // لتحديد عناصر او عناصر عن طريق اسم سمة وقيمتها
15. div [title] // لتحديد عنصر div عن طريق اسم السمة title..
16. [title ~= "word"] // لتحديد عناصر او عناصر بها سمة قيمتها تحتوي علي كلمة word..

17. [title *= "word"] // لتحديد عناصر او عناصر بها سمة قيمتها تحتوي علي كلمة او جزء من كلمة..
18. [title ^= "word"] // لتحديد عناصر او عناصر بها سمة قيمتها تبدأ بكلمة او جزء من كلمة..
19. P:first-child // لتحديد اي عنصر P يكون العنصر الاول في الاب..
20. P:last-child // لتحديد اي عنصر P يكون العنصر الاخير في الاب..
21. P:first-of-type // لتحديد اي عنصر P يكون العنصر الاول من نوعه في الاب..
22. P: last-of-type // لتحديد اي عنصر P يكون العنصر الاخير من نوعه في الاب..
23. P:only-child // لتحديد اي عنصر P يكون العنصر الوحيد في الاب..
24. P: only-of-type // لتحديد اي عنصر P يكون العنصر الوحيد من نوعه في الاب..
25. P:not(:first-child) // لتحديد اي عنصر p ماعدا الاول ...
26. P:nth-child(2) // لتحديد اي عنصر P يكون العنصر الثاني في الاب..
27. P:nth- child(even or odd) // لتحديد اي عنصر P يكون عنصر رقمه زوجي او فردي..
28. P:nth-last-child(2) // لتحديد اي عنصر P يكون العنصر الثاني من الخلف في الاب..
29. P:nth-of-type(2) // لتحديد اي عنصر P يكون العنصر الثاني من نوعه في الاب..
30. P:nth-child(n+2) // لتحديد اي عنصر P يكون العنصر الثاني وما بعده في الاب..
31. P:root // لتحديد اي عنصر P يكون موجود في عنصر الـ html..
32. checked: // لتحديد اي عنصر في السمة checked مثل عنصر الـ input
33. empty: // لتحديد اي عنصر فارغ
34. disabled: // لتحديد عنصر ادخال معين له السمة disabled..
35. required: // لتحديد عنصر ادخال معين له السمة required..
36. focus: // لتحديد عنصر ادخال معين له السمة focus.. (عندما يكون فوكس .. او مهياً للادخال)
37. a :hover // لتحديد عنصر (عند الوقوف عليه بال مؤشر)
38. a :visited // لتحديد عنصر اللينك (الذي تم زيارته)
39. div ::selection // لتحديد الـ selection للعنصر ..div

40. `div::placeholder` // لتحديد الـ placeholder للعنصر `div`.
41. `div::first-letter` // لتحديد أول حرف نصي في عنصر `div`.
42. `div::first-line` // لتحديد أول سطر نصي في عنصر `div`.
43. `div::before {content:" ";}` // لإنشاء عنصر في البداية داخل عنصر آخر. لابد ان يحتوي علي الخاصية `content` وقد تحتوي علي نص او `لا`.
44. `div::after {content:" ";}` // لإنشاء عنصر في النهاية داخل عنصر آخر. لابد ان يحتوي علي الخاصية `content` وقد تحتوي علي شيء او لا .

لاحظ: قيمة الخاصية `content` قد تكون فارغة "" او سلسلة نصية "Ibrahim" او `counter(counter name)` لاضافة عداد علي مجموعة من العناصر الموجودة داخل عنصر اب .. اسم العداد يكون قيمة تعيينها للخاصية `counter-increment : counter name` التي توضع في العنصر الاب.

او `attr(any html attribute)`

Variables

- تعيين المتغيرات:
اسماء المتغيرات تبدأ بـ `--` متبوعة باسم تعيينه انت مثل `--maincolor : red;`
(1) المتغيرات العامة: تكون داخل `{--maincolor : red; :root}`
(2) المتغيرات الخاصة: تعين لعنصر معين `p {--maincolor : black; }`
- استدعاء المتغيرات: نستدعي قيمة المتغير من الدالة `var()` مثل `div {color:var(--maincolor);}`

Global values

- هي مجموعة من القيم العامة التي يمكن تعيينها لاي خاصية..
- `inherit` // تستخدم لجعل العنصر يرث قيمة الخاصية من الاب..
 - `initial` // تستخدم لجعل قيمة الخاصية هي القيمة الافتراضية لها.. default value
 - `unset` // تعمل عمل الـ `inherit` اذا كان يمكن توريث القيمة.. و `initial` اذا كان لا يمكن توريثها..

