**Voorbereiden**

|  |
| --- |
| In dit hoofdstuk gaat u verder met het ontdekken van de mogelijkheden van JavaScript. U maakt kennis met herhalings- en keuzeopdrachten. Ook leert u JavaScript gebruiken in samenhang met het Document Object Model (DOM). U maakt kennis met objecten en reguliere expressies, en met het gebruik van arrays.  Dit hoofdstuk bevat online oefening met voorgedefinieerde scripts, waarbij u direct resultaat ziet en zelf kleine wijzigingen kunt aanbrengen in de code.  **Leerdoelen** Na dit hoofdstuk:   * kunt u zelf objecten maken * kunt u properties en methoden voor zelfgemaakte objecten definiëren * kunt u het for-statement gebruiken * kunt u het while-statement gebruiken * kunt u het do-while-statement gebruiken * kunt u het break- en continue-statement gebruiken * kunt u het switch-statement gebruiken * kunt u het window-object van het DOM gebruiken * kunt u properties van het document-object manipuleren en methoden van dit object gebruiken * kunt u het history-object gebruiken * kunt u URL-informatie lezen en bewerken met het location-object * kunt u het navigator-object gebruiken * kunt u basale reguliere expressies gebruiken * kunt u gebruikersinvoer testen * kent u de properties en de werking van de methoden van string, met nameindexOf(), lastindexOf(), substring() en charAt() * kunt u arrays gebruiken * kent u een aantal mogelijkheden van Math.   Dit hoofdstuk bevat uitleg over een aantal begrippen en nodigt u uit delen uit de verplichte literatuur (Flanagan) te bestuderen en online oefeningen uit te voeren via de website w3schools.com.  Aan het eind van dit hoofdstuk maakt u een inzendopgave, waarin u een groot aantal aspecten uit dit hoofdstuk moet toepassen.  **Debuggen van JavaScript** Naarmate programma's groter worden, is de kans op fouten ook groter en is het lastiger deze te vinden. Alle browsers beschikken over hulpmiddelen voor de programmeur om fouten op te sporen. In het vorige hoofdstuk is het *console* al genoemd, een plek waar u liefst als eerste gaat kijken als u vermoedt dat er een fout in uw code zit (en ook als u vermoedt dat er geen fout in zit).  Een ander handig hulpmiddel is de *debugger*, die u in Chrome onder de tab *Sources*kunt vinden. Hierin kunt u een script openen, aan de linkerkant van het script staan regelnummers. Als u op zo'n nummer klikt, zet u een zogeheten *breakpoint*(de breakpoints ziet u ook aan de rechterkant van het scherm). Het is nu mogelijk het programma te laten runnen (met Ctrl-R); de uitvoering stopt dan automatisch bij het breakpoint. Vanaf dat moment kunt u regel voor regel door de code lopen en aan de rechterkant van het scherm de waarden van variabelen in de gaten houden.  In een tekst als dit is het lastig uitleggen wat u waar precies kunt zien, beter is het daarom een van de vele video's te bekijken die over dit onderwerp op internet staan. Bijvoorbeeld dit [filmpje](https://www.youtube.com/watch?v=htZAU7FM7GI).  *Ingeschatte tijdsduur: 30 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Bestuderen**

