جافاسكربت - الوحدة السادسة: نموذج كائن المستند (DOM) والأحداث

صفحة الويب ليست صورة فقط، فالمتصفح يحتفظ بها كبنية تستطيع جافاسكربت قراءتها وتغييرها والزائر يرى. وتسمى هذه البنية نموذج كائن المستند، أو DOM، وتسمى أفعال الزائر أحداثا. وتربط أربع خطوات الكود بالصفحة: الاختيار، والتغيير، والاستماع، والبناء. وتجيب كل خطوة عن سؤال بسيط، وتستطيع نافذة console في المتصفح تشغيل كل سطر على الصفحة نفسها. وفي هذه الوحدة يستخدمها فادي، مطور ويب في شبكة عيادات طبية، ليجعل مواعيد صفحة الحجز تستجيب للنقرات.

1. الاختيار

يحول المتصفح HTML الصفحة إلى شجرة من العناصر، والكائن document هو مدخل هذه الشجرة. وتجد الدالة querySelector أول عنصر يطابق محددا selector، وهو لغة الأنماط نفسها في CSS، مثل اسم class أو id. وتعيد الدالة querySelectorAll كل العناصر المطابقة. والسؤال هو: أي عنصر في الصفحة نريد العمل عليه؟ وإذا لم يطابق شيء يعطي querySelector القيمة null، واستخدام هذه النتيجة لاحقا يسبب خطأ، لذلك يستحق الأمر الفحص. ويجب أن يعمل الكود بعد وجود العناصر، ولذلك يوضع الكود عادة في نهاية الصفحة أو يحمل بالسمة defer.

مثال: يحمل مدير المسرح قائمة تضم ممثلين كثيرين، ويريد أن ينادي واحدا منهم إلى الخشبة. ولا يقرأ القائمة كلها بصوت عال، بل يعطي وصفا واحدا: صاحب المعطف الأحمر. فيتقدم ممثل واحد بالضبط من الحشد، وإذا لم يرتد أحد معطفا أحمر فلا يتقدم أحد، وهذا يشبه القيمة null. والمحدد selector يعمل بالطريقة نفسها: يصف العنصر باسم class أو id، فتعيد الصفحة العنصر المطابق.

في العمل: تضم صفحة الحجز عند فادي قائمة بمواعيد الزيارة ومنطقة للرسائل. ويطرح سؤال الاختيار، فيختار منطقة الرسائل باسم id الخاص بها، وكل أزرار المواعيد باسم class المشترك بينها. ويتحقق في console أن الاختيار ليس null قبل كتابة المزيد من الكود. وتستطيع الخطوات اللاحقة الاعتماد الآن على هذين الاختيارين.

2. التغيير

بعد أن نمسك بالعنصر نستطيع تغيير ما يراه الزائر. تقرأ الخاصية textContent نص العنصر أو تضبطه، وتعامل كل شيء على أنه نص عادي. وتضيف الخاصية classList أسماء class أو تزيلها أو تبدلها، وبهذا يشغل الكود أنماطا جاهزة في CSS ويوقفها. وتغير الدالة setAttribute سمات مثل disabled. والسؤال هو: ماذا يجب أن يبدو أو يقرأ بشكل مختلف، وهل يكفي اسم class؟ وتستطيع الخاصية innerHTML إدراج HTML، لكنها إذا استقبلت نصا كتبه زائر فقد يحقن زائر خبيث كودا ضارا، ولذلك يبقى textContent الخيار الأكثر أمانا. وإبقاء الشكل في CSS وتبديل أسماء class فقط يبقي الكود قصيرا.

مثال: يقلب موظف الاستقبال في الفندق لافتة الغرفة من متاحة إلى مشغولة عندما يصل ضيف جديد. ولا يعيد بناء الباب ولا الغرفة. فالباب كما هو وحالته وحدها تتغير، فيرى الناس شيئا مختلفا. والخاصية classList تفعل ذلك بالعنصر: إضافة اسم class أو إزالته تقلب شكله وتترك العنصر نفسه في مكانه.

في العمل: عندما ينقر مريض على موعد يغير فادي الصفحة. ويطرح سؤال التغيير، ويضيف إلى الموعد المختار اسم class هو selected ليظهر التمييز، ويضبط نص منطقة الرسائل لتأكيد الوقت. ولأنه يستخدم textContent، لا يمكن لاسم مريض يكتب في حقل أن يعمل ككود. ويبقى نمط التمييز في CSS، ولا يفعل الكود إلا تشغيله.

