

# Modulewijzer

---

**MME 1 lente 2008**

## ***Basisprincipes van het programmeren***

Communication & Multimedia Design

**Auteur:**

Roos Groenewegen

**Docenten:**

Roos Groenewegen	Klas 1C
Barend Hendriks	Klas 1F
Arjo Post	Klas 1D, 1E
Michiel Mobach	Klas 1A, 1B

## **Titel: De basisprincipes van het programmeren**

**Code: CMDMME1L**

**ECTS: 1**

### **Competenties, Indicatoren en leerdoelen. (interface developer)**

*De student beheerst de basisprincipes van het programmeren [ niveau 1 ]*

- Je kent de basis begrippen van het programmeren: variabele, conditional statement, loop, array, functie.
- Je kunt de begrippen uit het bovenstaande leerdoel toepassen in eenvoudige vraagstukken.

*De student test het product op bugs en hij is in staat om eventuele bugs systematisch te vinden en op te lossen, waardoor hij het prototype volledig werkend kunt opleveren. [ niveau 1 ]*

- Je kunt een probleem opdelen in deelproblemen om het vraagstuk te vereenvoudigen.
- Je hebt inzicht in de wijze waarop een computer omgaat met informatie.
- Je kunt, wanneer er nog fouten in de code zitten, systematisch de code doorlopen om de fout alsnog te ontdekken.
- Je geeft, wanneer het niet lukt, niet meteen op maar kent het internet of boeken als een hulpbron.

*De student is in staat om een analyse te maken van alle taken die de gebruiker moet gaan kunnen uitvoeren. [ niveau 1 ]*

- Je bent in staat een opdracht te vertalen naar deeltaken van de gebruiker.
- Je bent in staat om deze deeltaken schematisch weer te geven in de vorm van een schema.

### **Korte beschrijving:**

Een HTML pagina kun je veel extra functionaliteit geven met behulp van scripts, bijvoorbeeld JavaScript. In deze periode gebruiken we deze taal om de basisprincipes van het programmeren uit te leggen. Als je de basis van programmeren snapt, worden alle volgende periodes een stuk gemakkelijker. Tijdens deze periode wordt er veel van je verwacht. Een hoge mate van inzet is essentieel. De volgende onderwerpen komen aanbod:

- HTML formulieren en CSS formulieropmaak.
- Inleiding programmeren; doel en context, serverside vs clientside, variabelen.
- Functionele flowcharts, UML.
- JavaScript; conditional statement, operatoren, loops, random, arrays en functions.
- JavaScript; rollovers, verschillende browsers, formvalidatie.

### **Werkvormen**

De lessen starten met kennisoverdracht, waarna je al snel zelf aan de slag gaat in de vorm van een kleine lesopdracht. Op deze manier wisselen kennisoverdracht, practicum en hulp bij opdrachten elkaar af. Daarnaast wordt er 2 keer per kwartaal in de les aandacht aan het project besteed.

### **Wijze van toetsing en beoordelingscriteria**

Je wordt getoetst aan de hand van opdrachten die je in de les en thuis maakt. Deze opdrachten zijn kleine oefeningen gebonden aan het thema van de les. Als je de stof al kent wordt er van je verwacht dat je dit eerst aantoonst in de les door het maken van een extra opdracht. Daarna kan er in overleg een opdracht op eigen niveau worden geformuleerd. De opdrachten worden beoordeeld aan de hand van de algemene criteria.

### **Wijzen van herkansen**

Je kunt dit vak herkansen volgens de algemene herkansingsprocedure.

### **Benodigd studiemateriaal (boeken e.d.)**

- 'JavaScript for the World Wide Web; Tom Negrino, Dori Smith (Engels, vijfde editie, ISBN: 032119439X)
- Basiscursus JavaScript 1.5; Peter Kassenaar (Nederlands, ISBN: 9039519668)

## Planning lessen

week	les	titel	opmerkingen	opdrachten
35	1	HTML Formulieren en CSS formulieropmaak	Briefing Project 1	
36	2	Inleiding programmeren in JS		
37		Voorjaarsvakantie		
38	3	Functionele flowcharts / UML		
39	4	Conditional Statements, operatoren		
40	5	Loops, Random, Array's		
41	6	Browser compatability, Formvalidatie		
42	7	HTML formulieren		
43	8	Huiswerkbegeleiding		
45		Vakbeoordelingen	Inleveren eind week 44	
46		Projectbeoordelingen	geen les	

### Wijze van beoordelen

Wanneer je extra werk hebt gemaakt tijdens de lessen, plaats ook deze online, dan wordt deze meegenomen in je beoordeling. Extra werk kan de vaste opdrachten niet compenseren, maar kan wel extra tellen.

Wanneer een opdracht niet gelukt is, dan wordt de wijze waarop je het hebt aangepakt beoordeeld:

- Heb je een flowchart gemaakt van hoe de structuur van je script zou moeten zijn?
- Heb je een analyse gedaan van de deelproblemen? En hoe je deze zou moeten oplossen? (dit kan gewoon op papier, een soort van geschreven oplossing, waarbij je nog geen code gebruikt)
- Heb je hulp gezocht?