|  |
| --- |
| **Objecten in JavaScript** U hebt eerder kennisgemaakt met de datatypen string, number en boolean. Dit zijn zogeheten primitieve typen of enkelvoudige typen. Ze zijn heel fundamenteel, omdat het bouwstenen zijn voor wellicht het belangrijkste datatype in JavaScript: object.  Een object heeft *properties*(eigenschappen, zoals kleur, grootte, leeftijd) en*methoden,* functies die iets met het object en eventueel met de properties van het object kunnen doen.  Een eerste, heel eenvoudig object dat de naam p krijgt, is het volgende:  <script> var p = {     firstName : "John",     lastName  : "Doe",     age       : 50,     eyeColor  : "blue" }; </script>  De inhoud van het object p staat tussen de accolades. Let op de komma's aan het eind van de vier regels. Het object p heeft vier properties, die achtereenvolgens de namen firstName, lastName, age en eyeColor hebben. Deze manier om een object te maken, heet een *object literal*. De inhoud van het object, dat wil zeggen de namen van de properties en hun waarden, worden letterlijk uitgeschreven. Zie [w3schools.com - objects](http://www.w3schools.com/js/js_objects.asp).  Er zijn verschillende manieren om objecten te maken.   * Met een literal: zie hierboven. Dit is een makkelijke manier voor een enkel object, maar is omslachtig als er meerdere soortgelijke objecten gemaakt moeten worden. * Met het keyword new. * Door het gebruik van een constructor-functie; dit is de meest gebruikelijke manier.   Zie [w3schools.com - object definition](http://www.w3schools.com/js/js_object_definition.asp) tot en met 'The this keyword'.  Een voorbeeld van een constructor-functie staat hieronder:  function person(first, last, age, eye) {     this.firstName = first;     this.lastName = last;     this.age = age;     this.eyeColor = eye; }  Deze functie maakt, als hij aangeroepen wordt, een object van het type person. Objecten van dit type krijgen vier properties: firstName, lastName, age en eyeColor. De waarden voor die properties geeft u door via de parameters van de functie. Op de volgende manier maakt u twee objecten:  var p1 = new person("John", "Doe", 50, "blue"); var p2 = new person("Sally", "Rally", 48, "green");  Als eerste wordt het object p1 gemaakt, met behulp van de constructor-functieperson(). De vier waarden in de aanroep,  "John", "Doe", 50, blue" komen in de vier parameters van de functie terecht. In de functie komt het keyword this voor. Omdat het in deze aanroep om p1gaat, is this hetzelfde als p1. Er staat dus eigenlijk:  p1.firstName = first; p1.lastName = last; p1.age = age; p1.eyeColor = eye;  Als p2 wordt gemaakt en de constructor-functie wordt aangeroepen met andere argumenten, is thishetzelfde als p2. U moet de code in de body van de functie dan lezen als:  p2.firstName = first; p2.lastName = last; p2.age = age; p2.eyeColor = eye;  **Methoden** Objecten kunnen niet alleen properties bevatten, maar ook methoden. Het woord methode is de benaming voor een functie die bij een object hoort.  Methoden kunt u in de constructor-functie definiëren. Een voorbeeld:  function person(first, last, age, eye) {     this.firstName = first;     this.lastName = last;     this.age = age;     this.eyeColor = eye;      this.fullName = function() {        return this.firstName + " " + this.lastname;     }         this.changeLastName = function(name) {         this.lastName = name;     } }  Deze constructor-functie maakt objecten met vier properties (firstName, lastName, age en eyeColor) en twee methoden. De methoden hetenfullName() en changeLastName().  De declaratie van de functie begint met het keyword this, gevolgd door een punt, gevolgd door de naam van de functie:  this.fullName  Tot zover lijkt het op de definitie van een property. Achter het isgelijkteken staat de definitie van de methode: het woord function, in dit geval een functie zonder parameters en dan de body van de functie.  De tweede methode, changeLastName, is op dezelfde manier gedeclareerd, maar deze methode heeft een parameter: name.  De methoden kunnen worden aangeroepen met de naam van een object, gevolgd door een punt, gevolgd door de naam van de functie, met tussen haakjes eventuele argumenten:  *// Maak een object met behulp van de constructor* var myMother = new person("Jola","Klynstra",48,"green");  myMother.changeLastName("Zalewska"); var fn = myMother.fullName();    *// Jola Zalewska*  *Ingeschatte tijdsduur: 45 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Oefenen**

|  |
| --- |
| In de volgende oefenopgave past u de zojuist geleerde kennis toe. Het is erg belangrijk de oefenopgave eerst zelf te maken, voordat u de uitwerking bekijkt.  Maak opgave 1, de opgave vindt u [hier](https://www5.loi.nl/access/content/group/255889f1-27c3-4e90-944f-11761ec46ec4/Bestanden%20oefenopgaven/Oefenopgaven_hoofdstuk_4.pdf).  Controleer uzelf aan de hand van de uitwerking via deze [link](https://www5.loi.nl/access/content/group/255889f1-27c3-4e90-944f-11761ec46ec4/Bestanden%20oefenopgaven/Uitwerkingen_oefenopgaven_H4.pdf).  *Ingeschatte tijdsduur: 60 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Bestuderen**

