تطوير متجر إلكتروني

الوحدة 1: مدخل:

الدرس 1: التعرف على المسار:

الدرس 2: تهيئة المشروع:

الدرس 3: التعريف ﺒ npm و Node.js:

- ابحث في المتصفح عن مدراء الحزم أو package manager واقرأ عنها قليلًا.

- مدير الحزم: هو نظام او برمجية تدير كل البرمجيات الخارجية الأخرى في المشاريع، أي لا يلزمنا في كل مرة تنزيل مكتبات وبرمجيات معينة في كل مشروع بل يكفي أن تكون تلك البرمجية مدعومة من قبل مدير الحزم الذي يتم التعامل معه ثم التنزيل بواسطتها وتلقائيا يتم إضافتها في المشروع، يساعد مدير الحزم على ضبط توافق البرمجيات في المشاريع ويساعد على حذف البرمجيات غير المستخدمة في المشروع.

أشهر مدراء الحزم:

* pip يستخدم مع لغة Python.
* npm يستخدم مع لغة JavaScript.
* composer يستخدم مع لغة PHP.

- في تطوير واجهات الويب تستخدم مع html وcss هي js والتي تستخدم من خلال بيئتها التشغيلية node.js، أي يناسب هنا npm، راجع عنها [هنا](https://academy.hsoub.com/programming/javascript/nodejs/%D8%AF%D9%84%D9%8A%D9%84%D9%83-%D8%A7%D9%84%D8%B4%D8%A7%D9%85%D9%84-%D8%A5%D9%84%D9%89-%D9%85%D8%AF%D9%8A%D8%B1-%D8%A7%D9%84%D8%AD%D8%B2%D9%85-npm-%D9%81%D9%8A-nodejs-r1465/)..

- راجع هذا [المقال](https://academy.hsoub.com/programming/workflow/%D9%85%D9%82%D8%A7%D8%B1%D9%86%D8%A9-%D8%A8%D9%8A%D9%86-%D9%85%D8%AF%D9%8A%D8%B1-%D8%A7%D9%84%D8%AD%D8%B2%D9%85-npm-%D9%88-yarn-r1597/) أيضًا.

- لتنزيل npm يجب تنزيل بيئة العمل node.js لاستخدام npm وهي مدير حزم خاص بهذه البيئة، ابحث عن هذه البيئة في مقالات الأكاديمية. وراجع أيضًا في موسوعة حسوب من [هنا](https://wiki.hsoub.com/Node.js)..

- راجع هذا [المقطع](https://www.youtube.com/watch?v=NseEi4a8_rY) من أكاديمية حسوب لاستخدام npm..

- node.js: هي بيئة تطوير أو منصة تسمح لنا بالتعامل مع نظام التشغيل من خلال js، يمكننا من خلالها قراءة الملفات الموجودة على الجهاز المحلي وتعديلها وإدارتها؛ وإنشاء خوادم محلية، والتعامل مع قواعد البيانات وغيرها، وأهم ما يميزها هي السرعة لأنها تستخدم محرك جافا سكربت V8 الذي طورته جوجل.

الدرس 4: التعريف ﺒ Webpack:

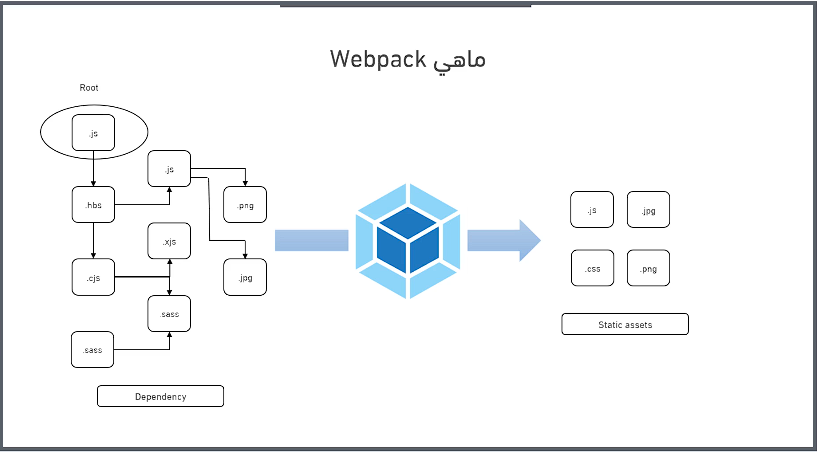
- لو بحثنا عن الأدوات التي يديرها npm نجد أن جميعها ذات هدف أساسي وهو أتمتة المهام الشائعة خلال تطوير واجهات الويب؛ مثال: عند القيام بتعديل أي شيء في ملفات صفحة الويب علينا إعادة تحميل الصفحة يدويًا لكي تظهر تلك التعديلات لكن باستخدام أدوات البناء أصبح مهمة التحديث تتم تلقائيًا،