Deze beschrijving van je aanpak levert geen voldoende op, maar kan je verlichting van de herkansing en extra hulp opleveren in de periode erna.

### Wijze van inleveren

Daarnaast gelden de algemene regels wat betreft becijfering, herkansingen en aanwezigheid.

Specificatie inleveren, locatie en lay-out

Nummer les	Titel opdracht	Link naar resultaat
		reflectie over hoe de opdracht is gegaan, hoeveel tijd er in is gaan zitten en waar je tegenaanbent gelopen, hoe je evtuele problemen hebt opgelost
Nummer les	Titel opdracht	Link naar resultaat
		reflectie over hoe de opdracht is gegaan, hoeveel tijd er in is gaan zitten en waar je tegenaanbent gelopen, hoe je evtuele problemen hebt opgelost
<b>Link naar een zipfile met de source van het totaal van alle opdrachten</b>		

Maak een html-pagina met een overzicht van alle opdracht. Vermeld hierbij de les, de titel van de opdracht en een link naar het eindresultaat. Tekstdocumenten bijvoorbeeld in pdf. Een toelichting op de opdrachten over hoe je te werk bent gegaan altijd op de regel er onder toevoegen. Naast eindresultaat moet je ook de sourcedocumenten online zetten. Voeg alle sourcefiles gezamenlijk in een zip document.

### Locatie

[http://student.cmd.hro.nl/\[studnr\]/jaar1/herfst/mme/](http://student.cmd.hro.nl/[studnr]/jaar1/herfst/mme/)

### Algemene beoordelingscriteria

Voor alle lessen in het eerste en tweede jaar van MME gelden minimaal de onderstaande beoordelingscriteria:

- Je hebt je opdrachten op tijd en volgens de specificaties ingeleverd.
- Je voldoet in voldoende mate aan de leerdoelen vooraf gesteld.

- Je hebt voldoende opdrachten per les tot en goed einde gebracht.
- Je hebt jezelf uitgedaagd.
- Je hebt bij problemen hulp gevraagd tijdens de lessen.
- Je bent op tijd begonnen met het maken van de opdrachten.
- Je bent gegroeid in je probleem oplossend vermogen.
- Je documentatie is op tijd, volledig en verzorgd ingeleverd.
- Je hebt de code qua naamgeving en commentaar voldoende verhelderd.
- Het resultaat van de opdrachten er netjes en verzorgd uit. (alleen correcte code is te minimaal)

# **Les 1 | HTML Formulieren en CSS formulieropmaak**

Week 36

## **Leerdoelen**

- Inzicht gebruik van formulieren in een html pagina
- Noodzaak gebruik van scripting talen binnen html pagina's met formulieren
- Syntax diverse form tags

## **Te behandelen onderwerpen**

Iets wat nog niet behandeld is in de html lessen is formulieren. Een formulier is ook een beetje bijzonder in html. Een formulier is bedoeld om gegevens te versturen en dat is iets wat html niet kan. Je zal een formulier dus altijd in combinatie met andere scripttalen gebruiken. Formulieren worden behandeld aan de hand van de volgende onderwerpen: Syntax, tekstvelden, Tekstkaders, Checkboxes, Radiobuttons, Dropdown menu, List, Buttons, Formulier kaders, de koppeling met javascript, Submit, post en get.

## **Werkvorm**

De les bestaat voor een groot deel uit klassikale instructie. De docent toont op welke wijze een formulier in elkaar zit en beschrijft de verschillende onderdelen. Daarna mogen de studenten zelf aan de slag met de opdracht.

## **Opdracht L1**

Je mag bij deze les kiezen of je oefening 1 en 2 uitvoert of anders alleen oefening 3. Oefening 3 is echter wel iets moeilijker. Deze oefeningen staan beschreven in de opdracht.

## **Toetsing**

De leerdoelen worden getoetst doormiddel van de opdracht behorende bij deze les.

## **Bronnen**

[http://www.w3schools.com/html/html\\_forms.asp](http://www.w3schools.com/html/html_forms.asp)

## **Les 2 | Inleiding programmeren in JS**

Week 37

### **Leerdoelen**

- Je kent de taal javascript en hebt een indruk van wat je er mee kan
- Je begrijpt wat programmeren is en wat er dit kwartaal van je verwacht wordt
- Je weet wat het verschil is tussen een variabele in javascript en in flash
- Je hebt een idee van de overeenkomsten tussen verschillende talen.

### **Te behandelen onderwerpen**

Waarom javascript, Verwachtingen, waarom programmeren, Wat is programmeren? Wat is javascript? javascript en HTML, Variabelen in javascript, het verschil met Flash, Invoer en uitvoer van data, een aantal handvaten voor de aanpak bij programmeren en als laatste het debuggen.

### **Werkvorm**

De les bestaat voor een groot deel uit college. De docent introduceert de taal javascript. Daarna laat hij zien wat er mogelijk is en mogen de studenten zelf aan de slag met de opdrachten.

