

Week 1

[01] - *Introduction and What Is SASS*

[02] - *SASS Compilation Tools*

[03] - *Import And Use And Advanced Architecture*

[04] - *Variables*

[05] - *Nesting And Parent Element*

[06] - *Property Declarations And Placeholder*

Assignments Week 1 From [1] => [6]

[07] - *Control Flow – If And Else*

[08] - *Create Triangle With If And Else*

[09] - *Interpolation*

[10] - *Comments And Documenting*

Assignments Week 1 From [7] => [10]

Week 2

[11] - *Mixin And Include*

[12] - *Loop – For*

[13] - *Loop – Each And Map*

[14] - Loop – While

[15] - Create Bootstrap Grid System

Assignments Week 2 From [11] => [15]

[16] – Function

[17] - Practice Mixin With Content

[18] - Practice Create Media Queries Mixin

[19] - The End And How To Master SASS

Assignments Week 2 From [16] => [19]

Tools And References

استخدام الـ @use والـ @forward في ملفات الـ SASS

الطريقة الثالثة لعمل SASS باستخدام الـ NPM Compile لملف الـ SASS

بنية الـ SASS او الـ SASS Architecture Structure وترتيب الملفات

الخاتمة وشوية فضفضة ❤️

لا تنسونا من دعواتكم ❤️

أذكر الله في نفسك ❤️



Never Forget That (Always Search)

By: Amr Elsayed

[01] - Introduction and What Is SASS

ملحوظة مهمة جداً في نهاية الملخص لازم تشووفها قبل ما تبدأ الـ Course دا اضغط هنا

SASS -----> Syntactically Awesome Style Sheet

الـ SASS عبارة عن إضافة او CSS Extension للـ

يعد الـ SASS مهارة مهمة جداً .. جداً للـ Front-End Developers

الـ SASS عبارة عن Preprocessor بمعنى إنك بتكتب الكود بالـ Syntax الخاص الـ SASS

وعن طريق الـ Compiler سيتم تحويل Code الـ SASS مكتوب بالـ CSS عشان المتصفح يقرأه ويفهمه

ليه استخدم الـ SASS ؟ !

1- عن الطريق الـ SASS هتقدر توفر وقت كبير جداً أثناء كتابة الـ Code

2- من خلال الـ SASS نسبة وجودة الأخطاء في الـ Code بتكون قليلة جداً

3- هتقدر من خلال الـ SASS تنظم ملفات الشغل وتخليلها مرتبة

4- ومن أهم مميزات استخدام الـ SASS هو إنك من خلاله بتقدر تستمتع بالمميزات الموجود في لغات البرمجة

من ضمن المميزات او الـ Features اللي هقدر استخدموها من خلال استخدام الـ SASS

• **Variables** -----> تقدر تنشأ متغيرات وـ Access عليها من أي مكان في الـ Code بتاعك وطبعاً المميز في الـ Variables هو إنك بسهولة تقدر تغير قيمتها وبالتالي تغير كل القيم اللي تم استخدام الـ Variables فيها

• **Functions** -----> بالرغم من وجود built-in Functions زي الـ Calc وغيرها إلا إنك تقدر تعمل Function خاصة بيك بتنفذ Action معينة

• **Loop** -----> عن طريقه تقدر تنشأ 50 عنصر بسطر كود بدل ما كنت بتعملهم يدوياً

• **Extend** -----> بدل ما تكرر الخواص لأكتر من عنصر ممكن تكتبهم مرة واحدة وتببدأ داخل كل عنصر تضيف الخواص دي بالـ Extend لأي عنصر موجود عندك

• **Control Flow** -----> استخدام الـ IF Condition واتخاذ القرار

• **Advanced Structure** -----> الـ SASS بيتيح لك بناء بنية محترمة جداً ومنظمة جداً يعني الـ Header بيكون في file وهكذا والـ Footer في file

[02] - SASS Compilation Tools

يمكنك تجربة تحويل كود من طريقة الـ **SASS** الى الـ **CSS** عن طريق موقع [sassmeister](#)

The screenshot shows two code editors side-by-side. On the left, under 'SCSS', is the following code:

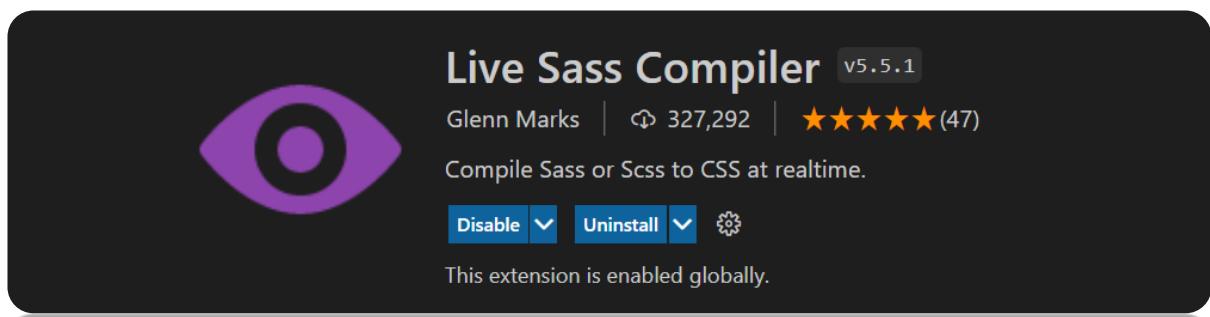
```
1 @for $i from 1 through 50 {  
2   .class-#${$i} {  
3     font-size: 10px + $i;  
4   }  
5 }
```

On the right, under 'CSS', is the resulting output:

```
1 .class-1 {  
2   font-size: 11px;  
3 }  
4  
5 .class-2 {  
6   font-size: 12px;  
7 }  
8  
9 .class-3 {  
10  font-size: 13px;
```

الموقع اللي عن طريقه قدرت اكتب كود **SASS** واعمله **Compile** لد **CSS** من باب التجربة
لو تلاحظ الكود اللي اتكلب بالـ **SASS** دا هو **Loop** الغرض منها تنشأ **Class 50** وفي كل **Class** يزيد حجم
الـ **Font-size** بمقدار **1px** عن اللي قبلة

داخل الـ **VS Code** عندي إضافة اسمها **Live Sass Compiler** لعمل الـ **CSS** الى الـ **SASS**
من



عشان اقدر استخدم إضافة الـ **SASS** هقوم بإنشاء ملف الـ **Live Sass Compiler**

The screenshot shows the VS Code interface with the 'Live Sass Compiler' extension installed. The top bar shows tabs for 'main.scss' and 'main.css'. The main editor area shows SCSS code:

```
1 @for $i from 1 through 30 {  
2   .class-#${$i} {  
3     font-size: 10px + $i;  
4   }  
5 }
```

A red arrow points from the bottom status bar to a tooltip that reads 'live compilation of SASS or SCSS to CSS'.

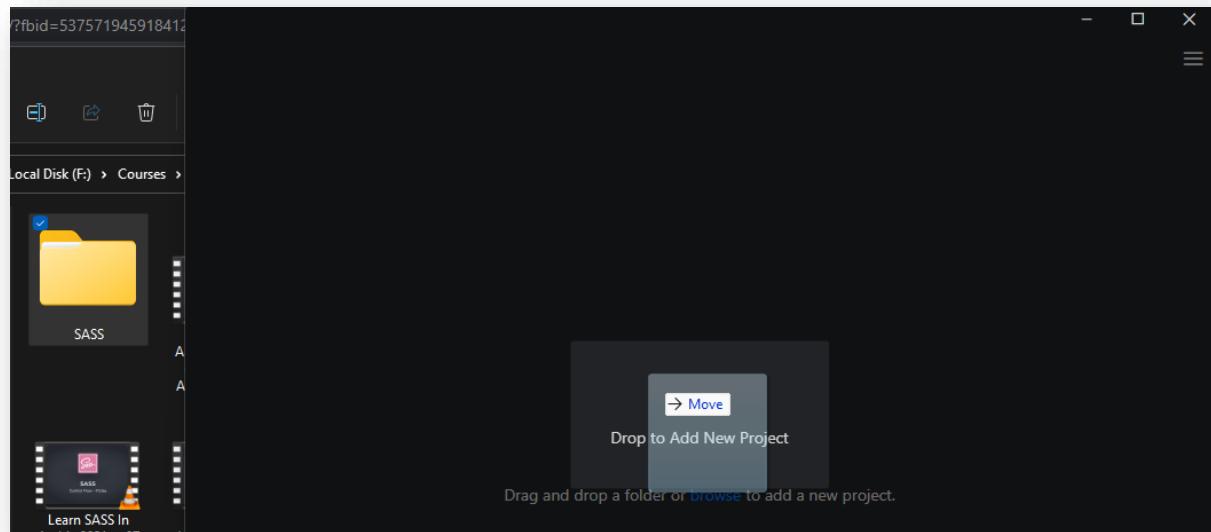
The bottom status bar shows icons for file count (0), warning count (0), quick fix (red arrow), 'Watch Sass' (green arrow), line number (Ln 6, Col 1), and a search icon (S).

وبعدين ما أنشئ ملف الـ **SASS** هكتب الكود في الملف دا بطريقة الـ **SASS** بعدها اعمل **Watch SASS** عشان اقدر استخدم الإضافة اللي ضيفتها في عمل **Compiling** للملف من الـ **SASS** الى الـ **CSS**

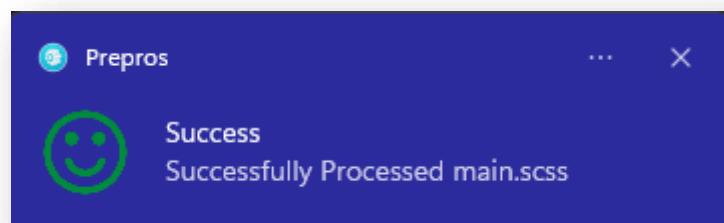
بمجرد ما تدوس على **Watch SASS** بيتم انشاء ملفين عندك ملف الـ **main.css** والـ **main.css.map**



ممكن تستغنى عن الإضافة خالص وتستخدم برامج خارجية زي [prepros](#) او زي [koala](#) او زي [Prepros](#) لانه حتي الان افضل من حيث المميزات اللي فيه



بمجرد ما اعمل أي تعديل في ملف الـ **Sass** بيبداً يقرأ التعديل برنامج **Propros** ويعمل **Compile** ويضيف الخواص اللي اتكتب في الـ **SASS** الى ملف الـ **CSS**



لا تنسونا من دعواتكم

By: Amr El-Sayed

[03] - Import And Use And Advanced Architecture

هنتعلم في الدرس دا مبدأ ال **import** او ال **Include** .. بمعنى ازاي اعمل ملف معين واستدعي الملف دا في مكان تاني عشان اقدر استخدمه وهنتعلم ازاي بيتم بناء بنية احترافية بسيطة في التعامل والتعديل على ال **Code**

مقدمة:

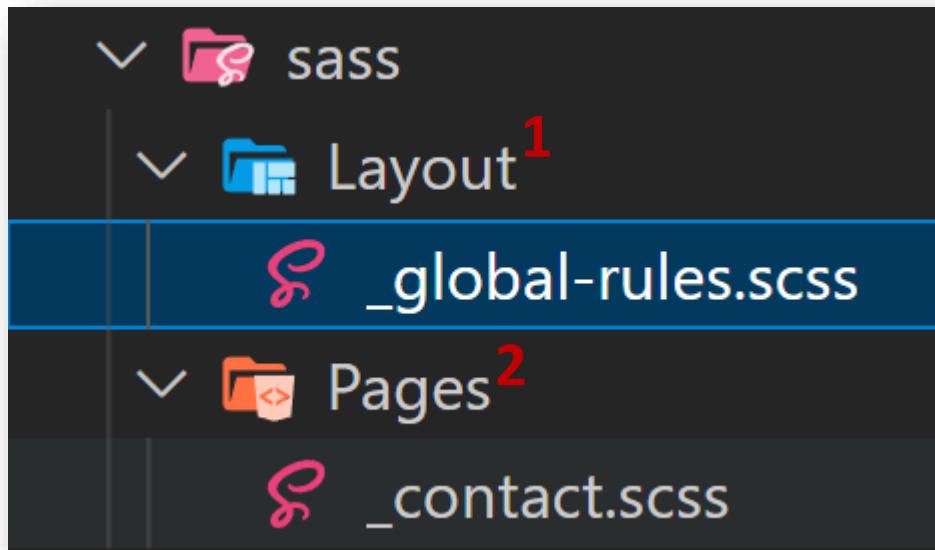
من مساوى إنك تخشن على ملف **CSS** مكتوب فيه **15 الف سطر كود** ومطلوب منك تعديل عليه هو انك وارد تكتب **Classes** كانت موجودة فعلاً ودا بالتالي يأثر ع تنسiqات كانت موجودة في مكان تاني ناهيك تماماً عن الحيرة والوقت اللي هتاخده عشان تدور ع الحاجة اللي تحتاج تعديله

لكن مع وجود ال **SASS** وبناء البنية السليمة اللي من خلالها بقدر اقسم تنسiqات الموضع لأجزاء .. عملية التعديل هتكون في الحالة دي سهلة وبسيطة جداً لأنك بتقسام شغلك لأكتر من صفحة وبتبدأ تستدعي الصفحات دي في ملف ال **SASS** الأصلي اللي في الآخر الملف دا بيتعمله **CSS Compile** لـ „،،، وهنشوف دا سوا حالا

هعمل الـ **New Folder** وهو اسمه **Sass** وجواها هحط الأجزاء اللي هشتغل عليها



بداخل الـ **sass** الـ **Folder 2** تانين الأول اسمه **Layout** و الثاني اسمه **Pages**



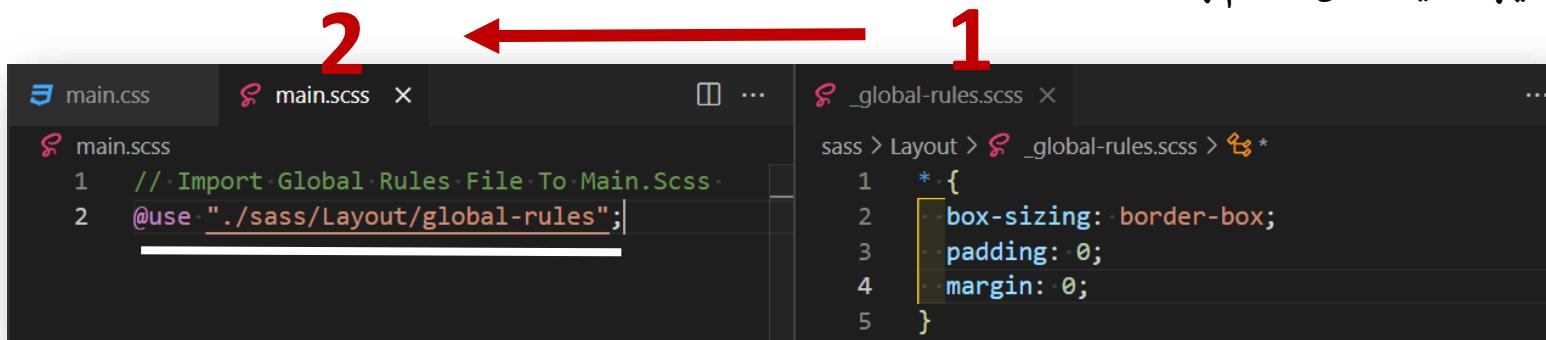
بداخل الـ **Layout Folder** الـ **_global-rules** دا انا عملت ملف اسمه **SASS** .. جواه هحط أي

وعايزك تركز اوكي آوي على الـ **_under score** الـ **_** اللي سبقت ملف الـ **SASS**

انا هنا حطيت الـ **_under score** الـ **_** دي قبل ملف الـ **SASS** لغرض واحد بس وهو اني بفهم الـ **Compiler** ان الملف دا انا مش عايز اعمله **Compile** .. يابشا بعد اذنك الملف دا ملکش دعوة بيه لاني هكتب فيه كود **SASS** وبعددين هستدعي الملف دا بالكود اللي فيه في ملف الـ **SASS** الكبير الرئيسي اللي هيتعمله هو

إذا ملف الـ **SASS** اللي انا مش عايز اعمله **Compile** لاني عملته عشان هستدعيه في ملف الـ **SASS** الأساسي
الكبير بسبق اسمه بعلامة الـ **_** **under score**

طيب خلينا نكمـل الكلام بتاعـنا

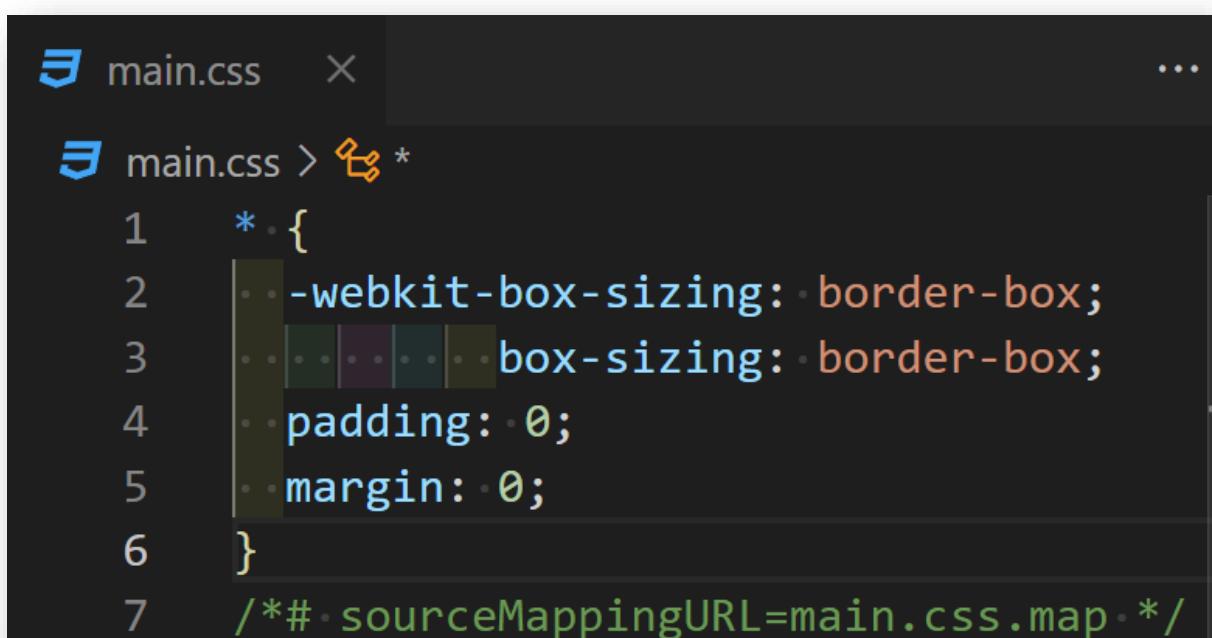


في ملف الـ **global-rules** _انا كتبت بعض الـ Global Rules اللي بستخدمها في بداية أي تصميم ليا

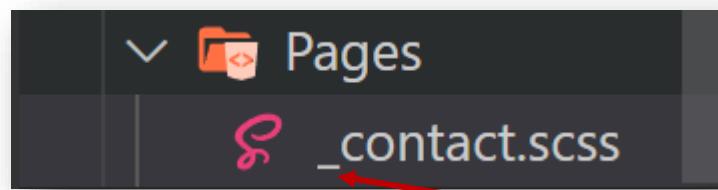
وفي ملف الـ **main** الأساسي بتبعي استدعيت ملف الـ **global-rules** _ ولاحظ ان ملف الـ **main** دا هو الوحيد اللي بيتحول أي كود جواه لـ **CSS**

لکن ال **global-rules** عشاں سبقتہ بال **under score** " _ " الملف دا هیتم تجاهل ال **Compiling** لیه