• **revert** // تستخدم لجعل قيمة الخاصية هي القيمة التي يعطيها لها المتصفح .. user agent

لاحظ: يمكن التحكم في خواص العنصر كلها عن طريق: **all : inherit or ..**

وحدات القياس في الـ CSS

- **1em** = 100% من حجم العنصر
- **1rem** = 100% من حجم الـ root (html element).
- **1vw** = 1% من عرض الصفحة. // view port width
- **1vh** = 1% من ارتفاع الصفحة. // view port height

** CSS Properties **

Background

- **background-color** : red
- **background-image** : url(“ URL for the image ”)
لاحظ: تستخدم (//) للرجوع خطوة الي الوراء من المكان الحالي في عنوان URL.
- **background-repeat** : repeat or no-repeat or repeat-x or repeat-y
 - **repeat-x** : لتكرار صورة الخلفية في الاتجاه الافقي.
 - **repeat-y** : لتكرار صورة الخلفية في الاتجاه الرأسي.
 - **repeat** : لتكرار صورة الخلفية في الاتجاهين الافقي والرأسي.
 - **no-repeat** : عدم تكرار صورة الخلفية.
- **background-attachment** : scroll or fixed
 - **scroll** : عند وجود صورة خلفية لعنصر كبير وعند الاسكروول لاسفل او اعلي تتحرك الصورة مع الاسكروول
 - **fixed** : لا تتحرك الصورة مع الاسكروول وتبقى ثابتة بالنسبة لموقعها في المتصفح.

- **background-position** : left top or left center or left bottom
center top or center center or center bottom
right top or right center or right bottom
لاحظ: يمكن استخدام البكسل والنسبة المئوية لتحريك صورة الخلفية 50px 20px او 20% 40% .

- **background-size** : auto or contain or cover
cover : تغطي الصورة كامل الطول والعرض حتي مع التغير فيهما
contain : تغطي الصور كامل العرض والطول يغطي منه علي حسب ابعاد الصورة.
لاحظ: يمكن استخدام البكسل والنسبة المئوية لتغيير ابعاد صورة الخلفية 100px 200px او 20% 40% .

Element sizing

- // الهوامش الداخلية. **padding** : 5px(top) 2px(right) 9px(bottom) 18px(left)

padding : 0px 2px 0px 0px

padding : 15px (15px 15px 15px 15px)

padding : 15px 12px (15px 12px 15px 12px)

padding : 51px 20px 18px 20px (51px 20px 18px 20px)

- **لاحظ:** يمكن استخدام **padding-top** : 10px و **padding-left** : 100px
ال **padding** لا تقبل قيم سالبة.

- // الهوامش الخارجية. **margin**: 5px(top) 2px(right) 9px(bottom) 18px(left)

- **لاحظ:** - يمكن استخدام **margin-top** : 10px و **margin-left** : 100px
- ال **margin** تقبل القيم سالبة.

- يمكن تعيين القيمة **auto** لترك القرار للمتصفح فيحاول جعل العنصر في المنتصف بجعل نصف
- الطول الفارغ للـ **margin-left** و النصف للـ **margin-right** .

- // الهوامش **border** : 3px (border-width) solid(border-style) red(border-color)

لاحظ:- الترتيب السابق ضروري.

- يمكن استخدام **border-width** و **border-style** و **border-color** ويمكن ان تأخذ كل خاصية من قيمة لـ اربعة قيم (top right bottom left).