|  |
| --- |
| **Namen van objecten zijn referenties** In JavaScript geldt dat de naam van een object een referentie is, een soort link of pointer. Dat heeft in sommige gevallen consequenties voor de betekenis van de code. Een voorbeeld (met de constructor-functie):  var p1 = new person("John", "Doe", 50, "blue"); var p3 = p1;  Het lijkt hier alsof p3 een kopie is van p1. Dat is echter niet zo. De variabele p1 is een referentie (verwijzing) naar het object met de gegevens "John" et cetera. De variabele p3 is een kopie van die verwijzing. Dus p3 verwijst naar hetzelfde object als p1.  Dat betekent dat een verandering in p3, ook een wijziging in p1 impliceert.  var p1 = new person("John", "Doe", 50, "blue"); var p3 = p1;    // p3 verwijst naar hetzelfde object als p1  p3.age = 33;    // Nu is ook p1.age gelijk aan 33  Omdat p3 een kopie bevat van de verwijzing naar het object en niet een kopie is van het object zelf, heet p3 ook wel een *ondiepe kopie*(shallow copy).  Met variabelen van een primitief type is dit anders:  var p1 = 20; var p3 = p1;  // p3 bevat een kopie  p3 = 100;     // Nu is p1 nog steeds 20  U ziet dat, ondanks de ogenschijnlijke overeenkomst tussen beide fragmenten, er een groot verschil is in betekenis.  *Ingeschatte tijdsduur: 15 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Oefenen**

|  |
| --- |
| In de volgende oefenopgaven past u de zojuist geleerde kennis toe. Het is erg belangrijk de oefenopgaven eerst zelf te maken, voordat u de uitwerking bekijkt.  Maak opgave 2 en 3, de opgaven vindt u [hier](https://www5.loi.nl/access/content/group/255889f1-27c3-4e90-944f-11761ec46ec4/Bestanden%20oefenopgaven/Oefenopgaven_hoofdstuk_4.pdf).  Controleer uzelf aan de hand van [deze uitwerkingen](https://www5.loi.nl/access/content/group/255889f1-27c3-4e90-944f-11761ec46ec4/Bestanden%20oefenopgaven/Uitwerkingen_oefenopgaven_H4.pdf#page=2).  *Ingeschatte tijdsduur: 60 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.    Onderkant formulier |

**Oefenen**

|  |
| --- |
| Test uw kennis op:   * [Objects](http://www.asmarterwaytolearn.com/js/69.html) * [Objects: Properties](http://www.asmarterwaytolearn.com/js/70.html) * [Objects: Methods](http://www.asmarterwaytolearn.com/js/71.html) * [Objects: Constructors](http://www.asmarterwaytolearn.com/js/72.html) * [Objects: Constructors for methods](http://www.asmarterwaytolearn.com/js/73.html).   *Ingeschatte tijdsduur: 120 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Bestuderen**

|  |
| --- |
| Het boek van Flanagan bevat een heel hoofdstuk over objecten. Er valt over objecten dan ook nog veel meer te vertellen dan hier is gebeurd. Leest u in elk geval:   * de inleiding van hoofdstuk 6 (Objects) * paragraaf 6.1.1 (Object Literals) * paragraaf 6.1.2 (Creating Objects with new).   *Ingeschatte tijdsduur: 40 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Bestuderen**

|  |
| --- |
| In een realistische toepassing kan het aantal gegevens of het aantal objecten soms groot zijn. Denk bijvoorbeeld aan de gegevens uit een database van een webwinkel. Een *array*biedt een manier om een grote hoeveelheid gegevens op te slaan.  Lees [hier](https://www5.loi.nl/access/content/group/255889f1-27c3-4e90-944f-11761ec46ec4/Bestanden%20SA/Arrays.pdf) meer over arrays.  *Ingeschatte tijdsduur: 60 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Oefenen**