طيب تعالى نبص ع ملف الـ **CSS** بعد ما عملت **import** لملف الـ **global-rules** داخل ملف الـ **main.scss** وملف الـ **main.scss** دا اتعلمه **Compiling** لكود مكتوب بالـ **CSS** ... تعالى بقا نشوف كود الـ **CSS** دا



تعالي نعيid نفس التجربة على Pages Folder الـ جواها ملف الـ `_contact.scss`



ولاحظ تاني الـ **under score** اللي قبل اسم الملف دا عشان ميحصلوش **Compiling** لاني عامله عشان بس اكتب فيه اكواود خاصية بصفحة الـ **Contact** وھعمل للملف دا نفسه **Import** في ملف الـ **main.scss** الرئيسي

```
sass > Pages > _contact.scss > .contact
1   .contact {
2     padding: 20px;
3     font-size: 20px;
4     background-color: #f9f9f9;
5 }
```

بعد ما كتبت الـ **code** في ملف الـ `main.scss` هروح ملف الـ `_contact.scss` اعمله **import** بقا
بعمل استدعاء في ملف الـ **main** الرئيسي

```
main.scss

// Import _contact.scss File To main.scss File
@use "./sass/Pages/contact";
```

بعد ما عملته استدعاء في ملف الـ **main.scss** الرئيسي الملف دا بقا هيتعمله **Compile** ويتحول اللي فيه لكود CSS مكتوب في ملف الـ **main.css**

```
main.css

main.css > *
1   * {
2     -webkit-box-sizing: border-box;
3     box-sizing: border-box;
4     padding: 0;
5     margin: 0;
6   }
7   .contact {
8     padding: 20px;
9     font-size: 20px;
10    background-color: #f9f9f9;
11  }
12 /*# sourceMappingURL=main.css.map */
```

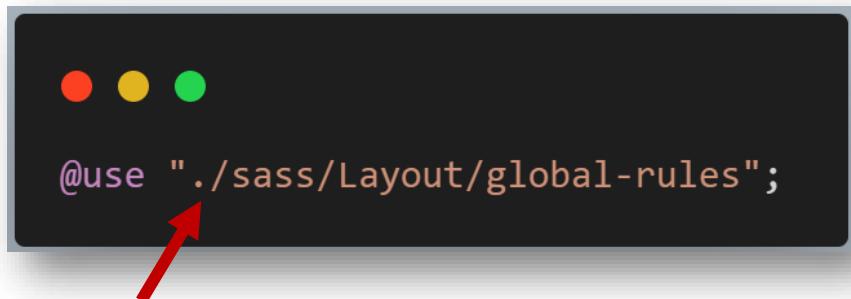
اهو ملف الـ **CSS** قرأ الكود اللي موجود في ملف الـ **SASS** اللي هو أصلًا ملف الـ **SASS** دا عبارة صفحات تانية عملتلها **Import** او استدعاء جوا ملف الـ **SASS** دا نفسه

طيب يبقى كدا فعلا لو عندي مشروع كبير جدا أقدر اقسمه لاكتير من صفحة كل صفحة فيها تنسيقات معينة خاصة بالصفحة دي وبعدين كل الصفحات دي اعملها استدعاء في ملف الـ **SASS** الرئيسي وبالتالي يتعمله **CSS** لكود **Compiling**

الطريقة دي هتساعدني جدا في تنظيم شغلي وسهولة الوصول لكل التنسيقات والتعديل عليها بسهولة عكس ما يكون عندي ملف **CSS** واحد وافضل ابحث جواه عن التنسيقات اللي تحتاج اعدلها والموضوع دا بيكون مرهق جدا ومضيع للوقت + انك وارد جدا تكتب **Class** يكون أصلًا موجود مسبقا فيعمل تأثير او ضرر ع تنسيقات تانية في صفحات تانية

شوية ملاحظات مهمه:

- لاحظ ان عملية الاستدعاء دي بتتم من خلال كلمة **use** كان الأول يستخدم كلمة **import** لكن الـ **use** أفضل من حيث الاستخدام
- طريقة الاستدعاء بتتم بعد كتابة كلمة **use** وبعدها بعمل علامتي تنصيص ".." وجوا العلامتين دول بحط نقطة . وبعدها الـ **/** وبختار مسار الملف اللي هيتم استدعاء داخل ملف الـ **SASS** الرئيسي



- الـ **Comment** اللي بتكتبه قبل كود الاستدعاء مش بيتم ترجمته وكتابته في كود الـ **CSS** اثناء عملية الـ **Compiling** دا بيكون عبارة عن **كومنت** ليك انت بتكتبه ليك انت

- الاكواد اللي بتكون في ملف الـ **SASS** دي بيطلق عليها مرحلة او مسمى الـ **Development Version** و بتكون في ملف الـ **CSS** بيطلق عليها مسمى الـ **Production Version** اما الاكواد اللي بيتعملها **Compiling**

لا تنسونا من دعواتكم

By: Amr El-Sayed

عبارات تحفيزية ❤️



Ahmed Abdelmonem

5d ·

...

من فترة كده قررت احمل LinkedIn وأشوف الابلكيشن ده
بيقول ايه... المهم مكانتش تجربة سعيدة يعني لقيت أن ناس
كثير قدي وأصغر مني محققين حاجات رائعة وأنا قاعد مكانني
ف كنت داخل على دور اكتئاب وجلد ذات كده
لحد ما عديت على آية (وَلَا تَتَمَنُوا مَا فَضَّلَ اللَّهُ بِهِ. بَعْضُكُمْ عَلَىٰ
بَعْضٍ لِلرِّجَالِ نَصِيبٌ مِمَّا أَكْتَسَبُوا وَلِلنِّسَاءِ نَصِيبٌ مِمَّا أَكْتَسَبْنَ)
طيب انا مش هتمنى ولا هحسد حد على رزقه بس بردو نفسي
اتقدم شوية ف كانت تكملاة الآية (وَسُلُّوا اللَّهُ مِنْ فَضْلِهِ إِنَّ اللَّهَ
كَانَ بِكُلِّ شَيْءٍ عَلِيمًا)
وفي الآخر الواحد مش في سباق مع حد ولو فضلت تبص لي
أعلى منك في الكاريير او المستوى الاجتماعي عمرك ما هترضى
لأن دايماً في حد أعلى مش معنى كده انك تكسل وترىح المهم
تبقى نسخة احسن من نفسك ف احرص على ما ينفعك واستعن
بالله ولا تعجزن

اذكر الله ❤️

سبحان الله وبحمده عدد خلقه ورضا نفسه وزنة عرشه ومداد كلماته

اللهم لك الحمد حمداً طيباً كثيراً مباركاً فيه كما ينبغي لجلال وجهك ولعظيم سلطانك يا رب العالمين

اللهم صل وسلام وبارك على نبينا محمد ﷺ

أَلَا بِذِكْرِ اللَّهِ تَطْمَئِنُ الْقُلُوبُ

لا تنسونا من دعواتكم

By: Amr El-Sayed

[04] - Variables

استخدام الـ **Variables** ديمى بيكون أعظم حاجه في البرمجة تخيل لو فيه لون معين موجود في كل مكان عندك في الصفحة موجود في الـ **Header** و موجود في الـ **Footer** وفي الـ **Content** بتابع الصفحة يعني موجود في اكتر من 50 مكان مثلاً وحبيت فجأة تغير اللون دا الي **لون تاني خالص** .. هل منطقي تشواف اللون دا موجود فين وتغييره 50 مرة بایدك؟ هنا بقا يجي دول الـ **variables** او **المتغيرات** .. من طبعتها انها **بتأخذ قيمة معينة** وبتبدأ تعوض عن القيمة دي باسم **المتغير** في كل مكان عايز تحط فيه القيمة دي لو في يوم حبيت تغير اللون دا لللون **لون تاني خالص** كل اللي عليك إنك تغير قيمة الـ **Variable** دا فقط مرة واحدة والتغير هيحصل في كل العناصر اللي موجود فيها الـ **Variable** دا

في الـ **CSS** كان برضو عندي استخدام للـ **SASS** لكن في الـ **Variables** لكن في الـ **SASS** استخدم المتغيرات بيكون مختلف شوية في فرق بين الاثنين ... يعني علي سبيل المثال لما يكون عندي متغير في الـ **CSS** بالشكل دا:

```
main.css

:root {
    --main-color: red;
}

.contact {
    padding: 20px;
    background-color: var(--main-color);
}
```

اسم المتغير نفسه وليس قيمته الفعلية

في الـ **CSS** كان المتغير يستخدم اسمه عادي لكن مكنش بيظهرلي قيمته الفعلية عكس الـ **SASS** لما بكتب فيه متغير وابدا استخدمنه لما بيتعمل **compile** للملف دا وابص في ملف الـ **CSS** بلاقي قيمة المتغير مش اسم المتغير عكس ما بيكون موجود في متغيرات الـ **CSS** لاحظ:

```
main.scss

$alt-color: green;

.header {
    color: $alt-color;
}

main.css

.header {
    color: green;
}
```

قيمة المتغير نفسه وليس اسمه

في عندي حاجتين مهمين جداً وهما الـ **Local Variable** والـ **Global Variable** .. أي الفرق بينهم؟!
الفرق بين الاثنين ببساطة شديدة جداً إن الـ **global variable** متاح للاستخدام من أي مكان داخل أي عنصر
لكن الـ **local variable** متاح للاستخدام داخل الـ **scope** اللي هو محظوظ فيه فقط

Global Variable

```
main.scss

// Global Variable
$alt-color: green;

.header {
  color: $alt-color;
}
```

Local Variable

```
main.scss

.footer {
  // Local Variable
  $alt-color: black;
  background-color: $alt-color;
}
```

بص بقا في ملف الـ **CSS**

```
main.css

.header {
  color: green;
}

.footer {
  background-color: black;
}
```

الـ **Header** خد اللون بتاعه من الـ **global variable** اللي هو اللون **Green**

اما الـ **footer** اللي حطيت فيه **alt-color** داخل الـ **scope** او النطاق بتاعه خد اللون بتاعه دا من الـ **local variable** **black** اللي هو اللون **variable**

ودا الفرق بين الـ **global** و الـ **local**

طيب لو حذفت الـ **local variable** الموجود في الـ **Footer** دا يحصل أي ؟ !

main.scss

```
.footer {  
    // Local Variable  
    // $alt-color: black; X  
    background-color: $alt-color;  
}
```

main.css

```
.footer {  
    background-color: green;  
}
```

لاحظ بعد ما حذفت الـ **local variable** الموجود في الـ **Footer** .. استعان بالـ **global** فورا

في حالة لو عايزة احول الـ **local variable** دا لـ **global variable** وهو موجود داخل الـ **Scope**

في الحالة دي هكتب جنبه **!global** اسمه **flag** .. وبالتالي ايه اللي هيحصل؟

الـ **header** هيأخذ اللون **green** من الـ **global var** اللي قبله .. الـ **footer** و الـ **copy-right** هيأخذوا اللون **black** لأن الـ **header** كدا كدا اللون **black** دا موجود جواه كـ **local variable**

والـ **copy-right** هيأخذ اللون **black** برضو لأنني حولت الـ **local var** دا لـ **global var** من خلال **!global flag** ... ولو مسكت الـ **header** دا حطيته في الآخر تحت خالص هيأخذ اللون الـ **black** برضو

SASS

main.scss

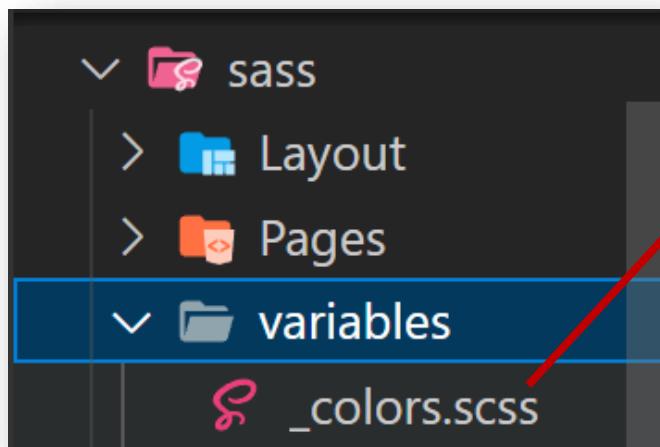
```
// Global Variable  
$alt-color: green;  
  
.header {  
    color: $alt-color;  
}  
  
.footer {  
    // Local Variable  
    $alt-color: black !global;  
    background-color: $alt-color;  
}  
  
.copy-right {  
    background-color: $alt-color;  
}
```

CSS

main.css

```
.header {  
    color: green;  
}  
  
.footer {  
    background-color: black;  
}  
  
.copy-right {  
    background-color: black;  
}
```

طيب خليني اعمل **Variables** خاص بال **SASS** دى واحظ جواه ملف **Variables** خاص بال **Folder**
واعمله استدعاء في ملف الـ **SASS** الرئيسي



```
_colors.scss ×  
sass > variables > _colors.scss > ...  
1 $main: orange;  
2 $alt: violet;
```

SASS - main.scss

```
// Import Variables  
@use "./sass/variables/colors" as*;
```

طريقة الاستدعاء المرادي بتكون مختلفة شوية .. وبعد الاستدعاء هبدأ استخدم المتغيرات في ملف الـ **SASS**

Sass - main.scss

```
.sidebar {  
background-color: $alt;  
color: $main;  
}
```

وادي الكود لما يتعمله **CSS Compile** لـ

CSS - main.css

```
.sidebar {  
background-color: violet;  
color: orange;  
}
```

ممكن اكتب طريقة الاستدعاء العادي جدا اللي انا متعود عليها

SASS - main.scss

```
// Import Variables  
@use "./sass/variables/colors";
```

ولكن اكتب اسم ملف الـ **SASS** بتابع الـ **Colors** قبل المتغيرات بتابعه

SASS - main.scss

```
.sidebar {  
    background-color: colors.$alt;  
    color: colors.$main;  
}
```

ممكن كمان اعمل الـ **responsive** الخاصة بال **media screen** في متغير ... يعني ع سبيل المثال فاكر كنت بكتب الـ **Media Queries** ازاي في الـ **CSS** كانت دي طريقة كتابة الـ **media screen**

CSS- main.css

```
@media (max-width: 767px) {  
    .sidebar {  
        padding: 20px;  
    }  
}
```

لكن في الـ **SASS** انا ممكن احط **الميديا** دي في متغير واعوض عنها بالمتغير دا بحيث لو غيرت قيمة المتغير دا بعدين الـ **Media Query** تتغير هي كمان

SASS - main.scss

```
$media_mob : "max-width: 500px";  
  
@media ($media_mob) {  
    .sidebar {  
        padding: 20px;  
    }  
}
```

CSS - main.css

```
@media (max-width: 500px) {  
    .sidebar {  
        padding: 20px;  
    }  
}
```

يعني في حال تغير القيمة **500px** لـ **750px** مثلا هيتغيرها في جميع الـ **Media Queries**

[05] - Nesting And Parent Element

في الـ **CSS** العادي كان أسلوب الـ **Nesting** بتعايي كالالتالي

CSS - main.css

```
.parent {  
    font-weight: bold;  
}  
.parent .child {  
    font-size: 18px  
}  
.parent .child .grand-child {  
    font-size: 15px;  
}
```

لكن في الـ **SASS** الموضوع أجمل بكثير عشان مش مضططر كل شوية تكتب الـ **nesting** من أوله بنفسك يعني انت هتكتب **parent** مرة وجووا الـ **child** هتكتب **child** مرة وجووا الـ **grand-child** هتكتب **grand-child** مرة

SASS - main.scss

```
.parent {  
    padding: 30px;  
    .child {  
        font-size: 18px;  
        .grand-child {  
            color: green;  
        }  
    }  
}
```

CSS - main.css

```
.parent {  
    padding: 30px;  
}  
.parent .child {  
    font-size: 18px;  
}  
.parent .child .grand-child {  
    color: green;  
}
```



المميز في الـ **Nesting** بإستخدام الـ **SASS** هواني بعمل **parent block of code** خاص بالـ **parent** دا والعناصر اللي جواه

لو جيت ع الـ **parent** دا وشيلت الخواص اللي فيه مش هيأخذه معاه او يكتبه أصلاً خالص وهيفضل محافظ ع الـ **Nesting**

```
SASS - main.scss
.parent {
  .child {
    font-size: 18px;
    .grand-child {
      color: green;
    }
  }
}
```

```
CSS - main.css
.parent .child {
  font-size: 18px;
}
.parent .child .grand-child {
  color: green;
}
```

اما في حالة لو عايز أقلل مستوى الـ **child** شوية هشيل الـ **grand-child** دا من داخل الـ **parent** واحطه داخل الـ **parent** نفسه وبكدا يقل مستوى الـ **parent**

```
SASS - main.scss
.parent {
  padding: 30px;
  .child {
    font-size: 18px;
  }
  .grand-child {
    color: green;
  }
}
```

```
CSS - main.css
.parent {
  padding: 30px;
}
.parent .child {
  font-size: 18px;
}
.parent .grand-child {
  color: green;
}
```

في حالة الـ **Grouping**

```
SASS - main.scss
.parent-one,
.parent-two {
  padding: 30px;
  .child {
    font-size: 20px;
  }
}
```

```
CSS - main.css
.parent-one,
.parent-two {
  padding: 30px;
}
.parent-one .child,
.parent-two .child {
  font-size: 20px;
}
```

طيب طرق التعامل مع الـ **Selectors** اللي اتعلمنته في الـ **CSS** هطبقه بالشكل دا في الـ **SASS**

يعني علي سبيل المثال الـ **direct child**

```
SASS - main.scss
.parent > {
  .child {
    font-size: 50px;
  }
  .test {
    color: red;
  }
}
```

```
CSS - main.css
.parent > .child {
  font-size: 50px;
}
.parent > .test {
  color: red;
}
```

اذا معني الكود دا في حالة لو عملت **parent** جنب الـ **direct child selector** مباشرة زي المثال كدا فدا معناه اني بطبق الـ **direct child** لكل العناصر اللي تحت الـ **parent**

لكن لو عايز اختار اول عنصر في المثال عشان يكون هو الـ **direct child** في المثال عشان يكون هو الـ **direct child**

```
SASS - main.scss
.parent {
  > .child {
    font-size: 50px;
  }
  .test { ✗
    color: red;
  }
}
```

```
CSS - main.css
.parent > .child {
  font-size: 50px;
}
.parent .test { ✗
  color: red;
}
```

من ضمن أمثلة الـ **next element** هو الـ **selectors +**

SASS - main.scss

```
.parent {  
  > .child {  
    font-size: 50px;  
  }  
  .test {  
    color: red;  
  }  
  + p {  
    font-weight: bold;  
  }  
}
```

CSS - main.css

```
.parent > .child {  
  font-size: 50px;  
}  
.parent .test {  
  color: red;  
}  
.parent + p {  
  font-weight: bold;  
}
```

يعتبر كل العناصر **parent class + next selector** مباشرة

SASS - main.scss

```
.parent + {  
  .child {  
    font-size: 50px;  
  }  
  .test {  
    color: red;  
  }  
  p {  
    font-weight: bold;  
  }  
}
```

CSS - main.css

```
.parent + .child {  
  font-size: 50px;  
}  
.parent + .test {  
  color: red;  
}  
.parent + p {  
  font-weight: bold;  
}
```

ممكن اكتب الـ **direct child** بشكل مختلف ومنظم أكثر

SASS - main.scss

```
.parent {  
  > {  
    .element-one {  
      font-size: 20px;  
    }  
    .element-two {  
      font-size: 10px;  
    }  
  }  
  .not-direct-child {  
    padding: 15px;  
  }  
}
```

CSS - main.css

```
.parent > .element-one {  
  font-size: 20px;  
}  
.parent > .element-two {  
  font-size: 10px;  
}  
.parent .not-direct-child {  
  padding: 15px;  
}
```

كدا direct child بتابع الـ **Block Of Code** موجودين جوا الـ **element-one** و **element-two** direct child خارج نطاق الـ **not-direct-child** اما الـ