من المميزات الأخرى التي توفرها تلك الأدوات تقليص حجم الملفات المستخدمة في عرض الصفحة وبالتالي زيادة سرعة تحميل الصفحة وتسمى الميزة minification وهي مفيدة جدًا ولها خصوصية في الاستخدام مثل هذه [الأداة](https://www.minifier.org/) يقوم بتقليص حجم الأكواد وجعلها في سطر واحد ويؤدي ذلك بشكل عام إلى زيادة سرعة الموقع وهناك العديد من الأدوات مثلها،

وكما هو معروف أنه يصعب قراءة الملفات المقلصة لذلك الحل الذي جلبته أدوات البناء لهذه المشكلة هو إبقاء الملفات التي يتعامل معها المطور غير مقلصة حتى يسهل عليه قراءتها والتعديل عليها مع إنشاء مجلد جديد منفصل يحتوي النسخة المقلصة من الملفات وتلك الملفات المقلصة هي التي يتعامل معها الخادم وهكذا تقوم أدوات البناء تلك بأتمتة المهام بشكل أسرع.

- جرت العادة أن نسمي مجلد العمل والتطوير بالمجلد المصدري src(source) أي مجلد البناء أو مجلد الملفات المقلصة التي تنشئها أداة البناء وبالنسبة للملفات التي تنشئها أداة البناء جرت العادة أن يسمى ﺒ dist.

- ابحث في المتصفح عن webpack واقرأ عنها.



- [موقع ويب باك](https://webpack.js.org/)..

- تلخيص أحد المدربين من التعليقات: الويب باك (Webpack) هو أداة لتجميع وتعبئة ملفات الويب ولكن دوره ليس فقط تقليص عدد الملفات ودمجهم في ملف واحد بل يوفر العديد من المميزات الأخرى ولنري معا ما هي:

تجميع الملفات:

الويب باك يقوم بتجميع ملفات JavaScript، CSS، الصور وأي نوع آخر من الأصول (assets) في ملفات مجمعة. هذه العملية تعرف بـ "التجميع"(Bundling).

يمكن أن يقوم بدمج ملفات JavaScript المختلفة في ملف واحد أو عدة ملفات حسب الإعدادات مما يقلل من عدد الطلبات إلى الخادم (HTTP requests) وبالتالي يساهم في تحسين أداء تحميل الصفحة وسرعة الخادم لديك.

التقليص (Minification).

هي عملية تقليص حجم الملفات عن طريق إزالة المسافات البيضاء والتعليقات والأكواد المكررة وتقليص الأسماء مثل المتغيرات والدوال إلى أقل عدد من الأحرف الممكنة ويساعد ذلك على تحسين سرعة تحميل الصفحة وأيضا صعوبة قراءة الكود الخاص بك وهذا يوفر أمانا لموقعك.

التوزيع (Code Splitting).

الويب باك يدعم تقسيم الكود (Code Splitting) مما يعني إمكانية تقسيم الكود إلى أجزاء أصغر يتم تحميلها فقط عند الحاجة إليها بدلا من تحميل جميع الأكواد والتي لن تحتاجها جميعها وهذا يساعد على تحسين أداء التطبيق بشكل كبير.