- **outline** : none // لازالة الـ outline هي هوامش خارج الـ border

- **display : inline** لتحويل العنصر الي عنصر inline (حيث توضع العناصر بجوار بعضها)
- **display : block** لتحويل العنصر الي عنصر block (حيث توضع العناصر اسفل بعضها)
- **display : inline-block** يأخذ خواص عناصر الكتلة ، باستثناء ان العناصر توضع بجانب بعضها اذا كن العرض يكفيها.
- **display : none** // اخفاء العنصر وازالة مكانه
- **visibility : hidden** // اخفاء العنصر وابقاء مكانة فارغ
- **min-width : 100px or max-width : 100px or width : 100px** لاحظ : يمكن تعيين القيمة **fit-content** لل **width** لجعل عرض العنصر فقط عرض المحتوي.
- **min-height : 100px or max-height : 100px or height : 100px**
- **border-radius : (top) (right) (bottom) (left)** // لاضافة حواف دائرية للعنصر.
border-top-left-radius :5px
- **box-shadow : (H shadow) (V shadow) (blur) (spread) (color) (inset or outset)** لاضافة الظل حول العنصر
- **box-sizing : content-box or border-box** لاحظ: اذا كانت قيمة **content-box** الخاصة الهوامش تكون خارج العنصر .. اما اذا كانت قيمة **border-box** الخاصة الهوامش تكون داخل العنصر
- في العناصر المرصوصة فوق بعضها يتم دمج الـ **vertical margin** اي يحسب وحد فقط منهما
\\ margin collapse

Fonts & Texts

لاستيراد خط محلي في الـ **CSS**: يضاف واحدة لكل وزن لكل خط....

```
@font-face {
    Font-family : "font name" ;
    font-weight: 100 or 200;
    font-style: normal;
    Src : url( "font-name.woff" );
}
```

- `color : green` // لون النص text color
- لاحظ: `transparent` تعني لون شفاف.
- `font-size : 15px` // حجم النص
- `font-family : Arial , Helvetica , ...`
- `font-style : italic or normal`
- `font-variant : small-caps` // لتحويل الحروف الي كابيتل ولكن بحجم اصغر
- `font-weight: bold` // لجعل النص سميك.
- `text-align : left or center or right`
- `vertical-align : top or middle or bottom`
- `text-decoration : none or underline or line-through or overline`
- `text-transform : capitalize or uppercase or ...`
- `text-indent : 50px` // المسافة الفارغة في بداية النص .
- `letter-spacing : 2px` // المسافة بين الحروف , يمكن ان تكون القيمة سالبة.
- `word-spacing : 4px` // المسافة بين الكلمات.
- `line-height : 160%` // ارتفاع السطر النصي .
- `direction : rtl or ltr` // اتجاه كتابة النص
- `white-space : nowrap` // لا تسمح للنص بالنزول للسطر التالي حتي وان كان النص اكبر من عرض العنصر.
- `word-break : break-all` // لجعل الكلمة تكمل في السطر التالي اذا كان عرضها اكبر من عرض العنصر.
- `overflow-y or overflow-x or overflow :`
تأخذ عدة قيم `hidden` لازالة النص الخارج عن ابعاد العنصر
- `scroll` لاختفاء النص الخارج عن ابعاد العنصر مع امكانية الاسكروول
- `auto` نفس الاسكروول مع عدم اضافة الشريط في حال كان النص لم يخرج عن العنصر.
- `visible` اظهر النص حتي وان خرج عن ابعاد العنصر

- **text-overflow : ellipsis** // لوضع 3 نقط في نهاية النص تدل علي ان هناك جزء مقطوع من النص
- **لاحظ:-** يمكن استخدام الدالة $(100\% - 50px) / 6$ **calc** لتعيين قيمة لخاصية مثل الطول او العرض.. عن طريق وضع عملية حسابية داخلها.

Filters

- **filter: grayscale(70%) or blur(4px) or invert(20%) or..**
 - grayscale** لجعل الصورة ابيض واسود
 - blur** لجعل الصورة ضبابية.
 - invert** لعكس الون الصورة.
- **لاحظ:** ليس كل المتصفحات تدعم هذه الخاصية ... للتأكد من ذلك نستخدم موقع `can I use` .