|  |
| --- |
| In de volgende oefenopgave past u de zojuist geleerde kennis toe. Het is erg belangrijk de oefenopgave eerst zelf te maken, voordat u de uitwerking bekijkt.  Maak opgave 4, de opgave vindt u [hier](https://www5.loi.nl/access/content/group/255889f1-27c3-4e90-944f-11761ec46ec4/Bestanden%20oefenopgaven/Oefenopgaven_hoofdstuk_4.pdf).  Controleer uzelf aan de hand van [deze uitwerkingen](https://www5.loi.nl/access/content/group/255889f1-27c3-4e90-944f-11761ec46ec4/Bestanden%20oefenopgaven/Uitwerkingen_oefenopgaven_H4.pdf#page=4).  *Ingeschatte tijdsduur: 60 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Oefenen**

|  |
| --- |
| Test uw parate kennis via:   * [Arrays](http://www.asmarterwaytolearn.com/js/15.html) * [Arrays: adding and removing elements](http://www.asmarterwaytolearn.com/js/16.html) * [Arrays: removing, inserting, and extracting elements](http://www.asmarterwaytolearn.com/js/17.html).   *Ingeschatte tijdsduur: 90 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Bestuderen**

|  |
| --- |
| Flanagan geeft in hoofdstuk 7 achtergrondinformatie bij het onderwerp arrays.  Bestudeer:   * de inleiding van hoofdstuk 7 (Arrays) * paragraaf 7.1 (Creating Arrays) tot en met paragraaf 7.5 (Adding and Deleting Array Elements) * paragraaf 7.8 (Array Methods).   Het is altijd nuttig om voorbeelden die u in Flanagan leest zelf uit te proberen in een script.  *Ingeschatte tijdsduur: 90 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Bestuderen**

|  |
| --- |
| **Loops** Een loop of lus is een herhalingsopdracht waarmee een of meer statements herhaald kunnen worden. Kenmerk van een herhaling is een criterium op grond waarvan besloten wordt of de herhaling doorgaat of niet.  Een voorbeeld van een herhalingsopdracht is zoeken en vervangen in een tekstverwerker. Het vervangen gaat in principe door tot het einde van het document. Een ander voorbeeld is de tafel van tien van de basisschool: tien keer moet een uitkomst worden uitgerekend, daarna stopt de berekening.  Lees [hier](https://www5.loi.nl/access/content/group/255889f1-27c3-4e90-944f-11761ec46ec4/Bestanden%20SA/Loops.pdf) meer over loops.  *Ingeschatte tijdsduur: 70 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Oefenen**

|  |
| --- |
| In de volgende oefenopgaven past u de zojuist geleerde kennis toe. Het is erg belangrijk de oefenopgaven eerst zelf te maken, voordat u de uitwerking bekijkt.  Maak opgave 5 en 6, de opgaven vindt u [hier](https://www5.loi.nl/access/content/group/255889f1-27c3-4e90-944f-11761ec46ec4/Bestanden%20oefenopgaven/Oefenopgaven_hoofdstuk_4.pdf#page=2).  Controleer uzelf aan de hand van [deze uitwerkingen](https://www5.loi.nl/access/content/group/255889f1-27c3-4e90-944f-11761ec46ec4/Bestanden%20oefenopgaven/Uitwerkingen_oefenopgaven_H4.pdf#page=5).  *Ingeschatte tijdsduur: 60 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Oefenen**

|  |
| --- |
| Test uw parate kennis op:   * [*for* loops](http://www.asmarterwaytolearn.com/js/18.html) * [*for* loops: flags, Booleans, array length, and loopus interruptus](http://www.asmarterwaytolearn.com/js/19.html).   *Ingeschatte tijdsduur: 60 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Bestuderen**

|  |
| --- |
| Bestudeer uit Flanagan:   * paragraaf 5.5.3 (for) * paragraaf 7.6 (Iterating Arrays).   *Ingeschatte tijdsduur: 45 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Bestuderen**

