جافاسكربت - الوحدة الثالثة: الشروط والحلقات التكرارية

تصير البرامج مفيدة حين تستطيع أن تختار وأن تكرر. فالشرط يجعل الكود يختار بين إجراءات، والحلقة تكرر إجراء دون إعادة كتابته. وتغطي أربع خطوات الأمرين معا: القرار، والدمج، والتكرار، والتحكم. وتجيب كل خطوة عن سؤال بسيط، ويمكن تجربة كل مثال في نافذة console بالمتصفح. وفي هذه الوحدة تستخدمها دينا، محللة مالية في شركة تأمين، لترتيب مطالبات العملاء اليومية.

1. القرار

تنفذ الجملة if كتلة من الكود فقط عندما يكون الشرط صحيحا. يكتب الشرط بين قوسين، وتكتب الكتلة بين قوسين معقوفين. ويمكن أن نضيف else if لاختبار شرط آخر، و else لكل ما تبقى. وتفحص جافاسكربت الشروط من الأعلى إلى الأسفل، وتنفذ فقط أول كتلة يكون شرطها صحيحا. والسؤال هو: ما الحالات، وبأي ترتيب يجب أن تفحص؟ ووضع الحالة الأكثر تحديدا أولا يمنع قاعدة عامة من أن تلتقط كل شيء. و else في النهاية تضمن ألا تبقى حالة بلا جواب.

مثال: تتبع موظفة الاستقبال في عيادة قاعدة بسيطة للزائرين: الموظفون يدخلون مباشرة، وأصحاب الحجز يأخذون بطاقة ومقعدا في قاعة الانتظار، والباقون ينتظرون عند المكتب حتى يتفرغ أحد. وتفحص الحالات بهذا الترتيب تماما، فيتبع كل زائر مسارا واحدا فقط. وهذه سلسلة if و else if و else: أول حالة تناسب الزائر هي التي تحسم، ولا يعامل أحد مرتين.

في العمل: ترتب دينا المطالبات بحسب المبلغ. وتطرح سؤال القرار: ما الحالات، وبأي ترتيب يجب أن تفحص؟ فالمطالبة التي تتجاوز حد المدير تذهب إلى المراجعة، والتي تتجاوز الحد الصغير تذهب إلى موظف، وما عدا ذلك يعتمد تلقائيا. وبما أن الفحوص تجري من الأعلى إلى الأسفل، تلتقط المطالبات الأكبر أولا.

2. الدمج

القواعد الحقيقية كثيرا ما تعتمد على أكثر من شرط. العامل المنطقي «و»، ويكتب بعلامتي ampersand، لا يصدق إلا إذا صدق الطرفان. والعامل المنطقي «أو»، ويكتب بخطين عموديين، يصدق إذا صدق أحد الطرفين على الأقل. وعامل النفي، ويكتب بعلامة التعجب، يعكس القيمة المنطقية. والسؤال هو: هل يجب أن تتحقق كل الشروط، أم يكفي واحد؟ وتعامل جافاسكربت بعض القيم على أنها خاطئة داخل الشرط، وهي الصفر والنص الفارغ و null و undefined و NaN والقيمة false نفسها، وتعد كل قيمة أخرى صحيحة. وهذا مفيد في فحوص مثل: هل يوجد اسم، لكنه قد يفاجئنا عندما يكون الصفر جوابا صحيحا.

مثال: يوافق المصرف على القرض فقط إذا كان للمتقدم دخل ثابت وليس عليه دين غير مدفوع، فيجب أن يتحقق الشرطان معا، تماما مثل عامل «و». ويكفي غياب أحد الشرطين ليتوقف الإجراء. أما المتجر الذي يمنح خصما للأعضاء أو للطلاب فالأمر عنده مختلف، لأن شرطا واحدا صحيحا يكفيه، وهذا هو عامل «أو».

في العمل: تحتاج قاعدة دينا للاعتماد التلقائي إلى أمرين: أن يكون المبلغ دون الحد الصغير، وألا يكون على العميل تنبيه سابق. وتطرح سؤال الدمج وتستخدم عامل «و»، لأن الشرطين يجب أن يتحققا معا. ولطابور المراجعة تستخدم عامل «أو»، لأن أيا من المبلغ المرتفع أو التنبيه السابق يكفي. وتفحص المطالبة التي مبلغها صفر، وهو يعد خاطئا في الشرط، فحصا منفصلا حتى لا يظن أنه مبلغ مفقود.

