جافاسكربت - الوحدة الخامسة: المصفوفات والكائنات

تأتي بيانات الأعمال الحقيقية في قوائم من السجلات، مثل الطلبات والعملاء والفواتير. وتحفظ جافاسكربت القائمة في مصفوفة array والسجل في كائن object. وتحول أربع خطوات هذه البيانات إلى إجابات: القائمة، والوصف، والتحويل، والتلخيص. وتجيب كل خطوة عن سؤال بسيط، وتكفي نافذة console في المتصفح لتجربة كل سطر. وفي هذه الوحدة تستخدمها سلمى، أخصائية عمليات المبيعات لدى موزع إقليمي، لإعداد تقرير طلبات الشهر.

1. القائمة

المصفوفة قائمة مرتبة تكتب بين قوسين مربعين، وتفصل عناصرها بفواصل. ولكل عنصر موضع يسمى الفهرس index، وأول فهرس هو الصفر وليس الواحد. وتخبرنا الخاصية length بعدد العناصر، فيقع آخر عنصر في الموضع الذي يسبق الطول بواحد. وتضيف الدالة push عنصرا في النهاية. والسؤال هو: هل الترتيب مهم هنا، وهل هذه قائمة من أشياء متشابهة؟ وطلب موضع غير موجود يعطي undefined وليس خطأ. والمصفوفة المصرح بها بالكلمة const تستطيع أن تستقبل عناصر جديدة، لأن const لا يمنع إلا أن نجعل الاسم يشير إلى مصفوفة مختلفة.

مثال: الطابور عند مكتب الخدمة يشبه المصفوفة: يقف الناس بترتيب ثابت، والشخص الأول فيه هو الموضع صفر في عد المبرمج، والذي يليه هو الموضع واحد، وهكذا. وينضم القادمون الجدد في نهاية الطابور، تماما كما تضيف الدالة push عنصرا في نهاية المصفوفة. وإذا سألنا كم شخصا ينتظر، فالجواب هو طول المصفوفة length، ويقف آخر شخص دائما في الموضع الذي يسبق هذا الرقم بواحد.

في العمل: تتسلم سلمى طلبات الشهر قائمة. وتطرح سؤال القائمة وتحفظها في مصفوفة واحدة اسمها orders، لأن ترتيب الوصول يهم التقرير. ويضيف زميل لها الطلبات المتأخرة بالدالة push، ويخبرها الطول بعدد الطلبات التي تتعامل معها. ولا ينسخ شيء إلى متغيرات منفصلة.

2. الوصف

يصف الكائن شيئا واحدا بخصائص مسماة، تكتب أزواجا من اسم وقيمة بين قوسين معقوفين. نقرأ الخاصية بنقطة يليها اسمها، ونغير خاصية أو نضيف أخرى بإسناد قيمة. وقد تكون القيم نصوصا أو أرقاما أو قيما منطقية أو مصفوفات أو حتى كائنات أخرى. والسؤال هو: ما الحقائق التي نحتاج إلى معرفتها عن هذا الشيء؟ والبيانات الحقيقية غالبا مصفوفة من الكائنات، مثل قائمة طلبات لكل طلب فيها عميل ومنطقة ومبلغ. والكائن المصرح به بالكلمة const يمكن تغيير خصائصه، لأن const يحمي الاسم فقط. واستخدام أسماء الخصائص نفسها في كل كائن يجعل القائمة كلها سهلة المعالجة.

مثال: صندوق بطاقات التعريف صورة جيدة لمصفوفة من الكائنات، وكل بطاقة منفردة هي كائن واحد. وتحمل كل بطاقة الحقول نفسها: اسما ورقم هاتف وشركة، تماما كما يحمل كل كائن من كائنات الطلبات أسماء الخصائص نفسها، والعثور على رقم الهاتف في أي بطاقة هو الإجراء نفسه دائما. وفي الكود تؤدي النقطة التي يليها اسم الخاصية هذا الإجراء، فقراءة رقم الهاتف في بطاقة واحدة تعبير قصير واحد، مهما كانت البطاقة التي نختارها.

في العمل: تصف سلمى كل طلب بكائن له الخصائص customer و region و amount و paid. وتطرح سؤال الوصف وتتأكد أن كل طلب يستخدم أسماء الخصائص نفسها تماما. والخاصية paid قيمة منطقية، والمبلغ رقم وليس نصا. وقراءة منطقة أي عميل صارت تعبيرا بسيطا وواضحا.