|  |
| --- |
| **Het while-statement** Een herhalingsopdracht als het for-statement gebruikt u in het algemeen als van tevoren bekend is om hoeveel herhalingen het gaat. Dat kunt u echter niet altijd weten. Als u het precieze aantal herhalingen niet van tevoren weet, is een while-statement meestal geschikter.  Een while-statement bestaat uit het woord while met daarachter tussen haakjes een test (conditie) en een body:  while( *conditie*) { *body* }  Als de conditie true is, wordt de body uitgevoerd en daarna automatisch opnieuw gekeken of de conditie true of false is. Zolang de conditie true oplevert, wordt de body uitgevoerd en dus steeds herhaald. Zodra de conditie de waarde false krijgt, is het while-statement klaar en gaat het programma verder met de regel na de sluitaccolade van het while-statement.  In het algemeen heeft een while-statement een structuur als in de volgende figuur.  388986S.JPG  Zie [w3schools.com - loop while](http://www.w3schools.com/js/js_loop_while.asp).  **Gelijkwaardigheid van for-statement en while-statement** In JavaScript zijn het for-statement en while-statement in wezen gelijkwaardig. Neem het volgende for-statement:  var s = ""; for( var k = 1; k <= 10 ; k += 2 ) {   s += k + "  "; } alert( s );  Ga na dat de uitvoer van dit fragment gelijk is aan: 1 3 5 7 9.  Het is niet moeilijk dit stukje code met een while-statement te herschrijven. De onderdelen van het controlegedeelte van het for-statement splitst u in drieën. De initialisatie gebeurt vóór het while-statement; de conditie van het for-statement wordt de conditie van het while-statement en de verandering van de controlevariabele doet u in de body van het while-statement:  var s = ""; var k = 1; while( k <= 10 ) {   s += k + "  ";   k += 2; } alert( s );  Het omgekeerde, een while-statement als for-statement schrijven, is net zo makkelijk. Elk while-statement heeft de volgende vorm:  statement; while( conditie ) {   statement;   statement;    ... }  Dit kunt u met een for-statement schrijven als hieronder, waarbij het eerste en laatste gedeelte van het controlegedeelte leeg zijn:  statement; for( ;conditie; ) {   statement;   statement;    ... }  Ondanks de gelijkwaardigheid is het gebruikelijk een for-statement toe te passen als het aantal herhalingen van tevoren vastligt.  *Ingeschatte tijdsduur: 45 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Oefenen**

|  |
| --- |
| In de volgende oefenopgave past u de zojuist geleerde kennis toe. Het is erg belangrijk de oefenopgave eerst zelf te maken, voordat u de uitwerking bekijkt.  Maak opgave 7, de opgave vindt u [hier](https://www5.loi.nl/access/content/group/255889f1-27c3-4e90-944f-11761ec46ec4/Bestanden%20oefenopgaven/Oefenopgaven_hoofdstuk_4.pdf#page=3).  Controleer uzelf aan de hand van [deze uitwerkingen](https://www5.loi.nl/access/content/group/255889f1-27c3-4e90-944f-11761ec46ec4/Bestanden%20oefenopgaven/Uitwerkingen_oefenopgaven_H4.pdf#page=6).  *Ingeschatte tijdsduur: 30 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Oefenen**

|  |
| --- |
| Test uw parate kennis op:   * [while loops](http://www.asmarterwaytolearn.com/js/41.html).   *Ingeschatte tijdsduur: 30 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Bestuderen**

|  |
| --- |
| * Bestudeer paragraaf 5.5.1 (while) uit Flanagan. * Bestudeer tevens [deze aanvullende leerstof](https://www5.loi.nl/access/content/group/255889f1-27c3-4e90-944f-11761ec46ec4/Bestanden%20SA/Het_do-while-statement.pdf) over het do-while-statement, het switch-statement en de statements break en continue.   *Ingeschatte tijdsduur: 75 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Oefenen**

|  |
| --- |
| In de volgende oefenopgaven past u de zojuist geleerde kennis toe. Het is erg belangrijk de oefenopgaven eerst zelf te maken, voordat u de uitwerking bekijkt.  Maak opgave 8 t/m 10, de opgaven vindt u [hier](https://www5.loi.nl/access/content/group/255889f1-27c3-4e90-944f-11761ec46ec4/Bestanden%20oefenopgaven/Oefenopgaven_hoofdstuk_4.pdf#page=3).  Controleer uzelf aan de hand van [deze uitwerkingen](https://www5.loi.nl/access/content/group/255889f1-27c3-4e90-944f-11761ec46ec4/Bestanden%20oefenopgaven/Uitwerkingen_oefenopgaven_H4.pdf#page=7).  *Ingeschatte tijdsduur: 150 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Oefenen**

|  |
| --- |
| Test uw parate kennis op:   * [*switch* statements: how to start them](http://www.asmarterwaytolearn.com/js/39.html) * [*switch* statements: how to complete them](http://www.asmarterwaytolearn.com/js/40.html) * [*do...while* loops](http://www.asmarterwaytolearn.com/js/42.html) * [*switch* statements: how to start them](http://www.asmarterwaytolearn.com/js/39.html) * [*switch* statements: how to complete them](http://www.asmarterwaytolearn.com/js/40.html).   *Ingeschatte tijdsduur: 150 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.    Onderkant formulier |