إدارة التبعيات:

يقوم الويب باك بتحليل الأكواد الخاصة بك ويقوم بمراقبة التبعيات بين الملفات والتأكد من تحميلها بالترتيب الصحيح فمن الممكن اعتماد ملف لديك على ملف أخر وهذا الملف يعتمد على ملف غيره ولذلك يتوجب تحميلهم بالترتيب وإلا فلن يعمل وويب باك يتأكد من هذه النقطة وهذه من قوة ويب باك التي تميزه من وجهة نظري.

تلخيصا لما سبق يوفر ويب باك لك تحسين الأداء وتقليل عدد الطلبات وحجم الملفات مما يساهم في تحسين سرعة تحميل الصفحة وتجربة المستخدم ويجعل إدارة الأصول والتبعيات أسهل وأكثر تنظيما.

الدرس 5: تثبيت Node.js:

- متغيرات البيئة: هي مجموعة من المتغيرات تؤثر على سلوك قيمها على سلوك البرامج أثناء تنفيذها يوجد لكل منها اسم وقيمة، متغير البيئة الذي اسمه path هدفها تحديد المسارات التي يبحث فيها النظام عن الأوامر الخاصة بالبرامج أو التقنيات فإذا كان هناك أوامر معينة مثل أوامر نود موجودة في المسار ووووووو لم أفهم ابحث عن متغيرات البيئة.....

الدرس 6: تثبيت Webpack:

- اعلم أن جميع الأدوات والتقنيات التي أنشئت للعمل ضمن بيئة node.js مثل webpack تخزن في مكان واحد وهو node.js package manager أو npm الذي تعرفت عليه سابقًا وبهذا نشبه npm بمتجر التطبيقات.

- ابحث عن "رخص البرمجيات حسوب" في المتصفح واقرأ..

- الأمر npm init يُستخدم لتهيئة مشروع Node.js جديد وإنشاء ملف تكوينpackage. json للمشروع. هذا الملف يحتوي على معلومات هامة حول المشروع مثل اسم المشروع، الإصدار، الكاتب، التراخيص، والأوامر المختلفة التي يمكن تشغيلها.

باستخدام الأمر npm init، سيُطلب منك الإجابة على سلسلة من الأسئلة لتخصيص ملف package. json وفقًا لاحتياجات مشروعك. إذا كنت ترغب في تخطي الأسئلة التفاعلية، يمكنك استخدام npm init -y لإنشاء ملفpackage. json بإعدادات افتراضية.

- الأمر npm install --save-dev webpack@5.75.0 يُستخدم لتثبيت مكتبة Webpack كاعتماد تطويري في مشروعك دعنا نفصل الأمر قليلاً:

* npm install: هذا الجزء من الأمر يُخبر مدير حزم Node (npm) أنك تريد تثبيت حزمة معينة، ويمكن اختصارها إلى npm i.
* --save-dev: هذه العلامة تُشير إلى أن الحزمة التي تقوم بتثبيتها يجب إضافتها كاعتماد تطويري (devDependency) في ملف package. json هذا يعني أن الحزمة ستكون ضرورية فقط أثناء تطوير المشروع وليس في الإنتاج، ويمكن اختصارها إلى -D.
* --save: تُستخدم لتثبيت حزمة معينة وإضافتها كاعتماد أساسي (dependency) في مشروعك. هذا يعني أن الحزمة ستكون ضرورية لتشغيل المشروع في كل من بيئة التطوير والإنتاج. عندما تستخدم --save، ستُضاف الحزمة إلى قسم dependencies في ملف package. json.

بإستخدام npm install --save package-name، سيتم تثبيت الحزمة المحددة وإضافتها كاعتماد أساسي.

يجدر بالذكر أنه في الإصدارات الحديثة من npm، العلامة --save ليست ضرورية لأن npm يقوم بذلك بشكل افتراضي عند تثبيت أي حزمة بدون تحديد --save-dev أو -D.