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

3. التكرار

تكرر الحلقة كتلة من الكود دون أن ننسخها. تحتوي الحلقة for بين قوسيها على ثلاثة أجزاء: نقطة بداية، وشرط للاستمرار، وخطوة بعد كل جولة. وأبسط منها الحلقة for of عندما نحتاج كل عنصر في قائمة بدوره، والحلقة while تتكرر ما دام شرطها صحيحا. والقائمة نفسها تسمى مصفوفة، وتشرح وحدة لاحقة المصفوفات بالتفصيل. والسؤال هو: ما الذي يتكرر، وما الذي يتغير في كل جولة؟ فمثلا `for (const claim of claims)` تقرأ هكذا: لكل مطالبة في قائمة المطالبات. وداخل الكتلة يحمل المتغير claim العنصر الحالي، ويمكن حفظ مجموع متراكم في متغير يصرح به قبل الحلقة.

مثال: يعد أمين الصندوق في نهاية اليوم نوعا واحدا من الأوراق النقدية مرة بعد مرة حتى تنتهي الكومة كلها. والمهمة هي نفسها في كل جولة، والورقة التي في يده وحدها تتغير، وهذا هو ما يتغير في كل جولة. والحلقة تصف ذلك مرة واحدة وتترك الآلة تكرره، سواء كانت الكومة صغيرة أو ضخمة.

في العمل: تملك دينا مطالبات اليوم في قائمة. وتطرح سؤال التكرار: ما الذي يتكرر، وما الذي يتغير في كل جولة؟ تزور الحلقة كل مطالبة، وتطبق القواعد من الخطوتين السابقتين، وتجمع المبالغ المعتمدة في مجموع. وعمل كان يستغرق ساعة من موظف ينتهي في جزء من الثانية، والكود نفسه يتعامل مع عشر مطالبات أو عشرة آلاف.

4. التحكم

أحيانا يجب أن تتخطى الحلقة عنصرا أو أن تتوقف مبكرا. تنهي الجملة continue الجولة الحالية وتنتقل إلى العنصر التالي، أما الجملة break فتنهي الحلقة كلها. والسؤال هو: متى يجب أن تتخطى هذه الحلقة، ومتى يجب أن تتوقف؟ ويجب أن يكون للحلقة طريق للنهاية أيضا، فإذا لم يصر شرط حلقة while خاطئا أبدا تعمل إلى ما لا نهاية وتتجمد الصفحة. وتغيير المتغير الذي يعتمد عليه الشرط داخل الحلقة يمنع ذلك. والمخارج الواضحة تجعل الحلقات آمنة على البيانات الحقيقية.

مثال: يسمح ضابط الأمن الذي يفحص الحقائب في المطار لحقيبة فارغة بالمرور سريعا وينتقل إلى التالية، لكنه يوقف الطابور كله عندما يدق الإنذار. الأول تخط، مثل continue، والثاني توقف، مثل break. وكلاهما قرار يتخذ داخل مهمة متكررة، ويبقى الطابور آمنا لأن كل مخرج واضح.

في العمل: تستخدم حلقة دينا continue لتخطي المطالبات المغلقة أصلا. وتطرح سؤال التحكم وتستخدم break فور نفاد ميزانية الاعتماد اليومية، لتنتظر المطالبات الباقية اليوم التالي. وتتأكد من أن حلقة while التي تعيد محاولة رفع مستند تغير عدادها في كل جولة. وتنتهي الحلقة الآن في كل مرة، ويثق الفريق المالي بالتقرير اليومي.

الخاتمة

يجري فرز دينا اليومي الآن من تلقاء نفسه وينتهي في كل مرة. فالقرار يسأل ما الحالات وبأي ترتيب تفحص، والدمج يسأل هل يجب أن تتحقق كل الشروط أم يكفي واحد. والتكرار يسأل ما الذي يتكرر وما الذي يتغير في كل جولة، والتحكم يسأل متى تتخطى الحلقة ومتى تتوقف. وفي الوحدة التالية نجمع مثل هذا المنطق في دوال يمكن إعادة استخدامها.

المصادر

- شبكة مطوري موزيلا (MDN): التحكم في سير التنفيذ ومعالجة الأخطاء – https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Control_flow_and_error_handling
- شبكة مطوري موزيلا (MDN): الحلقات والتكرار – https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Loops_and_iteration
- شبكة مطوري موزيلا (MDN): جملة if...else – https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/if...else
- شبكة مطوري موزيلا (MDN): جملة for...of – https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/for...of

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