SASS - main.scss

```
.parent {  
  > { Block of Code => Direct Child }  
}
```

اذا المثال اللي فات دا تحت الـ **parent** بحط علامة الـ **Direct Child** وافتتح القوسين **{}** وجواهم ببدأ اكتب كل العناصر اللي انا عايزة اعملها كـ **Select**

كل اللي فات دا كان طرق التعامل في الـ **Selectors** والـ **Nesting** اما القادر هو في طريقة التعامل مع

الـ **parent** نفسه

تخيل معايا ال HTML Code دا

HTML - index.html

```
<div class="box red">  
    Box - Red  
</div>  
  
<div class="box green">  
    Box - Green  
</div>
```

دلوقي انا عندي 2 الاتنين ليهم class اسمه box كل واحد فيهم ليه class مميز بلون غير الثاني عشان اعمل Select لكل عنصر من دول لو كتبت ال code كال التالي هيكون غلط :

SASS - main.scss

```
.box {  
    .red {  
        color: red;  
    }  
}
```

CSS - main.css

```
.box .red {  
    color: red;  
}
```

كدا غلط لأن كدا يعتبر كاني بقوله فيه عنصر ليه class red موجود جوا ال box ودا غلط أصلا وجود مسافة بين ال class red و ال box فدا معناه ان دا عنصر جواه عنصر red ودي مش الحقيقة الحقيقة هي ان عنصر ال box داليه class اسمه red مع بعض

SASS - main.scss

```
.box {  
    &.red {  
        color: red;  
    }  
}
```

CSS - main.css

```
.box.red {  
    color: red;  
}
```

الصح في الحالة دي هو استخدام رمز ال and او دا & لأنه بالشكل دا بيحل محل ال parent box او ال box لاحظ في ال CSS المسافة بين class وال box لأن المثال اللي كان فوق ودا معناه اني عملت Select لعنصر واحد ليه 2 class عكس المثال اللي

طيب طريقة التعامل مع الـ **pseudo element** التعامل بيكون ازاي ؟ ! نفس فكرة الـ **and** بالضبط او الرمز دا &

لو عايز اعمل **box** ع الـ **hover**

SASS - main.scss

```
.box {  
  &:hover {  
    font-size: 20px;  
  }  
}
```

CSS - main.css

```
.box:hover {  
  font-size: 20px;  
}
```

طيب لو عندي عنصر تاني داخل الـ **box** دا وحابب لما اعمل **box hover** العنصر اللي جواه دا يتغير

SASS - main.scss

```
.box {  
  .title {  
    font-size: 20px;  
  }  
  &:hover .title {  
    font-size: 30px;  
  }  
}
```

CSS - main.css

```
.box .title {  
  font-size: 20px;  
}  
.box:hover .title {  
  font-size: 30px;  
}
```

طيب لو عايز اعتمد علي تغير الـ **Box** دا اثناء تغير الـ **direction** بتاع الصفحة مثلا

SASS - main.scss

```
.box {  
  .title {  
    font-size: 20px;  
  }  
  [dir=rtl] & {  
    direction: rtl;  
  }  
}
```

CSS - main.css

```
.box .title {  
  font-size: 20px;  
}  
[dir=rtl] .box {  
  direction: rtl;  
}
```

♥ راقت لي ♥

دماغك كمبيوتر خارق

• حدث برامجه من خلال:



- الكتب
- البوتوكاست
- الخبرة

• احمي بطاريته من خلال:



- النوم الجيد
- التواصل مع الطبيعة
- التأمل والرياضة

• نظف قرصه الصلب عن طريق:



- حديث النفس
- العبادة
- ذكر الله

ادْعُو اللَّهَ

الزموا ذكر الله ففيه العون على متابعة الحياة

اللَّهُمَّ إِنِّي أَسأَلُكَ عِلْمًا نافِعًا، ورِزْقًا طَيِّبًا، وعَمَلاً مُتَقَبِّلًا.

اللَّهُمَّ أَعِنِّي عَلَى ذِكْرِكَ، وَشُكْرِكَ، وَحُسْنِ عِبَادَتِكَ.

اللهم صلّ وسلّم وزد وبارك على نبينا محمد ﷺ

أَلَا بِذِكْرِ اللَّهِ تَطْمَئِنُ الْقُلُوبُ

لا تنسونا من دعواتكم

[06] - Property Declarations And Placeholder

في خصائص في الـ CSS تكون لها short-hand زي علي سبيل المثال الـ font تعتبر لخواص كتير زي الـ font-family و الـ font-weight و الـ font-size

في الـ SASS في حاجه عظمه حرفيا لكتابه الـ Short-hand دي بسهولة شديدة جدا

SASS - main.scss

```
.box {  
    font-size: 15px;  
    font: { ←  
        size: 20px; ← }  
        weight: bold;  
        family: arial; } →  
    };  
    padding: {  
        top: 10px;  
        left: 5px;  
    };  
    margin: auto {  
        top: 10px;  
        bottom: 10px;  
    };  
    border: 1px solid red {  
        top: transparent;  
    };  
}
```

CSS - main.css

```
.box {  
    font-size: 15px;  
    font-size: 20px;  
    font-weight: bold;  
    font-family: arial;  
    padding-top: 10px;  
    padding-left: 5px;  
    margin: auto;  
    margin-top: 10px;  
    margin-bottom: 10px;  
    border: 1px solid red;  
    border-top: transparent;  
}
```

زي ما انت شايف كدا خاصية الـ font كتبت اسمها بس وبعدين داخل الـ {...} كتبت كل الخواص بتاعتتها زي الـ size و الـ family و الـ weight

نفس الفكرة في الـ padding ما هو برضو ليه short-hand زي الـ top و الـ bottom و الـ right و الـ left وكل اللي عليا إني كتبت اسم الخاصية وبعدين كتبت الـ short-hand ليها جواها

نفس الفكرة في الـ margin يكون auto من الجانبيين وبعدين حددت الـ top و الـ bottom بمقاسات معينة

ونفس الفكرة مع الـ border انا عامل border من كل الجوانب يكون 1px solid red ولكن في الـ top انا عايز اللون يكون transparent او شفاف

كل دا اترجم في الـ Css بمنتهي البساطة و اكتب بالـ SASS بطريقة بسيطة جدا

تعالي بقى نشوف قصة ال placeholder

في التصميم عندك ساعات بيكون عندك أجزاء كتير جدا واحدة نفس التنسيقات يعني وانت شغال ساعات بيكون عندك اكتر من **class** موجود فيهم 3 تنسيقات مثل موجودين فيهم كلهم جميا
كنت ممكن تتعامل معها بال **grouping** في ال CSS

ولكن في ال SASS في حاجة عظمة بتعمل نفس فكرة ال **grouping** بالضبط ولكن بطريقة افضل واسهل واسرع

SASS - main.scss

```
.main-box {  
background-color: white;  
padding: 15px;  
border: 1px solid #ccc;  
}
```

لفترضاني عندي التنسيقات دي في عنصر معين والمفروض نفس التنسيقات دي بالضبط ه تكون موجودة في اكتر من عنصر عندي في الصفحة

SASS - main.scss

```
.ads {  
background-color: white;  
padding: 15px;  
border: 1px solid #ccc;  
font-size: 20px;  
color: red;  
}
```

SASS - main.scss

```
.article {  
background-color: white;  
padding: 15px;  
border: 1px solid #ccc;  
font-size: 22px;  
color: green;  
}
```

عندي في ال SASS حل برمجي جميل جدا مفر للكتابة بشكل رهيب

SASS - main.scss

```
.ads {  
@extend .main-box;  
font-size: 20px;  
color: red;  
}
```

SASS - main.scss

```
.article {  
font-size: 22px;  
@extend .main-box;  
color: green;  
}
```

هنا ببساطة كأني بقوله امسك **main-box class** او فك الخواص اللي جواه في المكان دا

CSS - main.css

```
.main-box, .article, .ads {  
background-color: white;  
padding: 15px;  
border: 1px solid #ccc;  
}  
  
.ads {  
font-size: 20px;  
color: red;  
}  
  
.article {  
font-size: 22px;  
color: green;  
}
```

فه تكون دي النتيجة في ال CSS

شاف العناصر اللي ليها نفس التنسيقات

وعملهم groping مع بعض بنفس الخواص

بعد كدا شاف كل العناصر اللي ليها خواص مختلفة

عن الثانية فكتب كل عنصر لوحدة بخواصه المختلفة

طيب في حالة اني مفيش أصلا عندي عنصر ليه class اسمه main-box في الصفحة يعني العنصر دا ملوش وجود وانا كاتبه بس عشان يكون مجرد Source باخد منه الاكواود اللي هتتكرر اعملها extend في أماكن تانية

في الحالة دي بكتب قبل العنصر علامة %

SASS - main.scss

```
%main-box {  
background-color: white;  
padding: 15px;  
border: 1px solid #ccc;  
}
```

SASS - main.scss

```
.ads {  
@extend %main-box;  
font-size: 20px;  
color: red;  
}
```

معني الكلام دا اني بقوله معنديش أصلا عنصر ليه class بالاسم دا انا بس عامله عشان اخذ التنسيقات اللي فيه اضيفها في أي مكان انا عايزو ولكن هو فعلياً معنديش أصلاً عنصر بالاسم دا !

كإن بالظبط ال main-box دا زي خزنة كدا فيها شوية تنسيقات مش أكثر وقت ما حب استخدمها في أكثر من عنصر بدل ما اكتب التنسيقات دي بابيدي واكبر الكود أكثر وأكثر أقوله لا اعمل extend للتنسيقات دي في المكان دا وشكراً

Assignments Week 1 From [1] => [6]

[6] <= [1]

```
/* Assignment 1 */

.content {
  > .box {
    padding: 15px;
    > .title {
      font-size: 18px;
    }
  }
  .cat {
    padding: 15px;
  }
  &.premium::before {
    content: "Premium Box";
  }
  &.premium:hover {
    background-color: #eee;
  }
}
```

```
/* Assignment 2 */
```

```
.class > {
  .child {
    padding: 10px;
  }
}

.class {
  > .child {
    padding: 10px;
  }
}

.class {
  > {
    .child {
      padding: 10px;
    }
  }
}
```

```
/* Assignment 3 */
```

```
.class {
  font: {
    weight: bold;
    size: 20px
  }
}
```

```
/* Assignment 4 */

.parent {
  & + .sibling {
    margin: auto {
      top: 10px;
    }
    & ~ .last {
      padding: 10px;
    }
  }
}
```

```
/* Assignment 5 */

%group_one {
  background-color: white;
  padding: 20px;
}

.widget {
  background-color: white;
  padding: 20px;
}
.main-border {
  border: 1px solid #CCC;
}
.my-class {
  font-weight: bold;
  margin: 20px auto;
  @extend .widget;
  @extend .main-border;
}
```

```
/* Assignment 6 */

%widget {
  background-color: white;
  padding: 20px;
}
%main-border {
  border: 1px solid #CCC
}

.my-class {
  @extend %widget;
}
.my-class {
  @extend %main-border;
}
.my-class {
  font-weight: bold;
  margin: 20px auto;
}
```

```
/* Assignment 7 */
.box {
    h2, p {
        font-size: 20px;
    }
}

@media (max-width: 575px) {
    .box {
        h2, p {
            font-size: 16px;
        }
        h2 span , p span{
            font-weight: bold;
        }
    }
}
```

```
/* Assignment 8 */

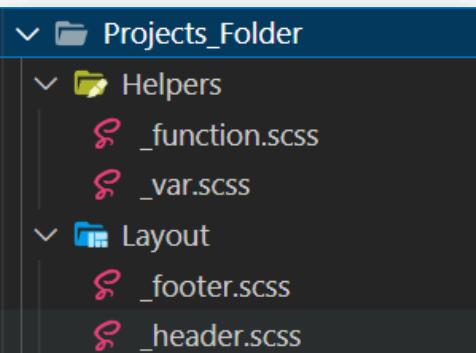
// The Code
// Do Not Edit Here
$red: #e91e63;
$green: #009688;

// Add What You Want Here
$link-color: #009688;
$link-hover: #e91e63;
// Do Not Edit Here
.box a {
    color: $link-color;
    &:hover {
        color: $link-hover;
    }
}
```

```
/* Assignment 9 */

//Helpers
@use "./Projects_Folder/Helpers/var";
@use "./Projects_Folder/Helpers/function";

//Layout
@use "./Projects_Folder/Layout/header";
@use "./Projects_Folder/Layout/footer";
```



```
/* Assignment 10 */
.box {
    font-size: 20px;
    &-holder {
        width: 1000px;
        &-content {
            margin: auto;
        }
    }
}
```

راقت لي ❤️



Abdulrahman Alaa

★ Favorites • Sep 28 at 11:30 AM • 🌎

...

ركز علي نفسك ، ركز في تطوير ذاتك فقط ، ربنا لو مآخر
عنك حاجة دلوقت عشان يقويك ، عشان اما يبعتلك الفرصة
تقدر تشيلها وتبقي قدما ، الفرص كدا كدا هتيجي ، استعدلها
عشان تستغلها ❤️✨



♥ اذكر الله

♥ الزموا ذكر الله ففيه العون على متابع الحياة

سبحان الله وبحمده سبحان الله العظيم

سبحان الله والحمد لله ولا إله إلا الله وأكبر ولا حول ولا قوه الا بالله العلي العظيم

سبحان الله وبحمده عدد خلقه ورضا نفسه وزنة عرشه ومداد كلماته

لا إله إلا أنت سبحانك أني كنت من الظالمين

أَلَا بِذِكْرِ اللَّهِ تَطمَئِنُ الْقُلُوبُ

لا تنسونا من دعواتكم

By: Amr El-Sayed

[07] - Control Flow – If And Else

تسألني أي أعظم حاجة بجد في لغات البرمجة اقولك الـ **Control Flow** او صنع واتخاذ القرار .. ليه ؟ ! عشان ببساطة شديدة **يمكنني** من اي اصنع اكتر من سيناريو واتحكم في كل سيناريو علي حسب تفاعل الـ **User** يعني ع سبيل المثال لو عندك موقع تحتاج تعمله **dark mode** او **light mode** او **عربي** و**انجليزي** كل دى **سيناريوهات** احتمال الـ **user** يتفاعل مع اي واحد منهم هنا بقى تيجي دور وعظمة الـ **IF and Else**

```
SASS - main.scss

$theme : 'dark';

.page {
    @if $theme == 'light' {
        background-color: white;
        color: #444;
    } @else {
        background-color: #444;
        color: white;
    }
}
```

عملت متغير اسمه **theme** جواه القيمة **light**

وبعد كدا عملت **Select** للـ **page** بتاعي كلها بقا وكتبت جوا الـ **class** بتاع الـ **IF** بتاعي بقوله لو المتغير **theme == "light"** يبقى في تنسيقات معينة هتتطبق زي ان الصفحة يكون لونها ابيض والكلام لونه اسود

لو متغير الـ **theme** دا بقا **"dark"** او اي قيمة غير **"light"** بقى يبقى تنسيقات تانيه هتتطبق زي ان الصفحة يكون لونها اسود والكلام لونه ابيض

```
@if Condition {
    Style
}
```

ودا **if** في الـ **SASS** بعد **if** بحط الشرط بتاعي ومن بعده بفتح **الاقواس { ... }** وجواها الـ **Syntax**

```
$theme : 'light';
```

CSS - main.css

```
.page {  
    background-color: white;  
    color: #444;  
}
```

```
$theme : 'dark';
```

CSS - main.css

```
.page {  
    background-color: #444;  
    color: white;  
}
```

هنا بقا عندي اكتر من سيناريو لو اليوزر حابب الـ **light mode** يبقى تنسيقات الـ **light mode** هتتطبق
لو حابب الـ **dark mode** يبقى تنسيقات الـ **dark mode** هي اللي هتتطبق

ممكن كمان تحط الـ **If Condition** داخل الخاصية نفسها مش بس داخل الـ **class** بشكل عام .. لا !
دا جوا الخواص نفسها اللي بتكتب داخل الـ **class** نفسه تقدر تعمل **IF Condition**

SASS - main.scss

```
$rounded : false;  
  
.box {  
    @extend %main-box;  
    border-radius: if($rounded, 6px, null);  
}
```

عندى متغير اسمه **rounded** قيمته
بتساوي **false**

عملت اسمه **class** وجواه عملت
اللي كنت بستخدمه عشان اخد
منه خواص احطها في أي **class** عندي

بعد كدا استخدمت خاصية الـ **border-radius** وجواها عملت الشرط بتاعي

قولت لو الـ **rounded** بتساوي **true** يبقى طبق خاصية الـ **border-radius** بالقيمة **6px**
وبعد كدا **null** دا في حالة لو قيمة الـ **rounded** دي **false** يبقى مفيش حاجه هتتطبق ولا هيكتب الـ **6px**

```
$rounded : true;  
  
.box {  
    @extend %main-box;  
    border-radius: if($rounded, 6px, null);  
}
```

```
.box {  
    border-radius: 6px;  
}
```

إذا اول ما المتغير قيمته تبقي **true** الخاصية هيطبق عليها الـ **6px** لو **false** مش هتتطبق أي حاجة

[08] - Create Triangle With If And Else

الدرس دا بمثابة Practice على ال Control Flow أو ال If Conditions

```
$direction : 'top';
```

```
<div class="arrow-div">Arrow</div>
```

```
@extend %main-box;
width: 300px;
height: 300px;
display: flex;
justify-content: center;
align-items: center;
font-size: 20px;
position: relative;
color: red;
margin: auto {
    top: 50px
};
font: {
    size: 20px;
    weight: bold;
    family: arial;
}
&::before {
    content: "";
    position: absolute;
    border: 20px solid transparent;
    @if $direction == 'top' {
        top: -40px;
        left: 50%;
        transform: translateX(-50%);
        border-bottom-color: red
    } @else if $direction == 'right' {
        right: -40px;
        top: 50%;
        transform: translateY(-50%);
        border-left-color: red
    } @else if $direction == 'bottom' {
        bottom: -40px;
        left: 50%;
        transform: translateX(-50%);
        border-top-color: red
    } @else if $direction == 'left' {
        left: -40px;
        top: 50%;
        transform: translateY(-50%);
        border-right-color: red
    } @else {
        @error "This Direction #{$direction} Is Not Valid"
    }
}
```

مسكت ال `div` بتاعي وحطت جواه بعض التنسيقات وعملت `@extend` للتنسيقات الموجودة في ال `main-box` كمان استخدمت أسلوب ال `short-hand` في كتابة الخواص اللي ليها `before` عشان اعمل السهم كل دا تنسيقات عاديّة جداً ومفهومه

تعالا نبدأ من أول ال `Control-Flow` انا عندي متغير اسمه `direction` بعتمد ع قيمته في اتخاذ القرار لو كانت قيمة المتغير بتساوي `top` بيكون عندي شوية تنسيقات لل `top` دا وهكذا مع تغيير قيمة المتغير لو مفييش أي قيمة من الأربعة مكتوبة في المتغير بيطلع رسالة مكتوب فيها ان الاتجاه المكتوب خاطئ

`$direction : 'top';`

CSS - main.css

```
.arrow-div::before {
  content: "";
  position: absolute;
  border: 20px solid transparent;
  top: -40px;
  left: 50%;
  -webkit-transform: translateX(-50%);
  transform: translateX(-50%);
  border-bottom-color: red;
}
```

`$direction : 'bottom';`

CSS - main.css

```
.arrow-div::before {
  content: "";
  position: absolute;
  border: 20px solid transparent;
  bottom: -40px;
  left: 50%;
  -webkit-transform: translateX(-50%);
  transform: translateX(-50%);
  border-top-color: red;
}
```

`$direction : 'left';`

CSS - main.css

```
.arrow-div::before {
  content: "";
  position: absolute;
  border: 20px solid transparent;
  left: -40px;
  top: 50%;
  -webkit-transform: translateY(-50%);
  transform: translateY(-50%);
  border-right-color: red;
}
```

`$direction : 'right';`

CSS - main.css

```
.arrow-div::before {
  content: "";
  position: absolute;
  border: 20px solid transparent;
  right: -40px;
  top: 50%;
  -webkit-transform: translateY(-50%);
  transform: translateY(-50%);
  border-left-color: red;
}
```