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

3. الاستماع

الحدث شيء يقع في الصفحة، مثل نقرة أو ضغطة مفتاح أو تغير في حقل. وتربط الدالة addEventListener دالة، تسمى المعالج handler، بعنصر ونوع حدث. ويعمل المعالج في كل مرة يقع فيها ذلك الحدث، ويستقبل كائن حدث يخبرنا بالمزيد، مثل أي عنصر جرى النقر عليه. والسؤال هو: ماذا يجب أن يحدث عندما يقع هذا الحدث؟ فمثلا `button.addEventListener("click", showSlots)` يشغل الدالة showSlots عند كل نقرة. ونمرر الدالة نفسها بلا قوسين، لأن إضافة القوسين تشغلها فورا بدلا من انتظار النقرة. ويمكن لمستمع واحد على عنصر أب أن يعالج نقرات كل أبنائه، حتى الذين يضافون لاحقا، ويسمى هذا تفويض الأحداث.

مثال: في مطبخ المطعم جرس، وفي كل مرة يقرعه زبون يستجيب الطاقم بالطريقة نفسها. ولا يراقب الطاقم الزبون باستمرار، ولا يتحرك قبل الجرس. ومستمع الحدث ينتظر بالطريقة نفسها ولا يتفاعل إلا عند وقوع الحدث.

في العمل: يرفق فادي مستمع نقر واحدا بقائمة المواعيد بدلا من مستمع لكل زر. ويطرح سؤال الاستماع ويستخدم كائن الحدث لمعرفة أي موعد جرى النقر عليه. ويميز المعالج ذلك الموعد ويزيل التمييز عن السابق. ولأن المستمع على القائمة، تعمل المواعيد التي تضاف لاحقا دون كود إضافي.

4. البناء

كثيرا ما لا تكون العناصر موجودة في الصفحة في البداية، لأنها تأتي من بيانات. تصنع الدالة createElement عنصرا جديدا في الذاكرة، ونضبط نصه وأسماء class فيه بالطريقة نفسها. ثم تضعه الدالة append داخل عنصر أب، وعندئذ فقط يراه الزائر. والسؤال هو: أي بيانات يجب أن تصير عناصر، وأين توضع؟ وحلقة على مصفوفة كائنات، كما في الوحدة السابقة، تنشئ عنصرا لكل سجل. وبناء عناصر كثيرة واحدا بعد الآخر مقبول في صفحة حجز، ويبقى الكود مقروءا.

مثال: تتسلم بائعة الزهور قائمة طلبات صباح يوم الاثنين، ولا توجد باقة واحدة بعد. وتجهز لكل طلب بطاقة باقة عليها اسم الزبون والتاريخ، وتضعها على الرف المناسب. والبطاقات لا وجود لها قبل وصول القائمة، والقائمة الأطول تعني ببساطة بطاقات أكثر. وإنشاء العناصر ووضعها يعمل بالطريقة نفسها: تصنع الدالة createElement كل بطاقة جديدة، وتضعها الدالة append على رفها.

في العمل: يتسلم فادي المواعيد الشاغرة لليوم من نظام العيادة مصفوفة كائنات. ويطرح سؤال البناء، فيدور على المصفوفة وينشئ زرا لكل موعد، ونصه وقته. ويأخذ كل زر اسم class المشترك ويلحق بقائمة المواعيد. وعندما تضيف العيادة موعدا جديدا تظهره الصفحة دون أي تغيير في الكود.

الخاتمة

تستجيب صفحة الحجز عند فادي الآن لما يفعله المريض، ومواعيدها تأتي من البيانات. فالاختيار يسأل أي عنصر نريد، والتغيير يسأل ماذا يجب أن يبدو أو يقرأ بشكل مختلف. والاستماع يسأل ماذا يحدث عند وقوع الحدث، والبناء يسأل أي بيانات تصير عناصر وأين توضع. وفي الوحدة التالية نستخدم الأفكار نفسها لمعالجة النماذج، والتحقق مما يكتبه الزوار، وتذكره بالتخزين المحلي.

المصادر

- شبكة مطوري موزيلا (MDN): مدخل إلى نموذج كائن المستند DOM – https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction
- شبكة مطوري موزيلا (MDN): الدالة Document.querySelector – https://developer.mozilla.org/en-US/docs/Web/API/Document/querySelector
- شبكة مطوري موزيلا (MDN): الدالة EventTarget.addEventListener – https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener
- شبكة مطوري موزيلا (MDN): الخاصية Element.classList – https://developer.mozilla.org/en-US/docs/Web/API/Element/classList
- شبكة مطوري موزيلا (MDN): الخاصية Node.textContent – https://developer.mozilla.org/en-US/docs/Web/API/Node/textContent
- شبكة مطوري موزيلا (MDN): الدالة Element.append – https://developer.mozilla.org/en-US/docs/Web/API/Element/append

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