Gradient

- لانشاء تدرج لوني في عنصر مثل div نعين له الخاصية التالية
- **background-image : linear-gradient((direction or angle), (color stop), (color stop),...)**
 - **background-image : linear-gradient(to right or 90deg , red 20% ,green 40% ,...)**

Caret & Cursor

- **cursor: pointer or move ...** // لتغيير شكل المؤشر عن الوقوف علي عنصر ما.
- **caret-color : red;** caret هو مؤشر الكتابة ، يمكن تغيير لونه من الخاصية
- عند تعيين القيمة none للخاصية **pointer-events : none;** لاحد العناصر مثل الـ a فإن العنصر لن ينتقل الي اللينك و لن يكون هناك `..hover`.

Lists

- **list-style : type position image** // لتنسيق القوائم
 - type** <= نوع النقاط .
 - position** <= موضع نقاط القائمة داخل او خارج العنصر
 - image** <= قيمتها تكون ("URL for image") **url**.. صورة بدل من نقاط القائمة.
- **لاحظ:** الاستخدام الاساسي للقائمة `` يكون لعمل `navigation par`.

Position

- **position:** // تستخدم هذه الخاصية لتحديد مكان العنصر.. ولها 4 قيم. الخاصية تعمل بجانب الخواص الاربعة **top** و **bottom** و **right** و **left** .
 - = **relative** لتحرك العنصر بالنسبة لنفسه.
 - = **absolute** لتحرك العنصر بالنسبة للاب الذي يحويه ... يجب ان تكون قيمة الخاصية **position** للاب هي **relative** .
 - = **fixed** لتحرك العنصر بالنسبة للصفحة... ولا يتغير موضعه بالاسكرول
 - = **sticky** يكون العنصر في مكانه الطبيعي. ويتوقف عن الاسكرول عند موضع تحدده من **top** و ..
- **z-index: 2 or..** // لتحديد الترتيب الراسي للعناصر (لا تعمل الخاصية بدون الـ **position**) العنصر صاحب القيمة الاكبر لهذه الخاصية يظهر فوق الاقل ... يمكن ان تأخذ الخاصية القيم السالبة.

لاحظ: - احيانا يكون هناك تنسيق داخلي للعنصر وتنسيق في ملف الـ **CSS** وما يتم تطبيقه هو التنسيق الداخلي. ولتعيين خاصية لعنصر تطبق علي العنصر حتي وان كانت هناك خواص داخلية تقوم بنفس الوظيفة نكتب بعد قيمة الخاصية **!important** .
`\\important declaration`

Float

- **float: left or right** // توضع لعناصر الموجودة داخل عنصر جامع لجعل العناصر تطفو بجانب بعضها **لاحظ:** لجعل العناصر تنتظم داخل عنصر جامع لها . نجعل اخر عنصر هو عنصر مثل **<p>** فارغ ونعين له خاصية **clear : both or right or left** .. او لجعل العناصر علي اكثر من سطر.

Flex box

واحدة من افضل الخواص في اللغة .. تستخدم كبديل افضل من الـ **float** .. تستخدم لتنسيق عدة عناصر موجودة داخل عنصر اب يجمعها. لتفعيل الخاصية نكتب **display : flex or inline-flex** للعنصر الاب .

خواص الـ Flex box :

1. **خواص توضع في العنصر الاب:** // توضع مع الخاصية السابقة في العنصر الاب.

- **flex-direction : row or row-reverse or column or column-reverse** لترتب العناصر داخل الصفحة في صف في اتجاه الكتابة.
row

row-reverse لترتب العناصر داخل الصفحة في صف في عكس اتجاه الكتابة.
column لترتب العناصر داخل الصفحة في عمود مرتب من اسفل لاعلي.
column-reverse لترتب العناصر داخل الصفحة في عمود معكوس.

- **flex-wrap** : **wrap** or **nowrap** or **wrap-reverse**
wrap لجعل العناصر تنزل للسطر التالي اذا كان العرض لا يكفي .
nowrap لا تجعل العناصر تنزل للسطر التالي حتي اذا كان العرض لا يكفي فتغير عرض العناصر ليتناسب مع عرض الاب.
wrap-reverse تجعل العناصر الزائدة عن السطر تصعد للسطر الاعلي وليس للاسفل

لاحظ: هناك خاصية تجمع الخاصيتين السابقتين وهي **flex-flow** : **row wrap**;

- **justify-content** : **flex-start** or **flex-end** or **center** or **space-between**
flex-start لجعل العناصر في الجانب الايسر (اللغة E) من الصفحة.
flex-end لجعل العناصر في الجانب الايمن (اللغة E) من الصفحة.
center لجعل العناصر في المنتصف (اللغة E) من الصفحة.
space-between لتوزيع المسافة الفائضة من العرض.. كمسافات بين العناصر.
space-around لتوزيع المسافة الفائضة من العرض.. كمسافات علي يمين ويسار العناصر.
space-evenly هي حاصل دمج الخاصيتين السابقتين.