الفرق بين --save و --save-dev أن --save-dev تستخدم من أجل الواحدات التي تلزمنا في بيئة التطوير فقط وأما --save تستخدم من أجل الواحدات التي تلزمنا في بيئة الإنتاج فقط.

* webpack@5.75.0: هذا الجزء يُحدد اسم الحزمة التي تريد تثبيتها (Webpack) وإصدارها المحدد (5.75.0).

بمجرد تشغيل هذا الأمر، سيقوم npm بتثبيت إصدار Webpack 5.75.0 وإضافته إلى قسم devDependencies في ملف package. json.

- بعد تثبيت webpack سنحتاج إلى تثبيت أداة webpack-cli أي واجهة سطر أوامر ويب باك؛ فائدتها أنها تمكننا من تنفيذ الإعدادات الخاصة بويب باك من خلال مجموعة من الأوامر، الأمر npm i -D webpack-cli@5.0.1: سيقوم npm بتثبيت الإصدار 5.0.1 من Webpack CLI وإضافته كاعتماد تطويري في قسم devDependencies بملف package. json.

- هناك بيئتان لأي مشروع تطوير ويب باك؛ البيئة الأولى: هي بيئة التطوير development يكون فيها المشروع أثناء فترة إنشاءه وتكون الشيفرة حينها مقروءة ومفهومة للمطورين، البيئة الثاني: هي بيئة الإنتاج يكون فيها المشروع عندما يتم نشره بالشكل النهائي كمنتج للزوار وغالبًا تكون الشيفرة مختصرة ومقلصة وغير مقروءة، في [توثيق ويب باك](https://webpack.js.org/configuration/mode/) الرسمي أن ويب باك لديها 3 أوضاع أو modes يمكن تعيينها للمشروع..

الدرس 7: إنشاء ملف الإعداد webpack.config.js:

- لتتمكن ويب باك من تحزيم الملفات - غير ملفات js - يجب إنشاء ملف جديد باسم webpack.config.js، ولمعرفة كيفية إضافة الإعدادات فيه اتجه إلى [التوثيق الرسمي لويب باك](https://webpack.js.org/concepts/configuration/#introductory-configuration)، ننسخ الكود ونلصقها في الملف الجديد، قام المدرب بوضع entry على شكل object ووضع فيها app: وقيمته ملف الروت والذي هو ملف js؛ وفي output في path تركها كما هي - وهي ملف dist - وfilename قام بتغييرها إلى main.js.

- شرح الأكواد لملف webpack.config.js:

الصورة تعرض ملف تكوين Webpack يسمى webpack.config.js هذا الملف يستخدم لتحديد كيفية تجميع ملفات المشروع باستخدام Webpack هنا شرح مفصل لكل جزء من الأكواد:

const path = require(‘path’);

هذا السطر يستورد مكتبة path المدمجة في node.js، والتي توفر أدوات للتعامل مع مسارات الملفات والدلائل.

module.exports = {

هذا السطر يبدأ بتصدير كائن يحتوي على إعدادات webpack.

mode: ‘development’,

هذا السطر يحدد وضع التجميع ليكون في وضع التطوير. هذا يعني أن webpack سيقوم بتجميع الملفات بطريقة تسهل عملية التطوير، مثل تضمين معلومات التصحيح.

app: ‘./src/index.js’

هذا الجزء يحدد نقطة الدخول للتطبيق. هنا الملف الموضوع بين القوسين هو الملف الرئيسي الذي سيبدأ منه webpack عملية التجميع.

output: {

path: path.resolve(\_\_dirname, ‘dist’),

filename: ‘main.js’,

},

هذا الجزء يحدد إعدادات الإخراج: path يحدد المسار الذي سيتم فيه حفظ الملفات المجمعة. يستخدم الدالة التي فيه لتحديد المسار المطلق لمجلد dist داخل الدليل الحالي، filename يحدد اسم الملف المجمّع النهائي.

الدرس 8: إعداد Webpack لتحزيم ملفات HTML:

- بشكل عام لو بحثت في المتصفح نتائج البحث أغلبها عبارة عن أشياء اسمها loaders محملات وأشياء أخرى اسمها plugin الإضافات. في حالتنا هنا سنحتاج محمل html loader وhtml webpack plugin لتحزيم ملفات html إلى مجلد dist.