3. التحويل

تنشئ الدالة filter مصفوفة جديدة تضم فقط العناصر التي تنجح في اختبار. وتنشئ الدالة map مصفوفة جديدة يستبدل فيها كل عنصر بنتيجة دالة. وكلتاهما تستقبل دالة، والدوال السهمية من الوحدة السابقة تناسبهما تماما. ولا تغير أي منهما

حقوق الطبع والنشر محمية © - أكاديمية أساسيات (شركة أساسيات للخدمات الاستشارية)

المصفوفة الأصلية، فتبقى متاحة لأسئلة أخرى. والسؤال هو: أي العناصر نبقي، وماذا يجب أن يصير كل عنصر؟ فمثلا `orders.filter((order) => order.region === "North")` يبقي طلبات المنطقة الشمالية فقط. والجمع بين الاثنتين، بالتصفية أولا ثم التحويل، يحول قائمة طويلة من السجلات إلى قائمة قصيرة من القيم المطلوبة تماما.

مثال: لدى مسؤول التوظيف كومة من الطلبات، فيستبعد أولا كل طلب لا يستوفي الشروط الأساسية، وهذا ما تفعله filter تماما، لأنها تبقي العناصر التي تنجح في اختبار فقط. ثم يدون الاسم ورقم الهاتف فقط لكل شخص بقي، وهذا ما تفعله map، لأنها تحول كل عنصر إلى قيمة جديدة أصغر. وتبقى كومة الطلبات الأصلية على المكتب دون أن تمس، فإذا سأل أحد سؤالا مختلفا، تستطيع filter و map أن تبدآ من جديد من الكومة الكاملة.

في العمل: تصفي سلمى طلبات المنطقة الشمالية وتحولها إلى مبالغها. وتطرح سؤال التحويل: أي العناصر نبقي، وماذا يجب أن يصير كل عنصر؟ والنتيجة قائمة نظيفة بمبالغ تلك المنطقة، ومصفوفة orders الأصلية لم تمس. وتكرر النمط نفسه لباقي المناطق مع اسم منطقة مختلف.

4. التلخيص

تجمع الدالة reduce كل عناصر المصفوفة في قيمة واحدة. تستقبل دالة وقيمة بداية، وفي كل جولة تأخذ الدالة النتيجة الجارية والعنصر الحالي. وجمع المبالغ هو الاستخدام الكلاسيكي، وقيمة البداية فيه صفر. وتعيد الدالة find أول عنصر ينجح في اختبار، أو undefined إذا لم ينجح أي عنصر. والسؤال هو: ما الجواب الواحد الذي نحتاج إليه من هذه القائمة؟ وفي سؤال مثل الإجمالي أو أول طلب غير مدفوع، يحل سطر واحد من reduce أو find محل حلقة كاملة. ويجب أن نتذكر أن find تتوقف عند أول تطابق ولا تقول شيئا عن الباقي.

مثال: لا يقرأ أمين الصندوق عند إغلاق اليوم كل إيصال بصوت عال، بل يجمعها كلها في إجمالي جار يبدأ من الصفر، وهذا ما تفعله reduce بمبالغ المصفوفة. وإذا سئل عن أول إيصال بلا توقيع، فإنه يفحص الإيصالات واحدا بعد الآخر ويتوقف لحظة العثور على واحد، وهذا ما تفعله find. فالدالة reduce تحول القائمة كلها إلى إجمالي واحد، بينما تعيد الدالة find أول تطابق فقط ولا تقول شيئا عن الإيصالات التي تليه.

في العمل: تستخدم سلمى reduce على مبالغ المنطقة الشمالية ببداية من الصفر، لتنتج إجمالي المنطقة. وتطرح سؤال التلخيص وتستخدم find لتحديد أول طلب غير مدفوع، ليتصل الفريق المالي بذلك العميل أولا. والتقرير الذي كان يستغرق نصف يوم في جدول إلكتروني يصدر في ثوان. وعند إضافة طلب متأخر إلى المصفوفة تحدث إعادة تشغيل الكود نفسه كل الأرقام.

الخاتمة

يصدر تقرير سلمى الشهري الآن من مصفوفة واحدة للطلبات ومن أسطر قليلة. فالقائمة تسأل هل الترتيب مهم وهل العناصر متشابهة، والوصف يسأل ما الحقائق المطلوبة عن كل شيء. والتحويل يسأل أي العناصر نبقي وماذا يصير كل منها، والتلخيص يسأل ما الجواب الواحد الذي يجب أن تعطيه القائمة. وفي الوحدة التالية نربط مثل هذه البيانات بصفحة ويب عبر DOM والأحداث.

المصادر

- شبكة مطوري موزيلا (MDN): الكائن Array – https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array
- شبكة مطوري موزيلا (MDN): العمل مع الكائنات – https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Working_with_objects
- شبكة مطوري موزيلا (MDN): الدالة Array.prototype.filter – https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/filter
- شبكة مطوري موزيلا (MDN): الدالة Array.prototype.reduce – https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/reduce
- شبكة مطوري موزيلا (MDN): الدالة Array.prototype.find – https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/find
- شبكة مطوري موزيلا (MDN): التصريح const – https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/const

حقوق الطبع والنشر محمية © - أكاديمية أساسيات (شركة أساسيات للخدمات الاستشارية)