- **align-items** : **stretch** or **flex-start** or **flex-end** or **center**
stretch لجعل العناصر الموجودة داخل الاب تأخذ ارتفاعه .. واذا زاد ارتفاع عنصر يزداد ارتفاع الاب وبالتالي تزداد باقي العناصر والموجودة داخله الا اذا حددت ارتفاع العنصر .
flex-start لجعل العناصر الموجودة داخل الاب تكون في لاعلي رأسيا
flex-end لجعل العناصر الموجودة داخل الاب تكون في لاسفل رأسيا
center لجعل العناصر الموجودة داخل الاب تكون في الوسط رأسيا
لاحظ: يمكن استخدام الـ **Flex box** لتنسيق النص الموجود داخل عنصر مثل الـ **<div>** بنفس الخواص السابقة.

- **align-content** : **stretch** or **flex-start** or **flex-end** or **center** or..
يعمل علي التنسيق الرأسي للعناصر الموجودة في الاب في حالة انها كانت اكثر من صف ..
و يأخذ نفس القيم الموجودة في **justify-content** و **align-items** .
لاحظ: **align-items** يقوم بنفس الوظيفة وهي تنسيق العناصر رأسيا في حالة كنت صف واحد.

2. خواص توضع في العناصر الموجودة داخل الاب:

• Flex-grow: 0 or 1 or 2 or..

اذا وضعت قيمة الخاصية لعنصر اكبر من الاخر في العنصر صاحب القيمة الاكبر ينمو اكبر اذا وجدت مسافة.

• Flex-shrink: 1 or 2 or 3

عكس الخاصية السابقة.

• order: 0 or 1 or 2 or..

تحدد ترتيب العناصر داخل الاب فصاحب القيمة الاقل يأتي في الاول ..للاستخدام السليم لهذه الخاصية يفضل تعيينها لكل العناصر داخل الاب وليس لاحدها فقط.

• Flex-basis: 55px

عند تحويل الصف الي عمود عن طريق **flex-direction : column** فإن العرض للعناصر يظل عرض وبالتالي لن يتحول الصف الي عمود بشكل صحيح .. في حالة ارت ان يتحول العرض الي طول عند التحويل من صف لعمود .. تكتب العرض للعناصر بهذه الخاصية وليس بالـ width .

• Flex : (Flex-grow) (Flex-shrink) (Flex-basis)

• align-self : stretch or flex-start or flex-end or center

تقوم بما تقوم به الخاصية align-items والفرق ان هذه الخاصية تحدد لعنصر موجود داخل الاب للتحكم فيه كمستقل.

Grid

واحدة من افضل الخواص في اللغة .. تستخدم لتوزيع العناصر داخل الصفحة بكل اريحية اعتمادا علي تقسيم الصفحة لعدة اعمدة ...تستخدم لتنسيق عدة عناصر موجودة داخل عنصر اب يجمعها. لتفعيل الخاصية نكتب **display : grid** للعنصر الاب .

خواص الـ grid:

(1) خواص توضع في العنصر الاب: // توضع مع الخاصية السابقة في العنصر الاب.

• **grid-template-columns : 100px 100px 50px** // لإنشاء عدد من الاعمدة بعرض معين
or 25% 50% 5% ...
or auto auto auto ...

100px // لإنشاء ثلاث اعمدة عرض الواحد 100px or repeat(3 , 100px) ...
50 // لإنشاء ثلاث اعمدة عرضها 50 والباقي من عرض الاب يوزع علي عمودين. or 50px 1fr 1fr ...

لاحظ: الفرق بين auto و fraction ان الاول خجول والثاني طماع...