- توثيق إضافة [HtmlWebpackPlugin](https://webpack.js.org/plugins/html-webpack-plugin/) وطريقة استخدامه، راجع configuration منها..

- توثيق محمل [HtmlLoader](https://webpack.js.org/loaders/html-loader/) وطريقة استخدامه..

- ابحث عن الأكواد عبر الذكاء الاصطناعي إن احتجت..

- عندما نستخدم Webpack لتحزيم ملفات المشاريع مثل ملفات CSS وJavaScript سنواجه تحد يتعلق بكيفية إدارة الملفات التي يقوم المتصفح بتخزينها مؤقتا وهذه العملية تعرف بالتخزين المؤقت (Cache)، حيث يقوم المتصفح بحفظ نسخ من الملفات المستخدمة في الموقع لتسريع تحميل الصفحات عند العودة إليها.

وعندما يقوم المستخدم بزيارة موقع ويب ما يقوم المتصفح بتحميل الملفات مثل style.css وscript.js ويحتفظ بنسخة منها في الذاكرة المؤقتة وإذا تم تحديث هذه الملفات على الخادم، فإن المتصفح قد يستمر في استخدام النسخة القديمة المخزنة، مما يعني أن المستخدم لن يرى التحديثات الجديدة حتى يقوم بتحميل الصفحة مجددا وهذا قد يؤدي إلى تجربة مستخدم سيئة.

ولحل هذه المشكلة يقوم Webpack بتسمية الملفات بطريقة تعرف بالتسمية المجزأة وهذا يعني أنه عند تحزيم الملفات، يضيف Webpack سلسلة من الأحرف الفريدة إلى اسم كل ملف.

الدرس 9: إعداد Webpack لتحزيم ملفات CSS:

- علينا تثبيت إضافة ومحمل css.

- توثيق [css-loader](https://webpack.js.org/loaders/css-loader/).

- توثيق [MiniCssExtractPlugin](https://webpack.js.org/plugins/mini-css-extract-plugin/).

- تفاصيل مذكورة في الدرس راجعها..

- توثيق [CssMinimizerWebpackPlugin](https://webpack.js.org/plugins/css-minimizer-webpack-plugin/).

- فيه قروشة واجد.. راجعهن..

الدرس 10: تهيئة خادم Webpack المحلي:

- توثيق [DevServer](https://webpack.js.org/configuration/dev-server/).

- فيه قروشة كثيييييرة عد المقطع..

- وضح المدرب من الدقيقة 9:50 إلى 13:13 تقريبًا عن المنفذ port وفيم يستخدم..

- أعلى رقم محدد للمنافذ هو 65535.

- أمر سطر الأوامر: netstat -a لإظهار جميع المنافذ ports المحجوزة في الحاسوب.

- خاطرة: لا شيء مستحيل في عالم البرمجة..😊..

الوحدة 2: التهيئة العامة وبناء ترويسة الصفحة:

الدرس 1: تثبيت Bootstrap:

الدرس 2: معالجة الصور في Webpack:

- توثيق إدارة ملفات الأصول [Asset Management](https://webpack.js.org/guides/asset-management/).

الدرس 3: إضافة نموذج البحث لقائمة التنقل:

- توثيق [rtl css](https://rtlcss.com/).

الدرس 4: معالجة الخطوط في Webpack:

- يمكن جلب ملفات أنواع الخطوط من مواقع مثل [google fonts](https://fonts.google.com/) و [عرب فونتس](https://arbfonts.com/) وغيرها..

- توثيق القاعدة @font-face من css.

الدرس 5: بناء روابط قائمة التنقل:

الدرس 6: تنسيق نموذج البحث:

- صفحة [npm font awesome](https://www.npmjs.com/package/@fortawesome/fontawesome-free) تشرح طريقة التنزيل..