♥ عبارات تحفيزية ♥



♥ اذكر الله

♥ الزموا ذكر الله ففيه العون على متابع الحياة

اللهم صلّ وسلّم وزد وبارك علي نبينا محمد ﷺ
اللهم إني أستغفرك واتوب إليك، لا إله إلا أنت سبحانك إني كنت من الظالمين

أَلَا يَذِكُرِ اللَّهُ تَطْمِئْنُ الْقُلُوبُ

لا تنسونا من دعواتكم

By: Amr El-Sayed

[09] - Interpolation

من ضمن احـد افـضل مـمـيزـات الـ **SASS** هي الـ **interpolation** او بالعـربـي بـيـطـلـقـ عـلـيـها مـصـطـلحـ الـاقـحامـ لنفترض عـلـيـ سـبـيلـ المـثـالـ انهـ عـنـدـكـ **Website** بـيـعـرـضـ إـعـلـانـاتـ لـ **4 شـرـكـاتـ** مـثـلاـ فـيـ الـأـربعـ اـتـجـاهـاتـ الـلـيـ عـنـدـيـ سـوـاءـ **Bottom** او **Top** او **Right** او **Left**

كلـ شـرـكـةـ لـيـهـ صـورـةـ مـعـيـنةـ وـ **Background** مـعـيـنةـ خـاصـةـ بـهـوـيـتـهاـ هـنـاـ بـقـاـ تـيـجيـ حـلـاوـةـ الـ **interpolation** وـ اـسـمـ الشـرـكـةـ تـعـمـلـ الشـغـلـ كـلـهـ بـطـرـيقـةـ دـيـنـامـيـكـيـةـ لـاـنـ مـنـ خـلـالـ الـ **interpolation**

The diagram illustrates the SASS and CSS code blocks. A red bracket on the left side of the SASS block encloses the variable interpolation part of the selector and the background URL. A blue bracket on the left side of the CSS block encloses the resulting class name. A red arrow points from the SASS code to the CSS code, indicating the result of the interpolation.

```
SASS - main.scss
$Company : "Elzero";
$position: "left";

.ad-#${$Company} {
  font-size: 20px;
  background-image: url("imgs/#{$Company}.png");
  #{$position}: 0
}

CSS - main.css
.ad-Elzero {
  font-size: 20px;
  background-image: url("imgs/Elzero.png");
  left: 0;
}
```

فيـ المـثـالـ دـاـ مـثـلاـ اـنـاـ عـنـدـيـ مـتـغـيرـ اسمـهـ **company** قـيمـتهـ بـتـكـونـ اـسـمـ الشـرـكـةـ نـفـسـهـ زـيـ **Elzero** وـ عـنـدـيـ مـتـغـيرـ تـانـيـ اـسـمـهـ **position** قـيمـتهـ اـسـمـهـ **left** وـ هـشـرـ حلـكـ حـالـ النـقطـةـ دـيـ

عملـتـ أـيـ جـيـتـ عـلـىـ اـسـمـ الـ **class** وـ كـتـبـتـ **ad** وـ عـوـضـتـ مـنـ خـلـالـ الـ **interpolation** بـالـمـتـغـيرـ **company** فـبـقـيـ الـ **class** اـسـمـهـ **ad-Elzero** دـيـ هـيـ قـيمـةـ الـ **المـتـغـيرـ company**

بعـدـ كـدـاـ الـ **Background image** لـنـفـرـضـ انـ هـيـ فـيـ **Folder** لـيـهـ شـكـلـ معـيـنـ فـيـ التـسـمـيـةـ يـعـنيـ مـثـلاـ اـسـمـهـ بـيـكـونـ اـسـمـ الشـرـكـةـ وـ اـمـتـدـادـاـهـ **png**

فـيـ الـمـسـارـ نـفـسـهـ عـوـضـتـ عـنـ اـسـمـ الصـورـةـ الـلـيـ اـمـتـدـادـاـهـ **company** فـلـوـ اـتـغـيـرـ قـيمـتهـ لـدـيـ قـيمـةـ فـيـ أـيـ وـقـتـ اـسـمـ الصـورـةـ هـيـبـقـيـ نـفـسـ الـقـيمـةـ الـمـوـجـودـةـ لـلـمـتـغـيرـ **company**

بعـدـ كـدـاـ اـنـاـ عـنـدـيـ مـتـغـيرـ اـسـمـهـ **position** قـيمـتهـ **left** فـاسـتـخـدـمـتـ الـ **interpolation** فـيـ التـعـويـضـ عـنـ اـسـمـ الـ **الـخـاصـيـةـ** نـفـسـهـ وـ اـدـيـتـهـ الـ **قيـمةـ 0**

SASS - main.scss

```
$Company : "falcon";
$position: "right";

.ad-#${$Company} {
  font-size: 20px;
  background-image: url("imgs/#{$Company}.png");
  #{$position}: 0
}
```

CSS - main.css

```
.ad-falcon {
  font-size: 20px;
  background-image: url("imgs/falcon.png");
  right: 0;
}
```

بمجرد ما غيرت قيمة المتغيرات تم التعامل مع التغيير في كل أنحاء الـ **code** واتغير اسم **الشركة** و **الاتجاه** اللي هيكون فيه الإعلان **واسم الصورة** اللي ديمى بيكون ليه **Pattern** معين عندي اللي هو اسم الشركة فقط **png**

```
$Company : "Programming";
$position: "top";

.ad-#${$Company}-#{unique-id()} {
  font-size: 20px;
  background-image: url("imgs/#{$Company}.png");
  #{$position}: 0
}
```

```
.ad-Programming-uzz0okb {
  font-size: 20px;
  background-image: url("imgs/Programming.png");
  top: 0;
}
```

عندي كمان **built-in function** اسمها **unique-id()**

بتعمل **Generate** لارقام عشوائية ملهاش شبيه ابدا ممكن احطها جنب اسم الشركة للتميز اكتر واكتر

الـ **built-in function** **interpolation** دي بتتم من خلال استخدام الـ

عشان تستخدم الـ **interpolation** بيكون من خلال طريقة الكتابة دي ==> **#{}==>**

[10] - Comments And Documenting

اول نوع من أنواع الـ **Comment** هو الـ **Comments** اللي بيظهر ليك انت فقط كـ **Developer** مش بيظهر في ملف الـ **CSS**

```
// This Comment Will Not Show In CSS File
```

بيكون بالشكل دا

تاني نوع اللي بيظهر في ملف الـ **CSS** عادي ولكن مش بيظهر لو عملت **minify** او ضغط لملف الـ **CSS**

```
/* This Comment Will Show In CSS File But Not In Compressed Mode */
```

بيكون بالشكل دا

ثالث نوع اللي بيظهر في ملف الـ **CSS** وكمان بيظهر لو عملت **minify** لملف الـ **CSS**

```
/*! This Comment Will Show In CSS File And In Compressed Mode */
```

بيكون بالشكل دا

ممكن تكتبه في أي مكان في الـ **code** **Comment**

```
.box /* Multiple  
Lines  
Comment */ {  
    font-size: 20px; // Inline Comment  
}
```

ممكن داخل في الـ **comment** نفسه تستخدم الـ **interpolation**

```
/* This Comment Contains Interpolation => #{$company} */
```

```
/* This Comment Contains Interpolation => falcon */
```

Assignments Week 1 From [7] => [10]

[10] <= [7]

```
/* Assignment 1 */

$main: "school";
$child: "box";
$language: "en"; // English
$gap: 20px;

.#{$main} {
  padding: $gap;
  @if $language == "en" {
    direction: ltr;
  } @else {
    direction: rtl;
  }
  margin-left: $gap;

  & .#{$main}-#{$child} {
    padding: #{$gap - 10px};
  }
  &:hover .#{$main}-#{$child} {
    background-color: #eee;
  }
}
```

```
/* Assignment 2 */

// SASS Code
$main-color: red;
$alt-color: blue;

/* Theme Version: 1.0 */
/* Website Theme Main Color Is: #{$main-color} */
/* Website Theme Alternate Color Is: #{$alt-color} */
```

```
/* Assignment 3 */

// SASS Code
$number: 10;
$number: 8;
$number: 5;

.class {
  @if $number > 8 {
    color: blue;
  } @else if $number > 5 {
    color: green;
  } @else {
    color: red;
  }
}
```

```
/* Assignment 4 */

// SASS Code
$number: 10;
$color: red;
$theme: light;

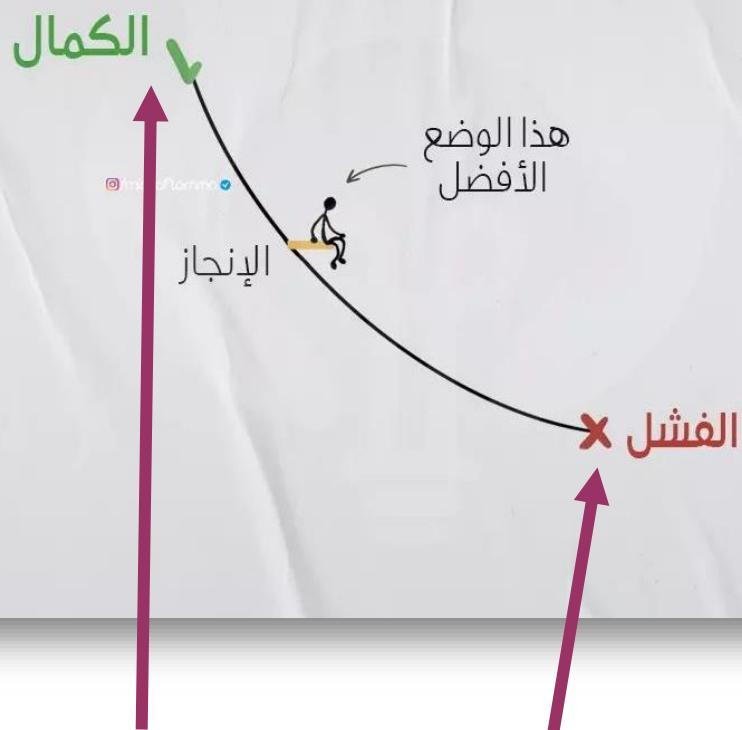
// Write If Condition Code Here
.class {
  @if $number > 5 & $color == "red" & $theme != "dark" {
    color: red;
  }
}
// Output
.class {
  color: red;
}
```

```
/* Assignment 5 */

// SASS Code
$number: 10;

.box {
  @if $number == 10 {
    background-color: #EEE;
    text-align: center;
  } @else if $number == 9 {
    background-color: #EEE;
  } @else if $number == 8 {
    /*Nothing*/
  }
}
// Output
.box {
  background-color: #EEE;
  text-align: center;
}
```

2- لا تبحث عن الكمال لأنك يمنعك من الإنجاز:



يعني ولا تنزل للقاع وتبقي عند نقطة الفشل ولا تدور ع نقطة الكمال لأنك عمرك ما هتوصلها
والأفضل ليك النقطة اللي في النص وهي نقطة الإنجاز

 اذكر الله

 الزموا ذكر الله فيه العون على متابعة الحياة

لا إله إلا الله وحده لا شريك له .. لـه الملك وـله الحمد وهو على كل شيء قادر

أَلَا بِذِكْرِ اللَّهِ تَطْمَئِنُ الْقُلُوبُ

لا تنسونا من دعواتكم

Never Forget That (**Always Search**)

[11] - Mixin And Include

الـ **Mixin** من أعظم الحاجات الموجودة في الـ **SASS** ... تعالى نشوف مثال

```
@mixin list-reset {  
  padding: 0;  
  margin: 0;  
  list-style: none;  
}  
  
ul.links {  
  @include list-reset()  
}
```

```
ul.links {  
  padding: 0;  
  margin: 0;  
  list-style: none;  
}
```

بكتب الـ **@mixin** واسم الـ **mixin** اللي انا عايزو زي اسم **list-reset** وجواها الخواص اللي هيتعملها في أي عنصر عندي زي ما استخدمت الـ **include** كدا في العنصر **ul.links**

واحد هييجي يقولي طيب واي لزمه الـ **mixin** ما انا ممكن استخدم الـ **@extend** اقولك لا ... دا الـ **mixin** دا هتللاقى فيه شغل برمجي ديناميكى محترم جدا وتعالي نشوف المثال الجاي

```
@mixin circle($dimensions) {  
  border-radius: 50% ;  
  width: $dimensions;  
  height: $dimensions;  
}  
  
.circle-100 {  
  @include circle(100)  
}
```

```
.circle-100 {  
  border-radius: 50%;  
  width: 100;  
  height: 100;  
}
```

زي ما انت شايف كدا انا عملت **@mixin** اسمها **circle** ومررت جواه **متغير** او بلغة الـ **JS** في الـ **function** دا يعتبر **Parameter** كدا وبعدين كتببت الخواص في الـ **mixin** بتاعي .. بعد كدا عملت متغير اسمها **circle-100** وجواه عملت **@include** للـ **circle** بتاعي ومكان المتغير او الـ **Parameter** حطيت قيمة او **Argument** لاحظ **القيمة** اللي مررتها اللي هي **100** تم التعويض عنها في الـ **mixin** مكان الـ **width** و الـ **height** في الـ **CSS**

```
.circle-200 {  
  @include circle(200);  
  background-color: red;  
  color: white;  
}
```

```
.circle-200 {  
  border-radius: 50%;  
  width: 200;  
  height: 200;  
  background-color: red;  
  color: white;  
}
```

بعض أول ما أغير قيمة المتغير الموجود في الـ **mixin** بيكراً القيمة الجديدة بشكل ديناميكي ويترجمها في الـ **CSS**

ممكن اعمل **mixin** يستدعي **mixin** تاني والـ **mixin** تالت خالص تعالوا نشوف

```
@ mixin circle($dimensions) {  
  border-radius: 50% ;  
  width: $dimensions;  
  height: $dimensions;  
}
```

```
@ mixin centering {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}
```

```
.center-circle {  
  @include circle(400);  
  @include centering;  
}
```

```
.center-circle {  
  border-radius: 50%;  
  width: 400;  
  height: 400;  
  display: -webkit-box;  
  display: -ms-flexbox;  
  display: flex;  
  -webkit-box-pack: center;  
  -ms-flex-pack: center;  
  justify-content: center;  
  -webkit-box-align: center;  
  -ms-flex-align: center;  
  align-items: center;  
}
```

لاحظ انا عندي الـ **mixin** **centering** وعملت **circle** بتاع الـ **mixin** جديده للـ **centering** وبعد كدا أنشأت عنصر اسمه **center-circle** وجواه ضيفت الـ **2 mixins** اللي عندي وكان ممكن اضيف الـ **circle mixin** دا جوا الـ **centering mixin** من البداية أصلاً واعمل **center-circle** في العنصر اللي اسمه **centering mixin** للـ **center-circle**

```

@mixin centering {
  display: flex;
  justify-content: center;
  align-items: center;
  @include circle(400); ←
}

.center-circle {
  @include centering;
}

```

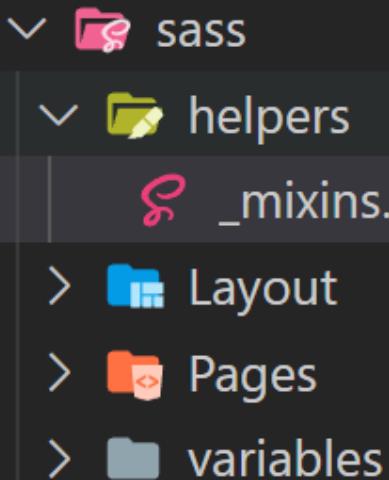
زي ما انت شايف كدا استدعيت ال circle mixin بداخل ال centering mixin

وبعدين عملت `@include` فقط للـ centering mixin داخل العنصر اللي اسمه .center-circle

وهيكون عندي نفس النتيجة بالضبط دون اختلاف وبكدا اقدر استدعي mixin جوا mixin

ولكن اول طريقة فوق هي الأفضل إنك تعمل كل mixin لوحده وستدعى كل mixin لوحده داخل العنصر

وزي ما قولتلك في الأول أجمل حاجة في ال SASS هي تنظيم الملفات والتنسيق يكون سهل الوصول ليه وسهل التعديل عليه لذلك انا ممكن اعمل Folder او بمعني أدق ملف خاص بال mixins وابداً استدعيها في ملف ال SASS الرئيسي من خلال ال `@use`



زي ما انت شايف كدا عملت helpers اسمها folder اللي جيت بعد كدا عملته وجواه ملف _mixins.scss في ملف ال SASS الرئيسي من خلال ال `@use`

```
// Import Mixins
@use "./sass/helpers/mixins" as *;
```

ودي المكتبة الخاصة بال mixins اللي تم ذكرها في الفيديو من خلال [اللينك دا](#)

انت كمان تقدر تبني مكتبة مليانة mixins خاصة بيك تستخدمنها في أي وقت ❤️

[12] - Loop – For

لوانت جاي من خلفية برمجية وفاهم يعني أي **Loop** دلوقتي بقا هنشوف الـ **SASS** شكله أي في الـ **loop** ممكن تسأل نفسك وتسألني سؤال ياتري ممكن استفيد أي من الـ **loop** في حالة الـ **SASS** ؟

لنفترض انك بتبني مكتبة كبيرة ومطلوب منك انشاء 20 او 30 = < class هل طبيعي تعمل كل **Class** من الـ 30 دول **بإيدك واحد واحد** ؟!