**Bestuderen**

|  |
| --- |
| Bestudeer uit Flanagan:   * paragraaf 5.4.3 (switch) * paragraaf 5.5.2 (do/while) * paragraaf 5.6.2 (break) en 5.6.3 (continue).   *Ingeschatte tijdsduur: 60 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Bestuderen**

|  |
| --- |
| **JavaScript en het DOM** Als een browser een HTML-pagina op uw beeldscherm zet, heeft de browser van die pagina eerst een DOM gemaakt. Het Document Object Model (zie: [Wikipedia.org - Document\_Object\_Model](http://nl.wikipedia.org/wiki/Document_Object_Model)) levert een manier om de verschillende elementen (inhoud, structuur, stijl) van een webpagina op te vragen als object en ze eventueel te bewerken.  De objecten van een webpagina zijn geordend volgens een boomstructuur, zie een voorbeeld op [deze pagina](http://www.w3schools.com/js/js_htmldom.asp).  Bovenaan in de hiërarchie van de boom staat het object document. In veel voorbeelden en oefeningen hebben we al gebruikgemaakt van dit object door zijn methode getElementById aan te roepen:document.getElementById("demo"). Het document-object heeft nog veel meer methoden (en properties), zie [w3schools.com - dom\_methods](http://www.w3schools.com/js/js_htmldom_methods.asp).  Vanuit JavaScript kunt u communiceren met de browser. De communicatie verloopt via objecten; de verzameling van die objecten wordt het BOM genoemd (Browser Object Model). Hoewel dit niet is gestandaardiseerd, verloopt het in veel (moderne) browsers op dezelfde manier.  **Het window-object** Een belangrijk object is het window-object. In feite staat het window-object in hiërarchie boven het document-object en is het document een property van het window-object. Ook alle globale variabelen die u zelf in een script declareert, zijn formeel properties van het window-object. Globale functies zijn methoden van het window-object.  Een van de properties van het window-object is het beeldscherm (screen). In plaats van window.screen mag u eenvoudigweg screen schrijven. Lees meer op [w3schools.com - window](http://www.w3schools.com/js/js_window.asp).  **Het location-object** Het adres van de [webpagina](http://nl.wikipedia.org/wiki/Uniform_Resource_Locator) die in de browser te zien is, kunt u opvragen via hetwindow.location-object. Ook hier mag u window weglaten en alleenlocation schrijven. De verschillende delen waaruit de URL bestaat, zijn beschikbaar als properties:   * location.href voor de hele URL. * location.hostname voor de domeinnaam. * location.pathname voor het pad en de bestandsnaam van de pagina. * location.protocol voor het gebruikte protocol, zoals http:// of https://. * Met location.assign kunt u de browser een nieuwe pagina laten tonen, bijvoorbeeld die van uzelf als u die hebt.   Zie [w3schools.com - window location](http://www.w3schools.com/js/js_window_location.asp) voor voorbeelden.  **Het history-object** Het window.history-object bewaart de browsergeschiedenis. De statementshistory.back() en history.forward() werken als de back- en forward-knoppen van de browser.  Met history.go() kunt u meerdere stappen terug of voorwaarts (als die er zijn). history.go(-3) is bijvoorbeeld hetzelfde als drie keer op de back-knop van de browser drukken. Als het getal tussen de haakjes van groter is dan het aantal stappen dat de browser kan maken, gebeurt er niets.  Lees meer op [w3schools.com - window history](http://http/www.w3schools.com/js/js_window_history.asp).  **Het navigator-object** Het window.navigator-object geeft informatie over de browser, zoals het feit of cookies zijn toegestaan en het type van de browser.  Lees meer op [w3schools.com - window navigator](http://www.w3schools.com/js/js_window_navigator.asp).  **Overzicht en voorbeelden** Op [deze pagina](http://www.w3schools.com/jsref/obj_window.asp) kunt u een overzicht van het window-object vinden. Zie voor voorbeelden: [w3schools.com - ex browser](http://www.w3schools.com/js/js_ex_browser.asp).  *Ingeschatte tijdsduur: 120 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Oefenen**