- **grid-template-rows** : 100px 100px 50px // لإنشاء عدد من الصفوف بارتفاع معين
or 25% 50% 5% ...
or auto auto auto ...
// لإنشاء ثلاث صفوف ارتفاع الواحد ثلث الارتفاع الكلي or repeat(3 , 1fr)
or 50px 30% auto ...
// لتحديد اقل واكبر لارتفاع للصف or minmax(100px , 150px)
// لإنشاء عدد صفوف (علي حسب المتاح) ارتفاع الواحد 50 or repeat(auto-fill , 50px)
// تصميم متجاوب or repeat(auto-fill , minmax(50px , 1fr))

- **gap** : (row-gap) (column-gap) // لتعيين المسافات او الفراغ بين الصفوف والاعمدة
gap : 10px 10px

- **justify-content** : flex-start or flex-end or center or space-between or ...
نفس الـ flex

- **align-content** : stretch or flex-start or flex-end or center or ..
نفس الـ flex

- **grid-template-areas** : “ x x y y y ” “ z z z z z ” ...
... الصف الثاني الصف الاول
// 1) نعين متغيرات للمساحات الناتجة من الصفوف والاعمدة
// 2) نعين اي مساحة لاي عنصر عن طريق الخاصية grid- area: x or y or z or .. التي توضع في الابناء
// 3) عن وضع قيمة (.) بدلا من x or y.. يترك مكان المساحة فارغة.

(2) خواص توضع في العناصر الموجودة داخل الاب:

- **grid- area**: x or y or z or ..

- **grid- column**: (grid- column-start) / (grid- column-end)
// لتعيين المساحة من العمود 1 الي العمود 5 (4 اعمدة) لعنصر ما **grid- column** : 1 / 5 or span 4

grid- row: (grid- row -start) / (grid- row -end) •

// لتعيين المساحة من الصف 2 الي الصف 4 (2 صفوف) لعنصر ما **grid- row : 2 / 4 or 2 / span 2**

لاحظ: يمكن ايضا كتابة الخاصية **grid- area** بهذه الطريقة:

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

grid- row : 2 / 2 / 5 / 6

Transform (2D)

transform: scalex(2 or 1.5 or..) •

لتمديد العنصر مرتين (عدد من المرات) في الاتجاه الافقي.. .. يتمدد العرض والهوامش و...

transform: scaley(2 or 1.5 or..) •

لتمديد العنصر مرتين (عدد من المرات) في الاتجاه الرأسي.. .. يتمدد الارتفاع والهوامش و...

transform: scale (scalex , scaley) •

يمكن كتابة قيم سالبة ... فيحدث انعكاس (افقي او رأسي او كلاهما)

transform: rotate(50deg or 156deg or..) •

لجعل العنصر يدور في اتجاه عقارب الساعة... اذا كانت القيمة سالبة يلف عكس عقارب الساعة

لاحظ: يمكن دمج الخاصيتين السابقتين كتالي **transform: scale (1.3) rotate(5deg)**

transform: translate (unit x , unit y) •

transform: translate (10px , -6px)

تستخدم لتحريك العنصر افقيا ورأسيا ، وهي افضل من تحريك العنصر عن طريق الـ **position** .

transform: skew (x , y) •

transform: skew (10deg , 6deg)

تستخدم في عمل انحراف للعنصر ..

لاحظ: يمكن تعيين العرض مثلا عن طريق **width: fit-content;** لجعل العرض مثل عرض محتوى العنصر فقط

- **transform: matrix** (scalex , skewy, skewx , scaley, translate, translatey) فقط لدمج مجموعة خواص.... لاحظ ان قيمة skewy, skewx تكون عبارة عن ظل الظاوية tan(angle)

- منشأ العنصر.. عند الدوران مثلا يدور حوله (المنشأ) **transform- origin: 50% 50% or center center**
- **transform- origin: 0% 50% or left center**
- **transform- origin: 100% 50% or right center**

هذه الخاصية تؤثر في طريقة عمل الخواص الاخرى،،،، وتعين للأب

Transform (3D)