هنا يجي دول الـ **Loop** اللي بسطرين **code** بالضبط تقدر تعمل دا بمنتهي البساطة تعالى نشوف مثال بسيط

```
@for $i from 1 through 5 {  
  .class-#${$i} {  
    font-size: #{10px + $i};  
  }  
}
```

```
.class-1 {  
  font-size: 11px;  
}  
.class-2 {  
  font-size: 12px;  
}  
.class-3 {  
  font-size: 13px;  
}  
.class-4 {  
  font-size: 14px;  
}  
.class-5 {  
  font-size: 15px;  
}
```

زي ما انت شايف كدا عملت **Loop** من خلاله تم انشاء 5 عناصر الـ **i** دي عبارة عن الـ **index** وبما اني بدأت الـ **loop** من 1 لحد 5 فمعني الكلام دا ان الـ **i** دي هتكرر 5 مرات من 1 الى 5 في الحالة دي انا قولتلle اعمل **class** وجنبه الـ **i** اللي هتنزيد كل مرة من 1 الى 5

وبعدين عملت الـ **font-size** ببسا من **i + 10px** اللي هي بتزيد بمقدار 1 ديميا بداية من 1 الى 5 ... فأول مرة هتكون **1 + 10px** بـ 11 وبعدين **1 + 11px** ه تكون بـ 12px وهكذا الحد ما يوصل لـ 5

```
@for $i from 1 to 5
```

```
@for $i from 1 through 5
```

طبعا في اثناء استخدام الـ **@for** في فرق بين اني اكتب **through** وبين اني اكتب **to** الفرق بينهم انه لو كتبت **5 through 1** فدا معناه انه هيعمل من أول 1 لحد 5 اما لو كتبت **to 5 1** فدا معناه انه هيعمل من أول 1 لحد 4 ... Not Including End

```
$dimensions: 0;

@for $i from 1 through 5 {
  .circle-#{100 + $dimensions} {
    width: $dimensions + 100px;
    height: $dimensions + 100px;
  }
  $dimensions: $dimensions + 100;
}
```

عملت متغير اسمه **\$dimensions** وحزنت قيمته بـ **0**

بعدين بدأت الـ **For Loop** بتاعي بالـ **SASS Syntax** زي ما اتعلمت

داخل الـ **Loop** عملت الـ **Class** بتاعي اسمه **circle** وطلبت انه يمسك الـ **\$dimensions** اللي قيمته **0** يزود عليه **100** في كل مرة هي عمل فيها الـ **Loop** اللي هما **5** مرات من **1** الى **5**

بعد كدا داخل الـ **class** نفسه استخدمت الخواص **width** و الـ **height** وعوضت عنها بالـ **100px** اللي قيمته **0** وزدت **100px** عشان يبدأ من أول

بعد كدا عملت معادلة هيكون فيها الزيادة اللي بقول فيها **\$dimensions: \$dimensions + 100;** واللي كان الغرض منها اني في كل مرة امسك قيمة الـ **\$dimensions** ازود عليها **100** لمدة **5** مرات اللي هما عدد مرات تكرار الـ **Looping**

فيقت العملية شغالة كال التالي الـ **\$dimensions** بـ **0** دخلته جوا الـ **Loop** وزدت على قيمة **100** فبقت قيمته الحالية اللي هيبدأ منها انشاء أول **class** هي الـ **class-100** $= <= 100$ دي بقت البداية لأول

ذلك الأمر في الـ **width** و الـ **height** من **100px** اللي هي قيمة الـ **\$dimensions** بـ **0** القديمة
بعدين عملت المعادلة زي ما شرحنا مسبقا

```
@mixin centering {  
    display: flex;  
    justify-content: center;  
    align-items: center;  
}
```

```
$dimensions: 100;  
  
@for $i from 1 through 5 {  
    .circle-#{$dimensions} {  
        width: #{$dimensions}px;  
        height: #{$dimensions}px;  
        border-radius: 50%;  
        @include centering();  
        background-color: red;  
        color: white;  
        margin: 50px auto;  
    }  
    $dimensions: $dimensions + 100;  
}
```

في المثال دا انا حبيت اعمله بشكل مختلف بنفسي وأطبق ع ال mixin كمان بالمرة مش عارف اللي عملته دا ولا لاء بس حبيت اجرب حاجة بنفسي **best practice**

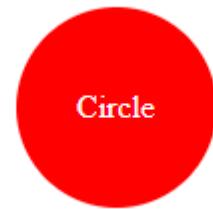
انا عملت قيمة ال \$dimensions ب 100 على طول وعوضت عنها في ال Class على طول CSS وعوضت عنها في ال height علي طول بس كتبت جنبها كلمة px عشان تبقى 100px في ال

centering كان معمول من الدرس اللي فات بتاع ال @mixin @include بعد كدا عملت بعدها كتبت شوية خواص لزوم باقي التنسيق

تعالي نشوف النتيجة

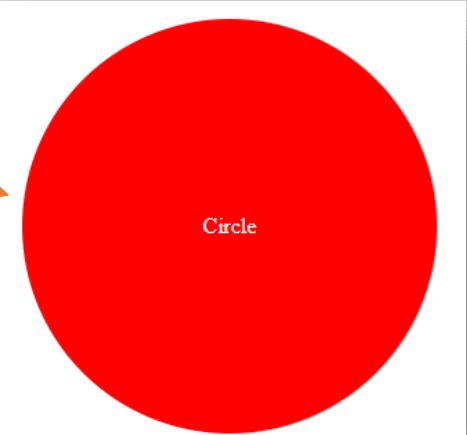
```
.circle-100 {  
  width: 100px;  
  height: 100px;  
  border-radius: 50%;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  background-color: red;  
  color: white;  
  margin: 50px auto;  
}
```

```
<div class="circle-100">Circle</div>
```



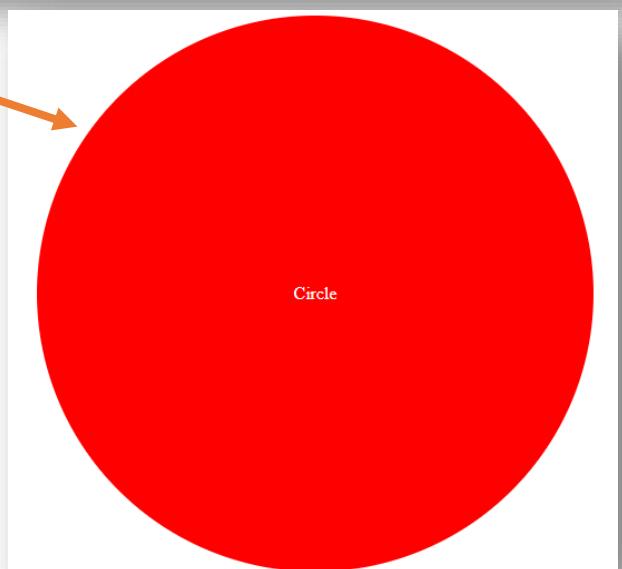
```
.circle-200 {  
  width: 200px;  
  height: 200px;  
  border-radius: 50%;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  background-color: red;  
  color: white;  
  margin: 50px auto;  
}
```

```
<div class="circle-200">Circle</div>
```



```
.circle-500 {  
  width: 500px;  
  height: 500px;  
  border-radius: 50%;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  background-color: red;  
  color: white;  
  margin: 50px auto;  
}
```

```
<div class="circle-500">Circle</div>
```



[13] - Loop – Each And Map

لو فيه تصميم انت شغال عليه وفيه اكتر من لون وحابب توزع الألوان دي كلها علي كل عنصر عندك او بمعنى آخر حابب كل عنصر عندك يكون ليه لون من الألوان اللي عندك في التصميم فأفضل واسع حل للموضوع دا هو استخدام الـ `@each` تعالوا نشوف

```
$themes: red, blue, green, orange;

@each $theme in $themes {
  .#$theme-theme {
    .product {
      background-color: white;
      border-bottom: 2px solid $theme;
      .title {
        color: $theme;
        font-weight: bold;
      }
    }
  }
}
```

انا كدا يعتبر عملت **Loop** باستخدام الـ `@each` ... تعالي نفهم اللي حصل

انا عملت متغير اسمه `$themes` حطيت جواه اكتر من **Color**

بعد كدا بدأت استخدم الـ `@each` .. بقول فيها كل `$theme` من اللي عندي موجود داخل المتغير `$themes` اللي هو أصلًا فيه اكتر من **Color**.. امسك بقا كل `$theme` في المتغير الرئيسي فوق طبق عليه التالي:

1- اسم الـ `$theme` دا في كل مرة هيحصل **Loop** فيها علي الـ `Colors` وجنبه كلمة `theme`

```
.#$theme-theme
```

اللي هو هيكون مرة `red` ومرة `green` ومرة `blue` فيها خذ أسماء الـ `Colors` دي من المتغير من خلال الـ `interpolation` ويضيف جنب كل واحدة كلمة `theme` فتبقي النتيجة مرة `red-theme`. ومرة `green-theme`. وهكذا تكون `blue-theme`.

2- كل `product` داخل كل `theme` عندي بالترتيب كدا هياخذ نفس اللون اللي الـ `loop` عدا عليه يعني كل الـ `products` هيكون ليها `background-color` ابيض، ولكن في الـ `border-bottom` لو الـ `$theme` اللي عليه الدور `red` يبقى الـ `border-bottom` `red` --- لو الـ `$theme` اللي عليه الدور لونه `blue` يبقى الـ `border-bottom` `blue` هيكون

3- كل `product` جواكل `title` هيكون ليه `Color` اللي عليه الدور في الـ `Loop` لو كان `red` يبقى الـ `title-color` هيكون `red` وهكذا

عارف لو مكنتش استخدمت الـ @each كان الطبيعي الكود يتكتب ازاي ؟!

كنت همسك كل لون لوحده اكتبه بـ ايدي وجواه احط التنسيقات بتاعته بـ ايدي يعني هـ كـ رـ العمـ لـ يـ دـ وـ يـ كـ لـ مـ رـ اـ مـ اـ دـ اـ دـ الـ \$themes هـ رـ وـ حـ اـ عـ مـ لـ اـ عـ مـ لـ يـ دـ كـ لـ مـ رـ اـ

لكن مع استخدام الـ @each الموضوع بـ اـ تـ وـ مـ اـ تـ يـ كـ وـ شـ غـ الـ بـ شـ كـ لـ دـ يـ نـ ا~ مـ يـ كـ كـ لـ مـ رـ ا~ ضـ يـ فـ \$theme تـ لـ قـ ا~ يـ ا~

هيـ تـ اـ خـ دـ اـ سـ مـ هـ وـ يـ تـ عـ مـ لـ class باـ سـ مـ هـ وـ فـ يـهـ تـ نـ سـ يـ قـ اـتـ مـ عـ بـ رـ ةـ عـ نـ لـ وـ نـ هـ هوـ

```
.red-theme .product {  
    background-color: white;  
    border-bottom: 2px solid red;  
}  
.red-theme .product .title {  
    color: red;  
    font-weight: bold;  
}  
  
.blue-theme .product {  
    background-color: white;  
    border-bottom: 2px solid blue;  
}  
.blue-theme .product .title {  
    color: blue;  
    font-weight: bold;  
}  
  
.green-theme .product {  
    background-color: white;  
    border-bottom: 2px solid green;  
}  
.green-theme .product .title {  
    color: green;
```

```

$socials: (
  "facebook": blue,
  "youtube": red,
  "github": black,
);

@each $platform, $color in $socials {
  .#$platform {
    background-color: $color;
    color: white;
    &::before {
      content: $platform;
    }
  }
}

```

```

.facebook {
  background-color: blue;
  color: white;
}
.facebook::before {
  content: "facebook";
}

.youtube {
  background-color: red;
  color: white;
}
.youtube::before {
  content: "youtube";
}

.github {
  background-color: black;
  color: white;
}
.github::before {
  content: "github";
}

```

في المثال دا مثلا انا عند **JavaScript** في الـ **Object** زي الـ **Value** و **Key**

ففوضت عن كل **Key** عندي باسم **\$color** وعن كل **Value** عندي باسم **\$platform**

وبدأت اعوض عن كل **key** بال **\$platform** وعن كل **value** بال **\$color** .. كل دا داخل المتغير الرئيسي اللي اسمه **\$social**

في اللي هيحصل في الكود كالتالي كل **\$platform** من عندي بيحل محل منصة من التلاتة اللي كتبتهم
فانا كانى بعمل **Loop** عليهم عشان يتكتبوا كلهم تحت بعض وكل منصة من دول هيكون جواها خاصية
الـ **background-color** قيمتها هتكون متوازية مع الـ **value** بتاعت كل منصة

يعنى وهو بيعمل **loop** عشان يكتب اول مرة هي Shawf الـ **facebook** بتعني اللي هي **blue** فه تكون
دي الـ **background-color**

بعدها هينتقل للـ **youtube** وي عمل نفس العملية بالظبط زي ما عملها مع
وكذلك على **github**

في الـ **before** هيكتب اسم الـ **platform** اللي هيعدى عليها وهو بيعمل **Loop**

في المثال التالي هنطبق نفس مفهوم الـ **destructuring** اللي اتعلمناه في الـ **JS**

```
$classes: "one" · 20px · red, "two" · 15px · green, "three" · 22px · blue;

@each $class, $font, $color in $classes {
  .#$class {
    color: $color;
    font-size: $font / 2;
  }
}
```

في المتغير الرئيسي اللي اسمها **\$classes** عوضت عن كل قيمة جواه بمتغير خاص بيها

فلما احط المتغير **\$class** بعد كلمة **@each** يبقى انا بعتر عن كلا من ال "one" و "two" و "three"

ولما احط المتغير **\$font** يبقى انا بعتر عن كل **font** بالترتيب زي **20px** و **15px** و **22px** وكل واحد مقسوم / 2

ولما احط المتغير **\$color** يبقى انا بعتر عن كل لون بالترتيب "red" و "green" و "blue"

وبس كدا وبدأت اعوض عنهم زي ما اتعلمنا في كل الأمثلة اللي فاتت

```
.one {
  color: red;
  font-size: 10px;
}

.two {
  color: green;
  font-size: 7.5px;
}

.three {
  color: blue;
  font-size: 11px;
}
```

راقت لي ❤️



Rand Suliman Rashwan

Tue at 3:46 PM · 🌎

...

في سنة رابعة في ولد شاطر اوي في القسم بتاعي عمل لينا جروب واتساب وكان قبل الامتحانات بيعت لينا pdf نحله فيه كل اسئلة الدكتور اللي قال عليها او هو استخلصها واسئلة كانت صعبة وكان بيجي منها وفعلا كان سبب رئيسي اني جبت A+ في معظم المواد وتقديرى التراكمي ارتفع بسبب آخر سنة

الولد ده بالصدفة النهاردة عرفت انه بعد فترة من التخرج أشتغل في مكان مكنش يحلم بيه وقالي بالحرف انا كنت بسعي لنجاحي ونجاح الناس وكنت معرفش ان ربنا هياجرني على الاثنين النهاردة ويجلبي الرزق ده كله أتمنى الناس تتعلم منه حاجة واحدة بس...(في السماء رزقكم وما توعدون) وان مفيش حد بيأخذ نصيب حد واجتهاده لما يتعلم حاجة منه ، وان فعلا ربنا هياجرك على اجتهادك لنفسك واللي ساعدتهم كمان

اذكر الله ❤️

الزموا ذكر الله ففيه العون على متابعة الحياة ❤️

حُسْنِي اللَّهُ لَا إِلَهَ إِلَّا هُوَ عَلَيْهِ تَوَكِّلْتُ وَهُوَ رَبُّ الْعَرْشِ الْعَظِيمِ

يا حي يا قيوم برحمتك استغيث أصلح لي شأني كله ولا تكلني الى نفسي طرفة عين الحمد لله بلا سبب، الحمد لله بلا طلب، الحمد لله بلا عدد، الحمد لله حتى يرضي.

أَلَا بِذِكْرِ اللَّهِ تَطْمَئِنُ الْقُلُوبُ

افتكر علينا بدعوة طيبة لوجه الله ❤️

By: Amr Elsayed

[14] - Loop – While

الـ **While Loop** نوع من أنواع الـ **loop** برضوري اللي اتعلمناهم مسبقاً

```
$start: 1;

@while $start <= 10 {
    .width-#${$start * 100} {
        width: $start * 100px;
        height: ($start * 100px) / 2;
    }
    $start: $start + 1;
}
```

عندى المتغير **\$start** بالقيمة **1** اللي هبدأ الـ **Loop** من عنده وكمان هعتمد عليه في التحقق من الشرط والانتهاء منه كمان

بدأت الـ **Loop** من خلال الـ **@while** وبعدها قولته الشرط بتاعي الـ **10 <= \$start**

طول ما الـ **\$start** اقل من او يساوي الـ **10** الشرط متحقق والـ **Loop** شغال تمام

طيب طول ما الشرط متحقق والـ **Loop** شغال هتعمل أي ؟ !

عمل **Class** اسمه **width** + قيمة الـ **\$start** اللي هي **1** مضروب في **100**

عشان اسم الـ **class** يكون **width-100** واللي بعدها والـ **Loop** شغال يكون **width-200** وهكذا

عمل الـ **width** الأول بـ قيمة الـ **\$start** اللي هي **1** مضروب في **100px** --- والـ **height** نفس الفكرة، ولكن القيمة تقسم على **2**

بعد كدا في الآخر بحط المعادلة **\$start: \$start + 1** اللي بدورها هتزود المتغير **\$start** دا من **1** لحد ما يوصل **10**

أول ما الـ **\$start** يساوي **10** الشرط كدا خلاص وصل للنهاية **وهيفي** الـ **Loop**
وهيتم انشاء **10 Classes** بناءً ع الشرط اللي انا حطيته والمعادلة اللي هتوقف الشرط دا

[15] - Create Bootstrap Grid System

هنتعلم في الدرس دا نعمل **Bootstrap Grid System** زي اللي في الـ **Grid System** كدا ... تعالوا نشوف
بس قبل ما نبدأ الأول لازم نفهم ان الـ **Bootstrap Grid System** في الـ **Grid System** مقسم إلى 12 عمود
يعني لو عندي **عناصر** عايزة لهم ياخدوا عرض الصفحة اذا كل عنصر هياخد 6 أعمدة يعني كل عنصر هياخد
50% من مساحة الألب

طيب لو عايزة اوزع تلت عناصر في الألب باستخدام الـ **grid** اذا المفروض كل عنصر ياخد 4 أعمدة
بحيث لو جمعت الـ 4 أعمدة اللي هيشغلهم 3 عناصر يكون المجموع 12 عمود

ولو عندي 6 عناصر عايزة اوزعهم في الألب يبقى كل عنصر هياخد 2 عمودين بالضبط عشان الـ 6 عناصر يشغلوا
الـ 12 عمود بالكامل مش كدا ولا أي ؟!

طيب انا هتعلم أي ؟! ... انا هعمل **For Loop** زي الـ **Grid System** باستخدام الـ **Bootstrap** بغرض انه
يعملني 12 كل **class** بيعبر عن عمود وكل ما تزيد العمدان وبالتالي يزيد الـ **width** بتاعتها

هبدأ بعد كدا استخدم الـ **Classes** دي على العناصر في الـ **HTML** عشان ابقي طبقت نفس نظام الـ **Bootstrap**

```
$grid_cols: 12;  
@for $i from 1 through $grid_cols {  
  .col-#{$i} {  
    width: percentage($i / $grid_cols);  
  }  
}
```

دلوقي انا عندي متغير اسمه **\$grid_cols** قيمته بـ 12 أي 12 عمود يعني

استخدمت الـ **@for** من 1 لـ **\$grid_cols** اللي هي بتساوي أصلًا 12 عشان يعمل **Class 12**

بعد كدا كتبت **col-#{\$i}**. عشان كل **col** يبقى جنبه رقم من الـ **Looping** اللي هيحصل بداية من **col-1** لحد **col-12**

بعد كدا كتبت خاصية الـ **width** وجواها استخدمت **percentage Method** اسمها بتحسب **النسبة المئوية**
من **العملية الحسابية** اللي هحطها جواها العملية دي بتتمثل في اني بقوله الـ **i** اول مرة هتكلون بـ 1 فاقسمها على
الـ **\$grid_cols** اللي هي 12 و الـ **method** بتاعت الـ **percentage** تحول النتيجة لنسبة مئوية

الـ **loop** مازال شغال وهيجي الدور علي ان الـ **i** هتكلون بـ 2 فهي عمل نفس الخطوات والـ **loop** مازال شغال
فهييجي الدور علي ان الـ **i** تبني بـ 3 فيعمل نفس الخطوات اللي قولنا عليها لحد ما الـ **i** توصل للـ 12 والـ **loop**
يقف

لاحظ ان كل مرة الـ `col-#{$i}`. بتببدأ من `col-1` لحد `col-12` بسبب زيادة الـ `i` من الـ **Loop**

وكمان لاحظ ان الـ `width` في كل `Col` هيقي مختلف عشان زي ما قولتلك من ثانية انه الـ `i` بتزيد قيمتها بفعل الـ **Loop** فأول مرة في `col-1` هيكون عبارة عن النسبة المئوية لحاصل قسمة الـ `i` اللي هي `1` دلوقتي على المتغير `$grid_cols` اللي هو قيمته `12`

```
.col-1 {  
    width: 8.333333333%;  
}
```

```
.col-2 {  
    width: 16.666666667%;  
}
```

```
.col-3 {  
    width: 25%;  
}
```

```
.col-4 {  
    width: 33.333333333%;  
}
```

```
.col-5 {  
    width: 41.666666667%;  
}
```

```
.col-6 {  
    width: 50%;  
}
```

```
.col-7 {  
    width: 58.333333333%;  
}
```

```
.col-8 {  
    width: 66.666666667%;  
}
```

```
.col-9 {  
    width: 75%;  
}
```

```
.col-10 {  
    width: 83.333333333%;  
}
```

```
.col-11 {  
    width: 91.666666667%;  
}
```

```
.col-12 {  
    width: 100%;  
}
```

زي ما قولت مسبقا لو عندك مثلا اكتر من عنصر في الصفحة وعايز كل العناصر دي تتوزع بحيث كل صفح يكون عبارة عن **3 عناصر** .. إذا هرروح على الـ **3 عناصر** دول احط فيها `col-4 class` الـ `col-4` بحيث ان كل عنصر يأخذ **4 أعمدة** فيكون مجموعهم `12` عمود بالضبط ومجموع الـ **3 عناصر** يكون **100%** من مساحة الألب

