

Doelstellingen

Tijdens dit labo maken we nog een oefening op DOM, zonder jQuery. In meer detail zijn de topics van vandaag: DOM en Events.

Werkwijze, timing en upload

Dit labo loopt over één sessie en moet op het einde van de week geupload worden via Toledo (zondag 20u). Upload één enkele zip met daarin je oplossingen.

Het huishoudelijk reglement is van toepassing.

Opgaves

1. Sudoku

Opmerking over de opgave: de HTML maakt gebruik van de HTML5 `contenteditable` attribuut. Dit werkt best in Webkit browsers; voor andere browsers is het principe hetzelfde maar zijn fixes nodig zoals een extra `<div>` in elke `<td>` voor Internet Explorer enz... Om de code niet nodeloos complex te maken werk je best in Chrome.

Open de opgave voor de Sudoku. We zullen dit in verschillende stappen scripten tot een volwaardig Sudoku spel.

1.1 Verbinden van het script aan de HTML

Maak een script aan en verbind het aan je HTML.

Tip: inspireer je op http://leercentrum.ikdoeict.be/slides/javascript/02_dom.html#/52

1.2 Ter voorbereiding: extra properties aan elke `<td>` toevoegen

Er zijn drie properties van elke tabelcel die verderop van pas zullen komen:

- wat is de kolomnummer van de cel?
- wat is de rijnummer van de cel?
- wat is de laatste waarde van de cel (voor het geval de gebruiker een waarde wijzigt en dan op ESC drukt)?

In plaats van deze proberen te achterhalen telkens een cel gewijzigd wordt, kun je ook in het begin alle cellen overlopen en deze waarden aan de cellen zelf kleven. Daarom: overloop alle cellen van de tabel en kleef aan elke cel deze drie properties

met de juiste waarden. Controleer je werk door een click-event aan de cellen te verbinden waarmee je deze drie waarden via een `alert()`, of beter nog, via `console.log()` controleert.

Tip: in plaats van aan elke cel apart het click event toe te voegen kun je dit ook doen voor de hele tabel ineens, zie http://leercentrum.ikdoeict.be/slides/javascript/02_dom.html#/47.

1.3 Editeren van de cel

Als op ESC gedrukt wordt, moet de waarde van de cel hersteld worden. Als op Enter gedrukt wordt of de cel focus verliest (“blur”), dan moet de waarde van de cel behouden blijven.

Tip: het detecteren van een ESC, Enter of blur van een cel zou niet zo’n probleem moeten opleveren, maar daar het juiste gedrag aan koppelen is niet zo eenvoudig als het op het eerste zicht lijkt. Aarzel niet te lang om de docent om raad te vragen als je de juiste strategie niet vindt.

1.4 Toekennen van de celwaarde

Bij het toekennen van een nieuwe waarde aan een cel moet het een en het ander gebeuren:

- controleer of de nieuwe waarde wel degelijk tussen 1 en 9 ligt en pas het zo nodig aan
- bereken het aantal lege cellen en pas `span#numEmptyCells` aan

Tip: het staat je vrij je eigen code te organiseren, maar in onze oplossing is hiervoor gebruik gemaakt van een aparte functie `commitValue(cell)`.

1.5 Controle van de celwaarde

Tenslotte:

- controleer of de celwaarde mogelijk is in het spel of niet (controleer enkel op dubbels op de horizontale rij; de controles op de verticale rij en binnen het 3x3 vierkant kun je eventueel nog doen als je veel tijd over hebt)
- bereken het aantal foute cellen en pas `span#numErrors` aan
- is het spel afgelopen, zet dan in `div#summary` de tekst “Gefeliciteerd!”

Hiervoor kun je de functie `commitValue(cell)` uitbreiden.

Tip: er is in de css reeds een class “invalid” voorzien voor ongeldige celwaarden; wijs deze in Javascript toe aan de cel indien de nieuwe waarde dubbels bevat op dezelfde rij. Het tellen van het totaal aantal foute cellen is dan met wat creativiteit één enkele regel code.