الدرس 7: تنسيق روابط قائمة التنقل:

الدرس 8: تثبيت حزمة Font Awesome:

الدرس 9: إضافة عنوان منبثق لأيقونة عربة الشراء:

- يحبذ مراجعة المسار الأول من دورة تطوير التطبيقات باستخدام JavaScript..

الدرس 10: إضافة اللمسات النهائية للشريط العلوي:

- تابعت مع المدرب بشكل جيد ولكن من العجلة لم اربطه في ذهني.. راجع الدرس..

الدرس 11: تثبيت ورفع المشروع على GitHub:

- الأمر git clone <repository-url يستخدم لإنشاء نسخة محلية من مستودع Git موجود، يقوم الأمر بنسخ جميع ملفات المستودع والتاريخ الكامل للإصدارات commit history والفروع branches إلى الجهاز المحلي.

الوحدة 3: مكونات الصفحة الرئيسية:

الدرس 1: قسم المنتجات المميزة:

الدرس 2: عارض الشرائح:

- لإزاحة العناصر من بوتستراب [offset classes](https://getbootstrap.com/docs/5.2/layout/columns/#offsetting-columns)..

الدرس 3: ملصق الخصم:

الدرس 4: التنقل بين الشرائح باستخدام الصور المصغرة:

- راجع صفحة توثيق [carousel](https://getbootstrap.com/docs/5.2/components/carousel/#how-it-works) من بوتستراب..

الدرس 5: التنقل بين الشرائح باستخدام أسهم التنقل:

الدرس 6: قسم مميزات المتجر:

الدرس 7: قسم المنتجات الأكثر مبيعًا:

الدرس 8: تنسيق قسم المنتجات الأكثر مبيعًا:

الدرس 9: إضافة الطبقة الزرقاء للمنتجات الأكثر مبيعًا:

الدرس 10: إضافة ملصق الخصم للمنتجات الأكثر مبيعًا:

الدرس 11: إضافة اللمسات النهائية لقسم المنتجات الأكثر مبيعًا:

- راجع هذه [المقالة](https://academy.hsoub.com/programming/javascript/%D9%81%D9%87%D9%85-%D8%A7%D9%84%D8%A3%D8%AD%D8%AF%D8%A7%D8%AB-%D9%81%D9%8A-%D8%AC%D8%A7%D9%81%D8%A7%D8%B3%D9%83%D8%B1%D8%A8%D8%AA-r690/)..

الدرس 12: قسم الماركات العالمية:

الدرس 13: تنسيق قسم الماركات العالمية:

الوحدة 4: التذيل وتجاوبية الصفحة:

الدرس 1: بناء تذييل الصفحة:

الدرس 2: إضافة نموذج الاشتراك بالنشرة البريدية للتذييل:

الدرس 3: إضافة روابط التواصل الاجتماعي للتذييل:

الدرس 4: تجاوبية الصفحة:

- ذكر المدرب أهمية العنصر [meta](https://wiki.hsoub.com/HTML/meta) في HTML.

الدرس 5: تجاوبية قائمة التنقل:

الدرس 6: تجاوبية الشريط العلوي مع شاشات متوسطة الحجم:

الدرس 7: تجاوبية الشريط العلوي مع مختلف شاشات العرض:

الدرس 8: تجاوبية قسم المنتجات المميزة:

الدرس 9: تجاوبية قسم المنتجات الأكثر مبيعًا ومميزات المتجر:

الدرس 10: تجاوبية قسم الماركات العالمية:

الدرس 11: تجاوبية تذييل الصفحة:

الوحدة 5: صفحة تفاصيل المنتج:

الدرس 1: إنشاء صفحة تفاصيل المنتج:

- راجع توثيق [breadcrumb](https://getbootstrap.com/docs/5.1/components/breadcrumb/) من بوتستراب وخاصةً القسم Dividers.

الدرس 3: إكمال تفاصيل المنتج:

- راجع هذه [الصفحة](https://www.w3schools.com/html/html_entities.asp)؛ تتحدث عن الرمز &nbsp;.