Assignments Week 2 From [11] => [15]

لينك التكليفات من [11] <= [15]

```
/* Assignment 1 */

@mixin arrow($direction) {
  content: "";
  position: absolute;
  border: 20px solid transparent;
  @if ($direction == 'top') {
    top: -40px;
    left: 50%;
    transform: translateX(-50%);
    border-bottom-color: red;
  } @else if ($direction == 'right') {
    right: -40px;
    top: 50%;
    transform: translateY(-50%);
    border-left-color: red;
  } @else if ($direction == 'bottom') {
    bottom: -40px;
    left: 50%;
    transform: translateX(-50%);
    border-top-color: red;
  } @else if ($direction == 'left') {
    left: -40px;
    top: 50%;
    transform: translateY(-50%);
    border-right-color: red;
  } @else {
    @error "#{$direction} Is Not Valid Direction";
  }
}

.element {
  &::before {
    @include arrow(top)
  }
}
```

```
/* Assignment 2 */

$grid_cols : 16;
$col: 1;

@while $col <= $grid_cols {
  .col-#{$col} {
    width: ($col / $grid_cols) * 100%;
  }
  $col: $col + 1;
}

/* Another Solution */
```

```
$grid_columns: 16;

@for $i from 1 through $grid_columns {
  .column-#${$i} {
    width: percentage($i / $grid_columns);
  }
}
```

```
/* Assignment 3 */

$names:
"books" 20px red 18,
"games" 30px green 18,
"dvds" 20px blue;

@each $activity, $padding, $color, $font in $names {
  .#${$activity} {
    padding: $padding;
    @if($font == null) {
      font-size: 16px;
    } @else if ($font == $font) {
      font-size: #{$font}px;
    }
    border-bottom: 2px solid $color;
    color: #444;
  }
}
```

```
/* Assignment 4 */

$init: 1;

@while ($init <= 5) {
  .circle-#${$init * 100} {
    width: $init * 100px;
    height: $init * 100px;
    border-radius: ($init * 100px) / 2;
  }
  $init: $init + 1
}
```

```
/* Assignment 5 */

@for $i from 1 through 5 {
  .circle-#${$i * 100} {
    width: $i * 100px;
    height: $i * 100px;
    border-radius: ($i * 100px) / 2;
  }
}
```

```
/* Assignment 6 */

@mixin placeholder() {
  textarea:-moz-placeholder {
    color: #777;
  }
  textarea:-webkit-placeholder {
    color: #777;
  }
  textarea:-ms-input-placeholder {
    color: #777;
  }
  textarea::-webkit-input-placeholder {
    color: #777;
  }
}

textarea#{placeholder} {
  color: #777;
}
@include placeholder();
```

```
/* Assignment 7 */

@mixin flex-center($width:400px) {
  display: flex;
  justify-content: center;
  align-items: center;
  width: $width;
}

.box-one {
  @include flex-center(400px)
}
.box-two {
  @include flex-center
}
```

لا تنسونا من دعواتكم

By: Amr El-Sayed



النملة اللي شايلة ورقة شجرة بالعافية بتعرض المساعدة ع الفيل اللي بسهولة شايل شجرة كاملة

ادْكُرِ اللَّهَ

الزَّمْوَادْكُرِ اللَّهَ فِيهِ الْعُونُ عَلَى مَتَاعِبِ الْحَيَاةِ

لَا إِلَهَ إِلَّا أَنْتَ سَبَّحَانَكَ إِنِّي كُنْتُ مِنَ الظَّالِمِينَ

أَلَا يَذِكُّرِ اللَّهَ تَطْمِئْنُ الْقُلُوبُ

لا تنسونا من دعواتكم

By: Amr El-Sayed

[16] – Function

الـ **function** فكرتها زي أي فكرة اتعلمتها في أي لغة برمجة عبارة عن **Arguments** و **Parameters** ببرضو تعالي نشوف **Very Simple Example** لشغل الـ **Function** ونشرحه بعد كدا ناخذ مثال **Advanced** شوية

ولكن قبل أي حاجة عايز اقولك الفرق بين الـ **Arguments** و الـ **Parameters** ببساطة اعتبار الـ **Parameter** كأنه اسم متغير عادي جدا والـ **Argument** هو قيمة المتغير دا بس كدا يعني هعمل متغير في الـ **function** اللي بيعبّر عن الـ **Parameter** وانا شغال همرر قيمة للمتغير دا اللي بتعتبر عن الـ **Argument**

```
@function half($size) {  
    @return $size / 2;  
}  
  
$width: 200px;  
  
.element {  
    width: $width;  
    height: half($width);  
}
```

```
.element {  
    width: 200px;  
    height: 100px;  
}
```

في المثال البسيط دا انا عملت **\$size** **Parameter** اسمها **half** ومررت جواها **Function** اسمه **half** وطلبت منها ترجعي الـ **\$size** دا لما يكون ليه قيمة، ولكن **مقسم على 2**

بعد كدا عملت متغير اسمه **\$width** قيمته بـ **200px**

وفي العنصر **.element**. كتبت خاصية الـ **width** بالقيمة الموجودة في المتغير **\$width** اللي هي **200px**

و في نفس العنصر كتبت خاصية الـ **height** و القيمة بتاعته كانت عبارة عن الـ **function** بتاعتي من خلال اسمها **half** ومررت جواها الـ **argument** بقا اللي هيحل محل الـ **\$size** وكان الـ **argument** دا هو قيمة المتغير **\$width** اللي هي **200px**

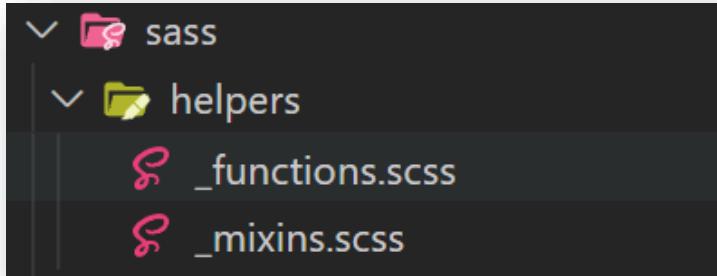
تذكر انا قولت ان الـ **Parameter** عبارة عن متغير اللي هو **\$size** و الـ **argument** هو قيمة المتغير دا اللي انا عبرت عنه من خلال الـ **\$width:200px** فهتكون النتيجة ان الـ **width** بـ **200px** و ان الـ **height** بـ **200px / 2** بـ **100px**

تعالي ناخد مثال يكون **Advanced Arguments** شوية لنفترض ان مش عارف عدد الـ **Arguments** اللي ه تكون عندي ثانية هو في حالة ممكن أكون فيها مش عارف عدد الـ **Arguments** !؟

اقولك اه لنفترض ان عندك **3 عناصر** تحت بعض في الصفحة وكل عنصر ليه **height** معين وانت عايز تعمل عنصر رابع يكون تحتهم، ولكن عايز تعرف الـ **top** بتاع العنصر الرابع دا هيبدا منين بشكل ديناميكي بناءً على مجموع الـ **height** للـ **3 عناصر** الموجودين قبل كدا

فيستخدم حاجه اسمها **reset parameters** ممكن اعوض عنها باكتر من **argument** عشان انت فعليا متعرفش الـ **3 عناصر** دول واحدين مساحة قدأي من الـ **height** معنديكش علم كل عنصر شاغل **height** قدأي عشان لو عرفت تقدر تحط العنصر الرابع دا تحتهم بالظبط او زي ما قولنا تحدد الـ **Top** بتاعه هيبدا منين بحيث يبقى تحت الـ **3 عناصر** بالظبط

بس قبل ما نبدا المثال دا تعالي نعمل بنية نضيف فيها كل الـ **functions** اللي هكتبها وبعدين ابدأ استدعى البنية دي في ملف الـ **SASS الرئيسي**



```
// Import Functions  
@use "./sass/helpers/functions" as *;
```

SASS

```
_functions.scss  
  
@function calculator($sizes...) {  
    $total: 0;  
    @each $size in $sizes {  
        $total: $total + $size  
    }  
    @return $total  
}
```

عملت اسمها **calculator Function** جواها استخدمت الـ **reset parameters** بعدن عملت متغير اسمها **\$total** قيمته **صفر** عشان هزود القيمة دي من خلال الـ **loop** اللي هعمله عملت **loop** باستخدام الـ **@each** دا كل **\$size** هيتكتب انا لسه معروفش أصلا كل **\$size** من اللي انا معروفش دا هيتكتب جوا الـ **... \$sizes** .. هات المتغير **\$total** دا بقيمه اللي هي **صفر** زود عليه الـ **\$size** اللي هيتكتب دا يعني الـ **\$total** دا **حاوية** كدا الكل الـ **Sizes** اللي هتكتب وظيفته يجيبي مجموعهم يعني بعدين قولته **@return** بقا الـ **\$total** دا بمجموعه النهائي

SASS

```
SASS - main.scss  
$width: 200px;  
  
.element {  
  width: $width;  
  height: half($width);  
  top: calculator(100px, 200px, 50px);  
}
```

CSS

```
CSS - main.css  
.element {  
  width: 200px;  
  height: 100px;  
  top: 350px;  
}
```

روحت بعد كدا الملف الـ **SASS الرئيسي** بعد ما استدعيت جواه ملف الـ **_functions.scss** استخدمت الخاصية **top** وجواه حطيت اسم الـ **function** بتاعتي ومررت جواها اكتر من **Argument** فال اللي انا عملتها دي واللي كان الغرض منها تعدي بالـ **loop** على كل **size** هكتبه جواها تجمعه مع اللي قبله وتحط المجموع دا في الـ **\$total** اللي هيرجعلي في الاخر من خلال **@return**

فبقت قيمة الـ **top** في الـ **CSS** هي عبارة عن **مجموع 100px + 200px + 50px** إذا بعد الجمع دا هيبقى الـ **top** قيمته **350px** عشان يكون **العنصر الرابع** دا محتاج **top** بالقيمة دي عشان يكون موجود تحت الـ **3 عناصر**

لا تنسونا من دعواتكم

By: Amr El-Sayed

[17] - Practice Mixin With Content

```
@mixin overlay($color, $opacity) {  
    position: absolute;  
    top: 0;  
    left: 0;  
    width: 100px;  
    height: 100px;  
    background-color: rgba($color, $opacity);  
}  
  
.black-overlay {  
    @include overlay(#000, .5)  
}
```

```
.black-overlay {  
    position: absolute;  
    top: 0;  
    left: 0;  
    width: 100px;  
    height: 100px;  
    background-color: rgba(0, 0, 0, 0.5);  
}
```

من ضمن مميزات ال **@content** هي إنك تقدر تستخدمه في عمل ال **Prefixes** مع ال **@mixin** في عمل ال **mixins** معه عظمة حرفياً

```
@mixin keyF($anim-name) {  
    @-webkit-keyframes #{$anim-name} {  
        @content;  
    }  
    @keyframes #{$anim-name} {  
        @content;  
    }  
}
```

استخدمت ال **@keyframes** في عمل ال **@mixin** وكمان بال **prefixes** بتاعته

جنب اسم ال **@mixin** بتاعي كتبت متغير عشان اعوض عنه مكان اسم ال **animation** اللي انا حابب اعمله

وبعددين استخدمت ال **@content** دى عشان اقدر احدث خصائص ال **animation** بما يتناسب مع اسمه

SASS

```
@include keyF(fade-in) {  
  from {  
    opacity: 0;  
  }  
  to {  
    opacity: 1;  
  }  
}  
  
@include keyF (go-up) {  
  from {  
    bottom: 0;  
  }  
  to {  
    bottom: 100px  
  }  
}
```

CSS

```
@-webkit-keyframes fade-in {  
  from {  
    opacity: 0;  
  }  
  to {  
    opacity: 1;  
  }  
}  
@keyframes fade-in {  
  from {  
    opacity: 0;  
  }  
  to {  
    opacity: 1;  
  }  
}  
@-webkit-keyframes go-up {  
  from {  
    bottom: 0;  
  }  
  to {  
    bottom: 100px;  
  }  
}  
@keyframes go-up {  
  from {  
    bottom: 0;  
  }  
  to {  
    bottom: 100px;  
  }  
}
```

زي ما انت شايف كدا استخدمت الـ **@mixin** في عمل الـ **prefixes** لا وكمان خليت الموضوع **динамички** من خلال استخدام الـ **@content**

فبقيت بكتب اسم الـ **@mixin** واحد اسماً **animation** وبعددين احط الخصائص بتاعتة الـ **css** وبالتالي هو بقى يترجمها في الـ **css**

يبقى انا ببساطة شديدة جداً بستخدم الـ **@mixin** عشان اعمل الـ **prefixes** بسهولة وكمان اتحكم في اسم الـ **Animation** بتاعي وبناءً على اسمه بحط الخواص بتاعتة

وتاني حاجة هي الـ **@content** اللي من خلالها بقدر استخدم أي اسم **animation** بشكل ديناميكي و الخواص اللي بحطها للـ **animation** دا بتكون هي كمان ديناميكية من اسم **animation** للثاني

حاسس انها مش واضح حالك اوي فخليني اشرحها بشكل ابسط تاني لو مفهومتش من اول مرة الكلام اللي فوق

دلوقي لو انا عندي **animation** اسمها **fade-in** وجواه بعض الخصائص موجودة في **from** و **to**

انا بقا عشان اعمل الموضوع بشكل ديناميكي جدا ومميز عملت أي ؟ !

عملت **animation** اسمه **keyF** واديته متغير اسمه **\$anim-name** المتغير دا بيعبر عن اسم أي **@ mixin** انا عايزة بعد كدا جوا ال **@ mixin** دا انا بدل ما اكتب خواص ثابتة مش هتنغير كتبت **@content**

كل دا هيأدي لأيه في الآخر ؟ !

بساطة هعمل **@include** لـ **\$anim-name** بتعني ومكان المتغير **\$anim-name** هحط أي اسم أي اسم انا عايزة وبعدين ابدأ اكتب الخصائص اللي انا عايزة في ال **animation** دا بداخل ال **from** وال **to**

ومن **animation** للثاني اقدر اغير الاسم براحتي لاني معتمد علي متغير وكمان اقدر اغير ال **content** اللي هتناسب اسم ال **animation** اللي انا كتبته دا بناءا علي ال **@conent** اللي اعتمدت عليها

فيما الموضوع ديناميكي جدا سوا في اسم ال **animation** وسواء في ال **content** او **الخواص** اللي هكتبها داخل ال **animation**

تعالي بقا غير اسم ال **go-up** **Animation** لـ **go-up** وبعدين افتح القوسين **{}** واكتب جواهم ال **content** المناسب للأسم ال **animation** دا بس كدا خلص الموضوع

```
    $anim-name
@include keyF(go-up) {
  from {
    bottom: 0;
  } to {
    bottom: 20px;
  }
}
```

```
    $anim-name
@include keyF(fade-in) {
  from {
    opacity: 0;
  } to {
    opacity: 1;
  }
}
```

[18] - Practice Create Media Queries Mixin

SASS

```
// If Condition + Mixin + Content
@mixin breakPoints ($point) {
  @if ($point == mobile) {
    @media (max-width: 767px) {
      @content;
    }
  } @else if ($point == small) {
    @media (min-width: 768px) and (max-width: 991px) {
      @content;
    }
  } @else if ($point == medium) {
    @media (min-width: 992px) and (max-width: 1199px) {
      @content;
    }
  } @else if ($point == large) {
    @media (min-width: 1200px) {
      @content;
    }
  }
}
```

انا هنا استخدمت الـ **@mixin** ومررت متغير فيها اسمه **\$point** عشان اعمل **Media** مخصصة لكل شاشة يعني لو كتبت الـ **mobile** بتنطبق الـ **small** ولو كتبت الـ **mobile** بتنطبق الـ **small** وهكذا تتطبق وهذا

فمثلاً جوا الـ **mobile** **Media** انا حددت ان الـ **\$point** اللي تخص **الموبايل** هتكون لحد **767px** فجوا الـ **media** دي انا استخدمت الـ **@content** عشان الشخص لما يحدد الميديا اللي هو عايزها يقدر يكتب فيها الخصائص والعناصر اللي هو عايزها

SASS

```
.media {  
    @include breakPoints(mobile) {  
        font-size: 15px;  
    }  
    @include breakPoints(small) {  
        font-size: 18px;  
    }  
    @include breakPoints(medium) {  
        font-size: 20px;  
    }  
    @include breakPoints(large) {  
        font-size: 22px;  
    }  
}
```

فهنا انا عوضت مكان الـ **mobile** بكلمة **\$point** في الـ **mobile** اللي كتبتها للـ **point** فبدأ يشوف الـ **mobile** اللي هي لحد **max-width: 767px**

بعدها مثلاً عوضت مكان الـ **small** بكلمة **\$point** فبدأ يشوف الـ **small** اللي كتبتها للـ **point** في الـ **small** اللي هي بداية من **max-width: 991px** **min-width: 768px**

بعدها عوضت مكان الـ **medium** بكلمة **\$point** فبدأ يشوف الـ **medium** اللي كتبتها للـ **point** في الـ **medium** اللي هي بداية من **max-width: 1199px** **min-width: 992px**

بعدها عوضت مكان الـ **large** بكلمة **\$point** فبدأ يشوف الخواص اللي كتبتها للـ **large** في الـ **large** اللي هي من **min-width: 1200px** الي ما لانهاية بقا

مكان بقا الـ **@conent** اللي انا كتبتها فوق في الـ **@ mixin** نفسه قدرت اعوض عنها في كل **media** مختلفة بخاصية الـ **font-size**

CSS

```
@media (max-width: 767px) {  
    .media {  
        font-size: 15px;  
    }  
}  
@media (min-width: 768px) and (max-width: 991px) {  
    .media {  
        font-size: 18px;  
    }  
}  
@media (min-width: 992px) and (max-width: 1199px) {  
    .media {  
        font-size: 20px;  
    }  
}  
@media (min-width: 1200px) {  
    .media {  
        font-size: 22px;  
    }  
}
```

في الفيديو دا عملت تطبيق ع الـ **media** بشكل

فقط اضغط ع الكلام

لا تنسونا من دعواتكم

By: Amr El-Sayed

[19] - The End And How To Master SASS

ملخص الفيديو عشان تكون شخص **SASS** في الـ **Master** فانت محتاج تقرأ عن المكتبات الموجودة عن الـ **@mixins** الموجودة وتقرا الكود وتفهمه

لازم تتشغل كتير بآيدك وتعمل مشاريع بالـ **SASS** عشان تتعود على العالم دا

أكتر حاجه هتفيدك في تعلم الـ **SASS** مش إنك تتطور في الـ **CSS** لااا ... إنك **تطور** في البرمجة بشكل عام لأن في الـ **SASS** أصلًا عمال تتعامل مع مفاهيم برمجية زي الـ **function** والـ **Loop** وغيرهم في الحالة دي انت بتتطور نفسك في البرمجة والبرمجة بتتطورك في استخدام الـ **SASS**

يعني هنا هتلaci مكتبة فيها كمية **Mixins** من اللي درسناها ولو كتبت في **SASS mixins GitHub** هتلaci مكتبات كتير

لينك الدروس وفيه لينكات المكتبات اللي اتفتحت في الفيديو

لا تنسونا من دعواتكم

By: Amr El-Sayed

Assignments Week 2 From [16] => [19]

لينك التكليفات من [16] <= [19]

```
/* Assignment 1 */

$num: 2;
$start: 100;

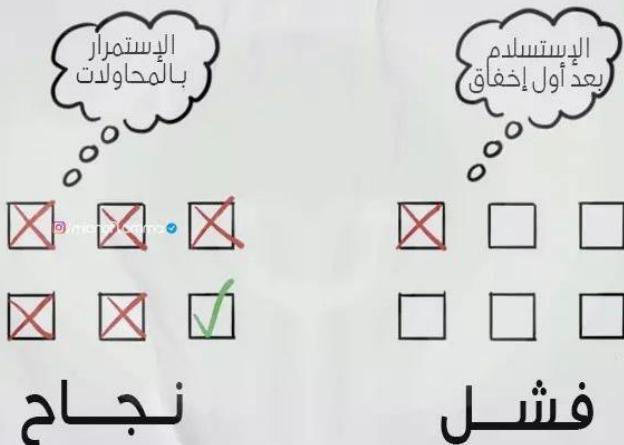
@function double($size) {
  @return $size * $num;
}

@for $i from 1 through 5 {
  @if $i == 1 {
    $start: $start;
    .box-#$start {
      width: #{$start};
      height: double($start);
    }
  } @else {
    $start: $start * 2;
    .box-#$start {
      width: #{$start}px;
      height: #{double($start)}px;
    }
  }
}
```

```
/* Assignment 2 */
// SASS Code
@function get-total($nums...) {
  $total:0;
  @each $num in $nums {
    $total: $total + $num
  }
  @return $total
}

.box {
  top: get-total(100, 50, 20);
  right: get-total(100, 50);
}
```

3- مادمت تحاول فأنت لم تفشل:



♥ اذكر الله

♥ الزموا ذكر الله ففيه العون على متابع الحياة

استغفروا الله العظيم الذي لا إله إلا هو الحي القيوم واتوب اليه.