|  |
| --- |
| Test uw parate kennis op:   * [The DOM](http://www.asmarterwaytolearn.com/js/58.html).   *Ingeschatte tijdsduur: 30 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Bestuderen**

|  |
| --- |
| Flanagan besteedt een heel hoofdstuk aan het window-object. Lees dit hoofdstuk in zijn geheel door. Veel van wat er in staat, hebt u mogelijk al gezien in de voorbeelden. Er worden ook een aantal nieuwe zaken besproken.   * Bestudeer hoofdstuk 14: The Window Object (Flanagan).   *Ingeschatte tijdsduur: 210 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Bestuderen**

|  |
| --- |
| Zoals de meeste programmeertalen beschikt JavaScript ook over wiskundige functies en constanten. Dit zijn methoden en constanten van het object Math. Er is een aantal methoden die typisch zijn voor wiskundige toepassingen, zoalsMath.sin(), Math.cos() en Math.tan() voor de sinus, cosinus en tangens.  Lees [hier](https://www5.loi.nl/access/content/group/255889f1-27c3-4e90-944f-11761ec46ec4/Bestanden%20SA/Math.pdf) meer over het object Math.  *Ingeschatte tijdsduur: 150 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Oefenen**

|  |
| --- |
| In de volgende oefenopgave past u de zojuist geleerde kennis toe. Het is erg belangrijk de oefenopgave eerst zelf te maken, voordat u de uitwerking bekijkt.  Maak opgave 11, de opgave vindt u [hier](https://www5.loi.nl/access/content/group/255889f1-27c3-4e90-944f-11761ec46ec4/Bestanden%20oefenopgaven/Oefenopgaven_hoofdstuk_4.pdf#page=4).  Controleer uzelf aan de hand van [deze uitwerkingen](https://www5.loi.nl/access/content/group/255889f1-27c3-4e90-944f-11761ec46ec4/Bestanden%20oefenopgaven/Uitwerkingen_oefenopgaven_H4.pdf#page=9).  *Ingeschatte tijdsduur: 120 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Bestuderen**

|  |
| --- |
| Flanagan wijdt een heel hoofdstuk aan reguliere expressies.   * Bestudeer hoofdstuk 10: Pattern Matching with Regular Expressions.   *Ingeschatte tijdsduur: 180 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

**Inzenden**

|  |
| --- |
| Maak uw oplossing in Komodo Edit. Splits de HTML-code en het script in twee bestanden. Test uw oplossing via KE in verschillende browsers voor u deze inzendt. Stuur alle bestanden, zowel het script als HTML, naar de LOI.  Maak de inzendopgave.  U ontvangt het beoordeelde werk binnen enkele dagen retour. Het resultaat kunt u vinden in de huiswerkresultaten.  *Ingeschatte tijdsduur: 120 minuten.*  Bovenkant formulier  **Status**  Klik Voltooien als u deze activiteit hebt afgerond.  Onderkant formulier |

* [Inzendopgave 242S3](https://www5.loi.nl/portal/tool/b4bd16be-c04d-4da9-829a-6131f20245ac/?userId=f15516b9-d75c-4412-9fef-6401482740e1&learningActionId=61301&modulePLSiteId=255889f1-27c3-4e90-944f-11761ec46ec4&moduleTitle=Programmeren%20in%20JavaScript&externalActionEntitylId=81928&panel=Main)

Maak een pagina met drie invoervakken en een knop.

* Eén invoervak is voor uw geboortedatum. Laat deze datum controleren op mogelijke onjuistheden.
* De overige twee invoervakken zijn voor een bedrag en een rentepercentage (per jaar). Controleer ook hier op mogelijke onjuistheden.

Als de drie zaken correct zijn ingevoerd en u klikt op de knop, dan verschijnt op de pagina een overzicht van het saldo de komende jaren totdat het bedrag is verdubbeld, als u het op de bank zet tegen het ingevoerde rentepercentage.

**antwoord**