الدرس 4: تنسيق تفاصيل المنتج:

- استخدم شيفرة جافاسكربت في الدرس؛ راجع المقطع لفهم الأكواد.

الدرس 5: إكمال تنسيق تفاصيل المنتج:

الدرس 6: وصف المنتج:

الدرس 7: التعليقات على المنتج:

الدرس 8: اختبار تجاوبية الصفحة:

الوحدة 6: صفحة عربة الشراء:

الدرس 1: إنشاء صفحة عربة الشراء:

الدرس 2: بناء منتجات عربة الشراء:

الدرس 3: تنسيق أسطر المنتجات في عربة الشراء:

الدرس 4: تحديث السعر الإجمالي للمنتج:

- يمكن جلب العدد المطلوب الذي اختاره المستخدم من خلال العنصر select حيث أن القيمة التي اختارها المستخدم سيوضع عليها الخاصية selected تلقائيًا.

وللوصول للعنصر select في js نستخدم أي محدد له في html سواء بإعطائه class أو بإعطائه id خاص به أو بإعطائه خاصية جديدة من عندنا - المتعارف عند المبرمجين عند تعريف خاصية جديدة لعنصر ما أن يبدأ ﺒ data -؛ نستخدم تلك المحدد في js بإضافة الدالة querySelectorAll ﻠ document ونضع داخل الدالة المحدد..

- حاول فهم شيفرة جافا سكربت.

الدرس 5: تحديث السعر الإجمالي لعربة الشراء:

الدرس 6: حذف المنتجات من عربة الشراء:

الدرس 7: اختبار تجاوبية الصفحة:

الوحدة 7: صفحة الدفع:

الوحدة 8: صفحة البحث وتواصل معنا:

الدرس 4: إضافة نتائج البحث:

- راجع صفحة توثيق [Pagination](https://getbootstrap.com/docs/5.0/components/pagination/) من بوتستراب.

الوحدة 9: معالجة التنسيقات عبر Sass:

الدرس 1: التعرف على Sass Preprocessor:

- ذكر تعريف عن Sass وسبب إنشائها وفيم تستخدم وغير ذلك.. راجع [صفحة التوثيق ﻠ Sass بالعربية](https://wiki.hsoub.com/Sass) من ويكي حسوب.

الدرس 2: معالجة تنسيقات Sass:

- راجع هذه [الصفحة](https://webpack.js.org/loaders/sass-loader/#root) لإعداد ويب باك ﻠ sass.

الدرس 4: استخدام التشعب في Sass:

- واجب: حاول استنباط التشعب من ملف \_custom\_table. scss وتطبيقها في ملف style. scss.

الدرس 5: كتابة توابع خاصة في Sass:

- في sass التوابع لا تسمى توابع بل تسمى مخاليط Mixins.

- راجع [المخاليط](https://wiki.hsoub.com/Sass/mixins) من موسوعة حسوب..

الدرس 7: تضمين bootstrap من خلال sass:

- إذا أردت إضافة تنسيقات بوتستراب في مشروع ما دون التعديل على تنسيقاته المصدرية نضمن الملف ذات الامتداد css وإذا الإضافة مع التعديل على بعض التنسيقات نضمن الملف ذات الامتداد scss..

الدرس 8: تخصيص Bootstrap باستخدام Sass:

- راجع [صفحة التوثيق](https://getbootstrap.com/docs/5.3/customize/sass/#file-structure) من بوتستراب لتخصيص بوتستراب باستخدام Sass.

- ينصح عند التخصيص بعدم التعديل على الشيفرة الأساسية وإنما في نسخة نستوردها من ملفات الإطار في مجلد التطوير.

- إذا أردت التخصيص في مشروع ما راجع الدرس لمعرفة الطريقة..

الوحدة 10: الخاتمة:

الدرس 3: آفاق مستقبلية:

- يمكنك تصفح موقع [themeforest](https://themeforest.net/?srsltid=AfmBOoo8sVdVjD0_E_XtMDHEIWBfZ8LcYFTrXj0hiYOV91UARnKX5XYX) لرؤية قوالب مواقع..