اللهم صلّ وسلّم وردّ وبارك على نبينا محمد ﷺ.

لا إله إلا انت سبحانك اني كنت من الظالمين.

لا حول ولا قوّة الا بالله العلي العظيم.

أَلَا بِذِكْرِ اللَّهِ تَطْمَئِنُ الْقُلُوبُ

لا تنسونا من دعواتكم

Never Forget That (**Always Search**)

Tools And References

- References

- [SASS Documentation](#)
- [SASS Dev Docs](#)
- [SASS W3cub](#)
- [SASS Guideline](#)

- Tools

- [The Sass Playground](#)
- [Convert To SASS](#)
- [Compass Framework](#)

لا تنسونا من دعواتكم

By: Amr Elsayed

وبكدا الحمد لله يكون اتممنا مخلص الـ SASS وأي حاجة جاية بعد كدا
هو حاجات من خبرتي البسيطة وقرأتي لكتاباً موضوع ومشاركة بعض
أصدقائي للمعلومات فحبببت اشاركتهم في المخلص عشان ميكونش
مقتصر بس على شرح الفيديوهات ويبقى فيه فايدة أكثر

ملحوظة مهمة جداً قبل ما تبدأ الكورس دا لازم تكون دارس HTML و CSS
و اساسيات الـ JS كويش عشان تقدر تفهم الكورس دا
وتتعامل معاه علي أساس انه حاجة هتساعدك
في بناء شغلك بشكل احترافي

افتكر علينا بدعوة طيبة ❤

By: Amr Elsayed

استخدام الـ @use والـ @forward في ملفات الـ SASS

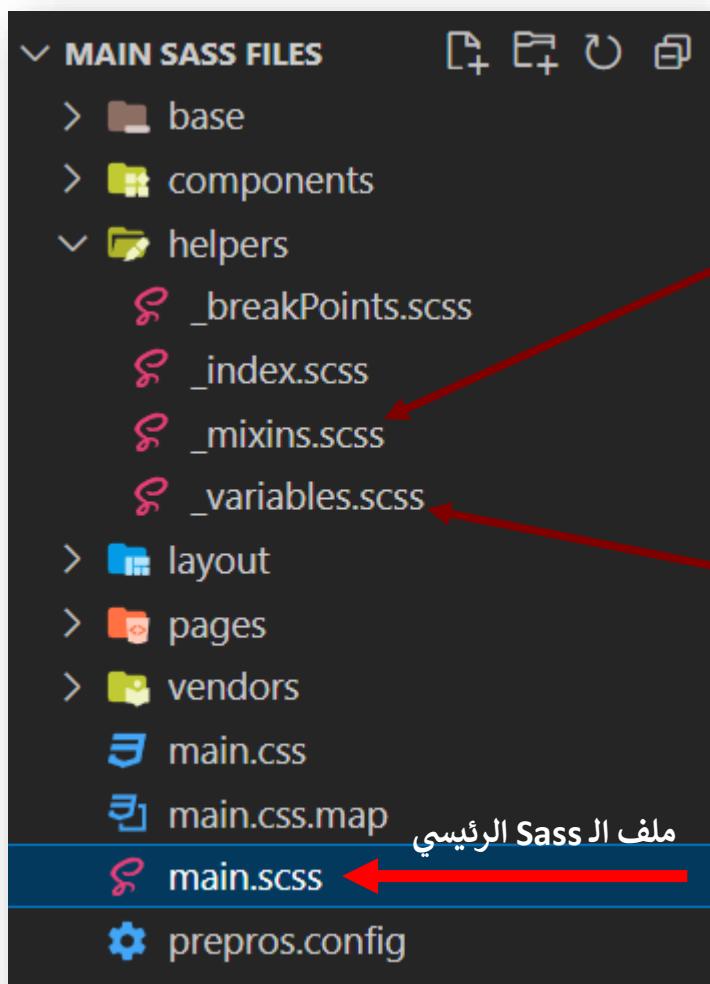
"عايز تركيزك يكون 100% في الكام صفحة الجاين دول عشان هنشرح حاجة مهمة جدا"

طبعا زي ما انا وانت عارفين ان استخدام الـ `@import` في استدعاء الملفات الفرعية الي ملف الـ `main.scss` الرئيسي هيكون **deprecated** او هيتم اهمال الـ `@import` دي ويتم الاعتماد على `@use`

أولا تعالي نوضح آلية عمل الـ `@use` واختلفها عن الـ `@import`

الـ `main.scss` دي بتشتغل **global** على الملفات بمعنى إني لو استدعيت الـ `variables` في ملف الـ `variables` في ملف الـ `@import` أقدر استخدمها في أي ملف وفي كل مكان .. عكس الـ `@use` اللي بتكون **local** وعشان أقدر استخدم اي `Variable` في اي ملف لازم استخدم `@use` لملف الـ `variables` في نفس الملف اللي عايز استخدم فيه الـ `Variables` يعني لازم استدعي ملف الـ `variables` في أي مكان وكل مكان وهنفهم النقطة دي باستفاضة مع تسلسل الشرح وهنفهم دور الـ `@forward` الجميل

انا معايا **بنية بسيطة** هبدأ أطبق عليها استخدام الـ `@use` وبعدين هقولك ازاي وليه تستخدم الـ `@forward`



ملف الـ `helpers` داخل `_mixins`

```
helpers > _mixins.scss > ...
1   @mixin flex {
2     display: flex;
3     justify-content: center;
4     align-items: center;
5 }
```

ملف الـ `helpers` داخل `_variables`

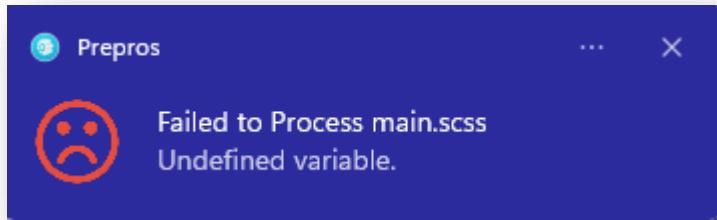
```
helpers > _variables.scss > ...
1   $red: #ff0000;
2   $green: #00ff00;
3   $blue: #0000ff;
```

ملف الـ `components` Folder داخل `_buttons`

```
components > _buttons.scss > ...
1   @use "./helpers/variables" as *;
2   .button_one {
3     color: $red;
4 }
```

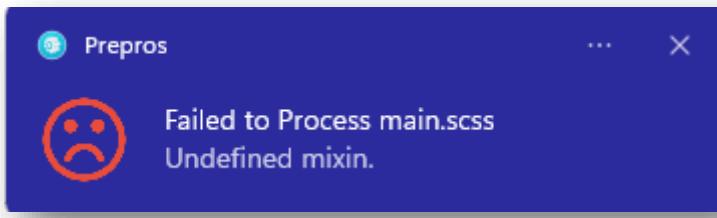
دا داخل ملف الـ `_buttons.scss` دا عشان ابدأ استخدم الـ `variables` كان لازم اروح لملف الـ `_variables` واستخدم الـ `@use` واستدعي ملف الـ `_variables`

ولو كنت روحت ملف الـ `main.scss` الرئيسي استدعيت فيه الـ `variables` **وتتجاهلت الاستدعاء** دا في ملف الـ `_buttons` كنت هلاقي مشاكل كتير من ضمنها ان ملف الـ `_buttons` مش قادر يتعرف على المتغير `$red` اللي انا بستخدمه!



فكان لازم لازم اروح استدعي ملف الـ `_variables` دا داخل ملف الـ `_buttons` نفسه

طيب أي رائيك لو قولتلك انا برضو عايزة استخدم الـ `mixin` اللي عملته فوق داخل ملف الـ `_buttons` نفس الفكرة لو روحت عملت `@use` لملف الـ `_mixins` دا في ملف الـ `main.scss` الرئيسي وبدأت استخدامه بشكل مباشر في ملف الـ `_buttons` هلاقيه برضو مش قادر يتعرف على ملف الـ `mixins` دا



ولازم برضو زي ما عملت مع الـ `variables` اروح اعمله مع الـ `mixin` واستدعيه هو كمان في ملف الـ `variables` عشان يبدأ ملف الـ `buttons` دا يقرأ الكود الموجود في ملف الـ `mixins` ويقرأ المكتوب في ملف الـ `buttons`

يعني بختصار عشان استخدم أي `button` او `variable` داخل ملف الـ `buttons` لازم استدعي الملفات دي شخصياً داخل ملف الـ `buttons`

ولو معملتش الخطوة دي وروحت استدعيتهم في ملف الـ `main.scss` على طول ملف الـ `_buttons` مش هيقدر يتعرف عليهم ... وخليل بالك إبني عامل **استدعاء** لملف الـ `_buttons` دا نفسه في ملف الـ `main.scss`

استدعاء ملف الـ `main.scss` داخل ملف الـ `_buttons` الرئيسي

A screenshot of a code editor showing the contents of a file named "main.scss". The file contains a single line of SCSS code: `1 @use "./components/buttons";`. The line number "1" is aligned with the beginning of the line, and the code is highlighted in purple.

فدلوقتي لو عندي اكتر من ملف في **Folder** الـ `helpers` دا هبقى مضطراً اروح استدعيهم كلهم تحت بعض في أي ملف عندي حابب إبني استخدم فيه ملفات الـ `helpers` دي

عشان لو استدعينهم في ملف الـ `main.scss` الرئيسي الملفات الثانية اللي هبدأ استخدم فيها **ملفات الـ helpers** دي مش هتقدر تتعرف عليهم زي تجربة ملف الـ `_buttons` كدا لو طبقتها مع أي ملف تاني

يعني م الاخر انت ممكن تستدعي ملف الـ `variables` والـ `mixins` جوا ملف الـ `buttons` او أي ملف تاني ولا إنك تستدعيهم في ملف الـ `main.scss` الرئيسي فقط يعني ممكن **تستغفي عن استدعائهم** في ملف الـ `main.scss` دا أصلاً وتستدعي بس ملف الـ `_buttons` دا هناك في الـ `main.scss`

هنا بقا يجي دور الـ `@forward` شايف اذا عمال احكيتك القصة الطويلة العريضة دي عشان أمهد لاستخدام `@forward` بعمل أي بقا في الـ `@forward`

ملف الـ `_index` داخل Folder الـ `helpers`

The screenshot shows a file tree under 'MAIN SASS FILES' with 'base', 'components', and 'helpers' folders. Inside 'helpers', there are four files: '_breakPoints.scss', '_index.scss', '_mixins.scss', and '_variables.scss'. A red arrow points from the text 'يعني عملت @forward لملف الـ _variables والـ _mixins والـ _breakPoints' to the '_index.scss' file in the file tree. To the right of the tree is the content of '_index.scss':

```
helpers > _index.scss
1  @forward "./variables";
2  @forward "./breakPoints";
3  @forward "./mixins";
```

داخل ملف الـ `_index` دا عملت `@forward` لكل الملفات الثانية اللي معايا في نفس Folder

يعني عملت `@forward` لملف الـ `_variables` والـ `_mixins` والـ `_breakPoints`

وبعد ما عملتلهم `@forward` داخل الملف دا هروح ملف الـ `_buttons` في Folder الـ `components`

اعمل `@use` لملف الـ `_index` نفسه اللي هو أصلًا **متخزن** جواه ملفات الـ `_variables` والـ `_mixins` والـ `_breakPoints`

استخدام الـ `@use` في استدعاء ملف الـ `_index` داخل ملف الـ `_buttons`

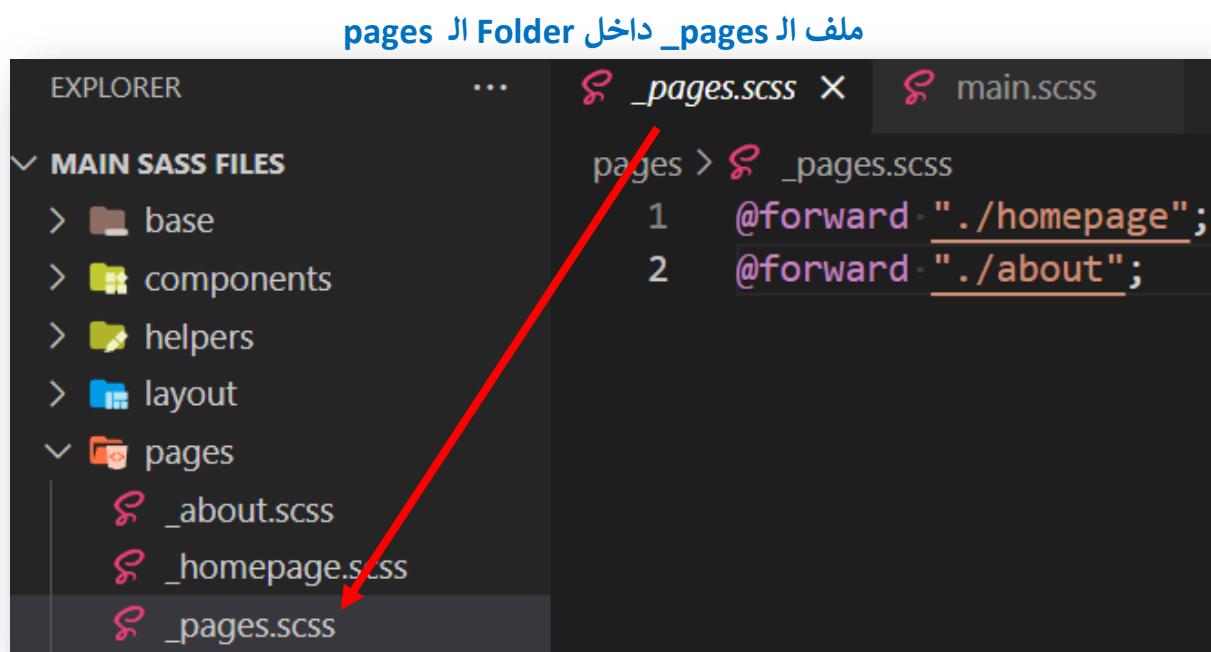
The screenshot shows the content of '_buttons.scss' which imports '_index.scss' using `@use`. The code includes three button styles: .button_one, .button_two, and .button_three, each with a different color and some flex properties.

```
components > _buttons.scss > .button_three
1  @use "../helpers/index" as *;
2  .button_one {
3    color: $red;
4  }
5  .button_two {
6    color: $green;
7  }
8  .button_three {
9    color: $blue;
10   @include flex;
11 }
```

فبدل ما اقعد في كل مكان اروحه اعمل `@use` دا **helpers Folder** داخل `_index.scss` او `_pages.scss` دا
لا انا ممكن اعمل لكل الملفات دي `@forward` في ملف واحد فقط واستدعي الملف الواحد فقط دا
في أي مكان انا حابب استخدم الملفات **المتخزن** جواه

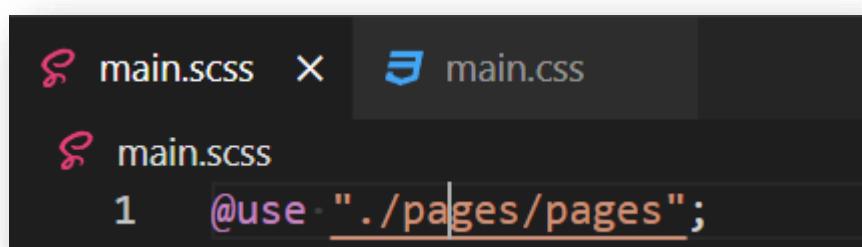
يبقى الغرض من الـ `forward` دا اني بقول لملف الـ `_index.scss` خزن جواك كل الملفات اللي هحطها هنا عشان هستخدمها في امكان تاني وبعد اذنك انا هستغير اسمك دا في اني استدعي الملفات اللي جواك

تعالي ناخد مثال تاني نفهم بيه



دلوقي انا عندي **Folder** اسمه `pages` جواه كل **الصفحات** بتاعت المشروع بتاعي يجي 10 صفحات مثلا هروح اعمل **ملف جديد** اسمه `_pages.scss` وجوا الملف دا هعمل `@forward` لكل الـ 10 صفحات اللي عندي
يبقى ملف الـ `_pages.scss` دا مخزن جواه 10 صفحات

هروح بقا ملف الـ `main.scss` الرئيسي بدل ما استدعي الـ 10 صفحات دول كلهم في **الملف الرئيسي** لا انا هستدعي ملف الـ `_pages.scss` دا فقط **المتخزن** فيه الـ 10 صفحات اصلا باستخدام الـ `@use` وبس كدا



عايز بقا في **pages** الـ **folder** اكتب شوية تنسیقات وعايز استخدم من اللي انا عملته في ملف الـ **variables** واستخدم **mixins** من ملف الـ **variable**

ملف الـ **Folder** داخل **_about**

```
main.scss × _about.scss × main.css  
pages > _about.scss > ...  
1 @use "../helpers/index" as *;  
2 .about {  
3   color: $red;  
4 }
```

شوفت أي اللي حصل ؟ روحت في ملف الـ **_about.scss** دا استدعيت ملف الـ **index** اللي هو أصلا متخزن فيه كل الملفات التانية سواء ملف الـ **mixins** او الـ **variables**

فبدل ما كل ملف اروحهحتاج استخدام فيه ملف الـ **mixins** او الـ **variables** أقوم مستدعيهم كلهم لا على أي .. انا هخزنهم في ملف جديد خالص اسمه **index.scss** مثلا ولما احتاجهم استدعي الملف دا بس لوحده كدا

وبكدا دا يكون استخدام الـ **@forward** والـ **@use** ودي النتيجة النهائية في ملف الـ **CSS**

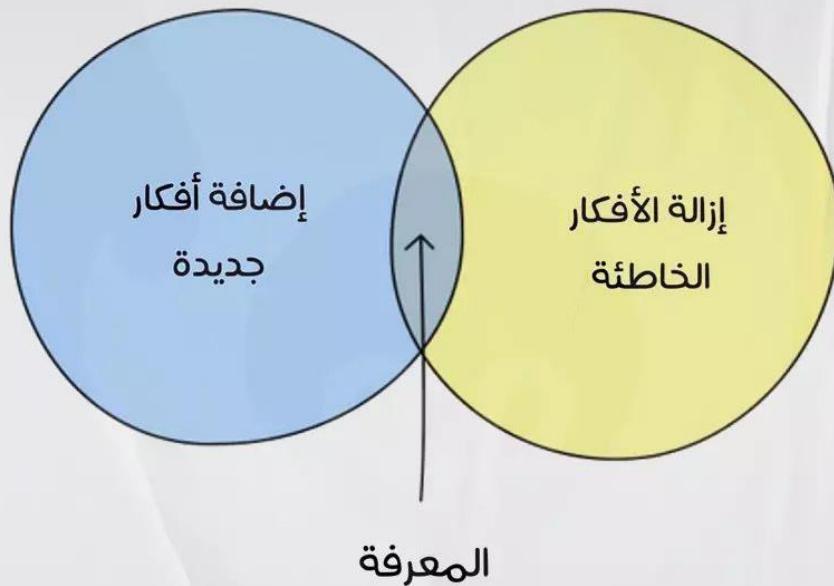
```
main.css ×  
main.css > ...  
1 .button_one {  
2   color: #ff0000;  
3 }  
4  
5 .button_two {  
6   color: #00ff00;  
7 }  
8  
9 .button_three {  
10  color: #0000ff;  
11  display: flex;  
12  justify-content: center;  
13  align-items: center;  
14 }  
15  
16 .about {  
17   color: #ff0000;  
18 }
```