- **transform: rotatex(50deg or..)**
لجعل العنصر يدور حول محور افقي X يمر بنقطة منشأه
- **transform: rotatey(50deg or..)**
لجعل العنصر يدور حول محور رأسي y يمر بنقطة منشأه
- **transform: rotatez(50deg or..)**
لجعل العنصر يدور حول محور عمودي Z يمر بنقطة منشأه
- **transform: rotate3d(0 or 1 , 0 or 1 , 0 or 1 , 50deg)**
هي تجميع للخواص الثلاث السابقة
- **transform: translate3d (unit x , unit y, unit z)** // لتحريك العنصر في الابعاد الثلاثة
لاحظ: لتحريك عنصر ما في الاتجاه Z يجب تعيين الخاصية **perspective : 300px** للاب.
- **backface-visibility: hidden or visible**
للتحكم في اظهار او اخفاء الوجه الخلفي للعنصر عند قلبه.
- **transform-style: preserve-3d**
تعين للاب بحيث عن قلبه وفيه عنصران يظهر العنصر الخلفي ... العناصر تحجز مكان في الـ 3d

• **animation-timing-function** : **ease** or **linear** // لتحديد منحنى الانتقال.

• **animation-direction**: **reverse** // لعكس اتجاه الانميشن.
• **animation-direction**: **alternate** // في الاتجاه الصحيح ثم يعكس الاتجاه وهكذا.

• **animation-delay** : **2s** // الوقت الذي ينتظره العنصر قبل البدء في الانميشن.

لاحظ: يمكن كتابة قيمة سالبة لاقتصاص جزء من بداية الانميشن.

• **animation-fill-mode**: **forwards** لجعل العنصر يأخذ تنسيقه من آخر فريم في الانميشن.... فمثلا اذا كان اخر فريم الخلفية سوداء تتغير غلفية العنصر الاساسية الي الاسود..

• **animation-fill-mode**: **backwards** لجعل العنصر يأخذ تنسيقه من اول فريم في الانميشن. يمكن استخدام القيمة **both**

• **animation-play-state**: **running** or **paused** للتحكم في ايقاف وتشغيل الانميشن ..

Media queries & responsive design

تستخدم للاستعلام عن شرط معين وبناء عليه يتم تنفيذ الكود الموجود داخلها من محددات وخصائص..

```
@media condition {  
    block of code  
}
```

(1) تغيير حجم الخط عند الطباعة...

```
@media print {  
    font-size : 15px }
```

(2) تغيير لون الخلفية اذا كان عرض العنصر من 0 الي 1000 بكسل..

```
@media (max-width: 1000px) {  
    div {background-color : green;} }
```

3) تغيير لون الخلفية اذا كان عرض العنصر من 500 الي 1000 بكسل..

```
@media (min-width: 500px) and (max-width: 1000px) {  
    div {background-color : green;} }  
}
```

لاحظ:

- يمكن وضع ملف css لل media كالتالي:
<link rel = "stylesheet" href= "style.css" media="print" >
يتم تطبيقه في حالة الطباعة... (ليس الافضل)
- لكل جهاز ابعاد معينة يمكنك التعرف عليها من الانترنت ..ابحث عن:
Media queries for standard devices
- عند تصميم موقع يستحسن البدء بالعمل علي التصميم الخاص بالهاتف... وهناك مينا يتم اضافتها وهي:
<meta name= "viewport" content= " width=device-width , initial-scale=1.0">
لكتابتها اسرع.. نكتب meta:vp ثم ...tab
- يمكن تعيين اكثر من class لعنصر واحد وبالتالي استدعائه باى واحد منهما... عن طريق:
<p class=" class1 class2 ">
- في صفحة الـ html وكتبت anyone. وضغطت enter سوف ينشأ <div class="anyname"><div> .
- تستخدم الخاصية scroll-behavior : smooth; لجعل الانتقال داخل الاقسام عن طريق اللينك مثلا يكون سلس..
- لاضافة ايقونات الي التصميم نستخدم العنصر التالي <i class="the icon class"><i> ... اشهرها التابعة لموقع font awesome .. تعامل معامل النص .. وتتكون بالاساس من ::before
- اذا كان هناك عنصرين داخل اب معين له الخاصية display : flex; ، وتريد لعنصر ان يأخذ باقي مساحة الاب تعيين له الخاصية flex : 1;