ولما تعمق وتقرأ بنفسك عنهم وتبداً تطبق وستستخدم هتحسن فعلا ان الموضوع كان مهم جدا فهمه

وانه مفيش مفر لإستخدامهم عشان ابني بنية سليمة واستدعي كل الملفات بتاعتي بطريقة احترافية

لا تنسونا من دعواتكم

راقت لي ❤️



ادْكُر اللَّهَ

الزموا ذكر الله ففيه العون على متابع الحياة

سُبْحَانَ اللَّهِ وَالْحَمْدُ لِلَّهِ وَلَا إِلَهَ إِلَّا اللَّهُ وَاللَّهُ أَكْبَرُ وَلَا حُوْلَ وَلَا قُوَّةَ إِلَّا بِاللَّهِ

اللَّهُمَّ صَلِّ وَسِلِّ وَبَارِكْ عَلَيْ نَبِيِّنَا مُحَمَّدَ ﷺ

اللَّهُمَّ إِنِّي أَسْتَغْفِرُكَ وَاتُّوبُ إِلَيْكَ، لَا إِلَهَ إِلَّا أَنْتَ سُبْحَانَكَ إِنِّي كُنْتُ مِنَ الظَّالِمِينَ

لا تنسونا من دعواتكم

الطريقة الثالثة لعمل SASS باستخدام الـ Compile

زي ما اتكلمنا مسبقا ان فيه طريقتين لعمل Compile لملف الـ SASS

- الاولى كانت من خلال إضافة الـ Live SASS Compiler
- والثانية كانت من خلال اما برنامج Koala او Prepros

"عايز تركيزك يكون 100% في الكام صفحة الجايين دول عشان هنشرح حاجة مهمة جدا"

فيه طريقة ثالثة خالص تقدر تقول Advanced شوية في عمل Compile لملف الـ SASS الى CSS

والطريقة دي عن طريق npm وهي اختصار لـ Node Package Manager

طيب ما هو الـ npm هو ببساطة تقدر تقول عليه مكان موجود فيه مجموعة كبيرة جدا من الحزم والمكتبات اللي ممكن اعملها install واستخدمها في شغلي

عشان استخدم الـ npm تحتاج مسبقا لتنصيب الـ node.js وهو عبارة عن بيئه عمل بيكون جواها الـ npm اللي أقدر فيما بعد استخدمه عشان أوصل لحزمة او مكتبة معينة تساعديني في شغلي

لحد كذا كل اللي فات دا مقدمة خلينا نخش في الشرح

بساطة شديدة جدا هروح للموقع دا [node.js download](#) ومن خلاله هنزل الـ node واعمله install عادي جدا جدا زي أي برنامج بعمله install و من خلال الموقع دا تقدر تعرف طريقة التنصيب

بعد ما عملت [install](#) للـ node عشانتأكد انه فعلا نزل وكل حاجه تمام هكتب الامر **-v** كذلك للـ npm هكتب الامر **-v** وب مجرد ما الاقي الـ Versions اتكلبت يبقى كل حاجه في الـ setup تمام

سواء كتبت الامر دا في الـ VS Code بتابع terminal او كتبته في الـ PowerShell النتيجة واحدة

```
PS C:\> node -v  
v16.17.0  
PS C:\> npm -v  
8.15.0
```

TERMINAL

```
PS F:\Courses> node -v  
v16.17.0  
PS F:\Courses> npm -v  
8.15.0
```

npm init

PS F:\Courses\sass> npm init

بعد كذا بعمل امر اسمه **npm init** الغرض من الامر دا انه يعمل تهيئه **package.json** لملف اسمه **package.json** طيب يحتاج الملف دا ليه ؟! .. عشان من خلال الـ file دا بقدر اشوف أي حاجه بعملها **Setup** وكمان من خلال الملف دا بقدر اسطب الـ **SASS** واعمله **Watch** عشان يتحول الـ **code** اللي فيه لـ **CSS**

بعد ما تعمل الامر **npm init** دا دوس Enter لحد ما توصل لكلمة yes/no اكتب yes او اكتب no بس
بعدها هيتم انشاء ملف الـ **package.json**

package.json

ودا اللي بيكون في ملف الـ **package.json**

```
package.json
package.json > ...
1  {
2    "name": "sass",
3    "version": "1.0.0",
4    "description": "",
5    "main": "main.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "",
10   "license": "ISC"
11 }
```

الخطوة اللي بعد كدا انا محتاج اسطب الـ **SASS** نفسه بقا اسطبه ازاي ؟!

من خلال موقع [npm](#) هفتحه وابحث في البحث عن SASS هلاقي الصفحة دي مثلا **node-sass** بتعطي ممكن بدل **install** بتاعي احطه في الـ **Terminal** بعدين اخده **Copy** وجواها امر الـ **Setup** هاخده **Copy** بعدين احطه في الـ **Terminal** اكتب حرف i بس

Install

npm install node-sass



npm i node-sass

وبكدا أكون عملت لـ **SASS Setup**

هروح بقا لملف الـ **Package.json** اللي انا كنت عملته تهيئه وهعدل الجزء بتاع الـ **scripts** هكتب فيه الامر دا

```
"scripts": {
  "sass" : "node-sass -w main.scss -o css/ --recursive"
},
```

والجزء دا أي الغرض منه ؟!

الغرض منه إنك لما بتعمل **Run** لأي حاجة موجودة في الـ **npm** يروح لملف الـ **Package.json** يبص ع الـ **scripts** اللي اسمه **object** دا وينفذ الحاجات اللي جواه

انا في حالي انا داخل جزء الـ **sass** كتبت **scripts** أي اسم انت عايزو ممكن تكتبه عادي بعدها كتبت الامر **node-sass Compile** لملف الـ **SASS** بتاعي ويحوله **CSS** اللي هو اللي بي عمل

فاكر من سطرين فاتو بالظبط قولتلك ان أي حاجه بكتبها جوا الـ **scripts** دا لما باجي استخدم الـ **npm** بيروح يبعض ع اللي مكتوب هناك وينفذه وهو دا بالظبط اللي هيحصل

انا بقوله روح شوف الامر اللي انا كاتبه ونفذه والامر دا عباره عناني بقوله:

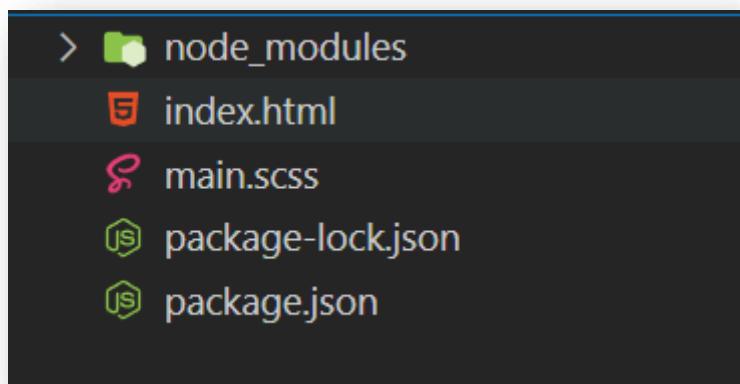
-1- اعمل **watch** لملف عندي اسمه **-w <= main.scss**

-2- وبعدين الـ **-o <= output** هيتم من خلال إنك تعمل فولدر جديد اسمه **css** وجواه تحط ملف الـ **css** اللي أصلًا هيكون عباره عن ترجمة لملف الـ **SASS** ولاحظ الـ **/** المحظوظه بعد **css** عشان اعرفه ان دا هيكون اسم **folder**

-3- اما الـ **--recursive** دي فهي **flag** الغرض منها إني أقوله وانت بتترجم ملف الـ **SASS** لـ **CSS** خلي كل شغلك جوا **فولدر الـ CSS** اللي هيتم انشائه مهمًا يحصل متخرجش منه او تعمل حاجة برا **فولدر الـ CSS** دا يعني الـ **output** اللي هيجي يتحط في الفولدر دا فقط

طبعا اللي هيحصل بقا انه هي Shawf الامر **-w <= watch** قبل ملف الـ **main.scss** وبعدين ي Shawf الامر **-o <= output** اللي هي قبل **فولدر الـ CSS** اللي انا قولته انه يعمل الفولدر دا وبعدين يعمل **Compile** لملف الـ **SASS** ويحط ملف الـ **CSS** من نفسه بنفس اسم ملف الـ **SASS** بالظبط داخل **فولدر الـ CSS**

دي الملفات اللي عندي اعمل ما اعمل اخر خطوه ... لسه فاضل اخر خطوه اني اشغل بقا الـ **compiler**



اخر خطوه بقا هكتب الامر دا في الـ **terminal** عشان اعمل **run** وتبداً عملية تحويل الملف من **SASS** لـ **CSS**

```
PS F:\Courses\SASS> npm run sass
```

npm run sass

للحظ إني عملت run للاسم اللي أنا كنت كتبته في ملف الـ **package.json** داخل الـ

وبالمناسبة انت ممكن تحط أي اسم تحبه عادي وتعمل للاسم دا

قصد الاسم اللي هنا دا

```
"scripts": {  
    "sass" : "node-sass -w main.scss -o css/ --recursive"  
},
```

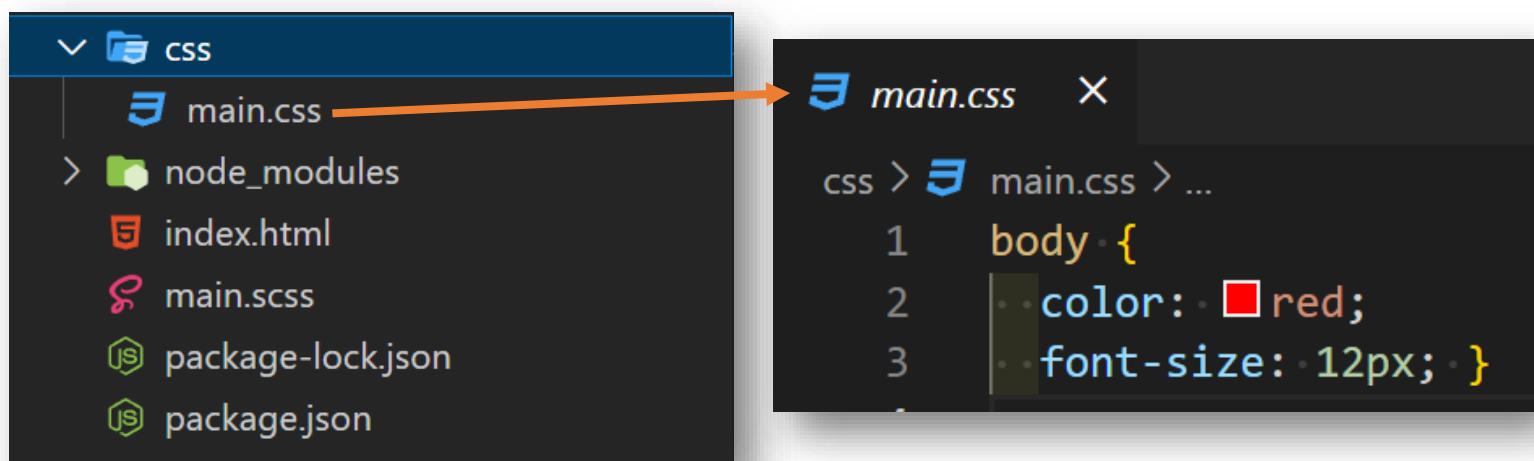
بعد ما عملت run عشان يبدأ الـ **SASS** روحت ملف الـ **Compiling** كتبت كود بسيط

```
main.scss  X  
main.scss > body  
1   body {  
2     color: red;  
3     font-size: 12px;  
4 }
```

بعد ما كتبت في الـ **SASS** رد عليا انه ترجم الـ **SASS** دا عمل **فولدر** جديد اسمه **css** وحط جواه ملف الـ **main.css**

```
=> changed: F:\Courses\SASS\main.scss  
Rendering Complete, saving .css file...  
Wrote CSS to F:\Courses\SASS\css\main.css
```

وبهكذا شكل لو روحت افتح ملف الـ **main.css** هلاقيه ترجم ملف الـ **SASS** لـ **CSS** زي ما انا طلبت بالضبط



وبكدا أكون استخدمت طريقة تالتة مختلفة وبالمناسبة في طرق تانية كتير جدا

راقت لي ❤️



Ghareeb Elshaikh

Sep 23 at 12:04 AM · 🇪🇬

...

مفيش علم بيجي بالساحل، هتصبر وهتتحاول مرة واتنين
وتلاتة هتوصل، لكن هتسسلم من أول محطة مستحيل تعرف
تكلمل ف أي حاجة.
ومن لم يذق مر التعلم ساعة، تجزع ذل الجهل طول حياته.

ادْكُر اللَّهَ ❤️

الزموا ذكر الله ففيه العون على متابعة الحياة ❤️

اللهم لك الحمد حمداً طيباً كثيراً مباركاً فيه كما ينبغي لجلال وجهك ولعظيم سلطانك يا رب

اللَّهُمَّ إِنِّي أَسأَلُكَ عِلْمًا نافعًا، ورِزْقًا طَيِّبًا، وعَمَلاً مُتَقَبِّلًا.

اللَّهُمَّ أَعِنِّي عَلَى ذِكْرِكَ، وَشُكْرِكَ، وَحُسْنِ عِبَادَتِكَ.

اللهم صلّ وسلّم وبارك على نبينا محمد ﷺ.

أَلَا بِذِكْرِ اللَّهِ تَطْمَئِنُ الْقُلُوبُ

افتكر علينا بدعوة طيبة لوجه الله

Never Forget That (**Always Search**)

By: Amr Elsayed

SASS Architecture Structure GitHub

في اللينك دا هتلaci ساس-بنية-بنية Sass-Architecture-Structure مبني من الصفر وممكن تعدل فيه وتضيف اللي انت عايزو وتحذف اللي مش محتاجة

الـ SASS ملوش قاعدة ثابتة في بناء البنية يعني البنية اللي في الـ Repo فوق ممكن تلaciها مختلفة في أماكن تانية ... انت نفسك ممكن تعمل بنية خاصة بيك وتشتغل عليها وتطورها زي ما قولتك مفيش قاعدة ثابتة في بناء بنية الـ SASS

صورة للشكل العام من الـ Structure

Sass-Architecture-Structure

```
sass/
|
|- base/
|   |- _reset.scss      # Reset All Elements
|   |- _container.scss  # Main Container For Layouts
|   |- _typography.scss # Typography rules
|   |- _base.scss       # @forward All Files in base Folder
|   ...
|   ...
|
|- components/
|   |- _buttons.scss    # Buttons
|   |- _components.scss # @forward All Files in components Folder
|   ...
|   ...
|
|- helpers/
|   |- _variables.scss  # Sass Variables
|   |- _functions.scss   # Sass Functions
|   |- _mixins.scss      # Sass Mixins
|   |- _breakPoints.scss # To make layouts Responsive
|   |- index.scss        # @forward All Files in helpers Folder
|   ...
|   ...
|
|- layout/
|   |- _header.scss     # Header
|   |- _navigation.scss # Navigation
|   |- _grid.scss        # Grid system
|   |- _footer.scss      # Footer
|   |- _layout.scss      # @forward All Files in layout Folder
|   ...
|   ...
|
|- pages/
|   |- _homepage.scss   # Home specific styles
|   |- _contact.scss     # Contact specific styles
|   |- _about.scss       # Contact specific styles
|   |- pages.scss        # @forward All Files in pages Folder
|   ...
|   ...
|
|- vendors/
|   |- _bootstrap.scss  # Bootstrap
|   |- _fontawesome.scss # fontawesome Library
|   |- _normalize.scss   # normalize
|   ...
|   ...
|
`- main.scss           # Primary Sass file
```

♥ عبارات تحفيزية ♥

CONSISTENCY VS INTENSITY

INTENSITY



CONSISTENCY



قليل دائم خيرٌ من كثير منقطع .. وحقيقي والله أحب الاعمال الى الله ادومها وان اقل يعني إنك تستمر ف تعلم حاجه بكميات بسيطة لفترة طويلاً جداً
افضل بكثير من انك تتعلم جزء كبير جداً من حاجة كل كام شهر
داوم على القليل افضل من الكثير المنقطع ❤️

سبحانك اللهم وبحمدك أشهد ان لا إله الا انت أستغفرُك واتوب اليك.

لا تنسونا من دعواتكم

Never Forget That (**Always Search**)

By: Amr Elsayed

الخاتمة وشويه فضفضة ❤

❖ في البداية وبفضل الله سبحانه وتعالي أتوجه بجزيل الشكر والامتنان والعرفان بالجميل للبشعهندس **أسامي الزيرو** اللي بفضل الله علينا ثم مجده مجهوده مكنش **Text** دا تم ربنا يجازيك كل خير ويكرمك ويتوسع رزقك ويملا حياتك خير وبركه وسعادة ويرزقك الفردوس الأعلى انت وكل احبابك

لينك القناة

لينك البلاي ليست الخاصة بالـ SASS

خطة الدراسة الـ SASS

جميع أ קוד دروس الـ SASS موجودة هنا

جميع تكليفات الـ SASS موجودة هنا

❖ استكمالا للمشوار وبعد تلخيص الـ **HTML** و الـ **CSS** و الجزء الأول من الـ **JS** .. اتماما تلخيص الـ **SASS** بفضل الله ارجوا من الله عز وجل ان يكون ذا نفع للجميع .. و ارجوا من الله عز وجل ان يجعل هذا العمل خالصا لوجهه الكريم وان يجعله في ميزان حسناتنا جميعا

❖ بعتذر عن وجود اي أخطاء املائية او أخطاء في أي كود فالكمال لله وحده واتمني لو لقيت اي خطأ سواء في الشرح او حل التكليفات تتواصل معايا وتبلغني بيها

❖ خصصت في الملخص دا تلت أجزاء .. الجزء الأول **لذكر الله** عشان متخليش حاجه ابدا تشغلك عن ذكر الله اللي قادر ينجيك ويفتح لك ابواب الرزق ويرزقك ويسهل عليك الفهم .. و جزئين **العبارات التحفizية** و **جمل وعبارات راقت لي** لكسر روتين القراءة والمذاكرة

❖ لو تعذر عليك فهم حاجة **حاول مرة واتنين وتلاته** لو موصلتش لحاجة راسلكي وان شاء الله أقدر اساعدك

❖ حابب أتوجه بالشكر لصديقى **أحمد مصطفى** لمساهمته في شرح استخدام الـ **@use** و الـ **@forward** و الـ **Sass-Architecture-Structure** ووضع بعض الملاحظات بالإضافة لإنه ساهم بشكل كبير جدا في الـ

❖ لأي حد هيوصله الكلام دا مش طالب منك غير طلب واحد وامانة عليك لازم تنفذه و هو إنك تفتكرني بدعة طيبة لوجه الله

Never Forget That Always Search ❖

[LinkedIn Account](#)

[GitHub Account](#)

[Codepen Account](#)

[Twitter Account](#)

[Mail Me](#)

[Facebook Account](#)

افتكروني بدعة طيبة لوجه الله واشوفكم ان شاء الله مع حاجه أفضل وأفضل في الملخصات الجاية

شکرا لكم ❤

By: Amr Elsayed