### **Te behandelen opdracht**

**Opdracht L2:** Deze opdracht bestaat uit 11 opdrachten. Aan elke opdracht zijn punten gekoppeld. In totaal dien je minimaal 10 punten te behalen. Je mag zelf kiezen welke opdrachten je doet, mits je uiteraard de benodigde punten behaalt. Indien je een beginner bent, begin dan met de eenvoudige opdrachten, maar daag jezelf ook uit en probeer ook de eens moeilijkere opdrachten te volbrengen. De punten geven tevens het niveau aan van de opdracht: 1 punt staat voor het niveau van de beginner, bij 5 punten mag je jezelf guru noemen.

### **Toetsing**

De leerdoelen van deze les worden getoetst doormiddel van de opdrachten die bij deze les horen.

### **Bronnen**

Learn javascript: <http://www.w3schools.com/js/default.asp>

## **Les 3 | Functionele flowcharts / UML**

Week 38

### **Leerdoelen**

De student kan een goede flowchart ontwikkelen, waarmee hij structuur en keuzemogelijkheden inzichtelijk kan maken. Hierbij maakt hij onderscheid in ingelegd gedrag en het gedrag van de gebruiker.

### **Te behandelen onderwerpen**

UML staat voor **unified modeling language**. En is een verzamelnaam voor allerlei technieken voor het visueel documenteren van processen. Zo'n document noemen wij een flowchart. Omdat flowcharts in heel veel verschillende bedrijfstakken voor allerlei verschillende processen worden gebruikt, bestaat er behoefte aan specifieke symbolen voor veel voorkomende oplossingen. Er zijn dus allerlei verschillende definities van UML. Elke bedrijfstak heeft zijn eigen specificatie. Voor het documenteren van computer systemen bestaat ook een vaste set aan iconen. Wij gaan gebruik maken van het iconen set zoals dat ook op de site van garret beschreven is: [www.jjg.net/ia/visvocab](http://www.jjg.net/ia/visvocab)

### **Werkvorm**

De les bestaat voor een groot deel uit een college waarin de docent toont op welke wijze je een flowchart realiseert en gebruikt. Docent beschrijft de betekenis van de verschillende onderdelen. Daarna mogen de studenten zelf aan de slag met de opdrachten.

### **Te behandelen opdracht**

Opdracht L3: Flowchart

### **Toetsing**

De leerdoelen van deze les komen terug in de opdracht die bij deze les hoort.

### **Links**

Website James Garret: [www.jjg.net/ia/visvocab](http://www.jjg.net/ia/visvocab)

## **Les 4 | Conditional Statements, Operatoren**

Week 39

### **Leerdoelen**

- De student weet waar if else constructs voor bedoeld zijn
- De student snapt hoe if else constructs gebruik worden en wanneer
- De student kan een eenvoudige if else construct maken/ toepassen

### **Te behandelen onderwerpen**

Als we aan het programmeren slaan dan doen we dit vanuit een aantal redenen.

- Programmeurs denken liever dan ze doen (je mag dit lui noemen)
- Computer zijn vele malen beter in doen van herhalings taken dan mensen.

Voorals we systemen gaan bouwen die groot zijn (zeg een pagina of 100) dan is het handig dingen te automatiseren (dus scripten). Daarnaast kun je met scripts van allerlei dingen testen (Bijvoorbeeld of iemand wel een formuler naar behoren heeft ingevoerd! Bij de laatste komen if - else constructs naar voren. Scripten is het beschrijven wat een systeem moet doen aan de hand een programma of script wat je schrijft. Dit programma ook wel een algoritme genoemd, loopt staps gewijs allerlei beslissingen af (aan de hand wat een gebruiker wil). Hier worden dus ook beslissingen gemaakt en ta dah: if - else constructs.

### **Werkvorm**

De les bestaat voor een groot deel uit klassikale instructie. De docent toont op welke wijze je conditional statment gebruikt en beschrijft de verschillende onderdelen en het verschil met flash. Daarna mogen de studenten zelf aan de slag met de opdrachten.

### **Te behandelen opdracht**

Opdracht L4: MiniQuiz

### **Toetsing**

De leerdoelen van deze les komen terug in de opdracht die bij deze les hoort.



## **Les 5 / Loops, Random & Arrays**

Week 40

### **Leerdoelen**

- De student weet waar loop constructs voor bedoeld zijn
- De student snapt hoe loop constructs gebruikt worden en wanneer
- De student kan een eenvoudige loop construct maken/ toepassen
- De student maakt kennis met het Math object

### **Te behandelen onderwerpen**

- Huiswerk (problemen) bespreken.
- Loop statement: (for...next)
- Loop statement: (while)
- Math object

### **Werkvorm**

De les bestaat voor een groot deel uit klassikale instructie. De docent toont op welke wijze je een loop gebruikt en beschrijft de verschillende onderdelen. Daarna mogen de studenten zelf aan de slag met de opdrachten en is er tijd voor individuele begeleiding van de huiswerkopdrachten.

### **Te behandelen opdracht**

Opdracht L5: Random image

### **Toetsing**

De leerdoelen van deze les komen terug in de opdracht die bij deze les hoort.

## **Les 6 / Arrays en Functions**

Week 41

### **Leerdoelen**

- De student weet waar arrays voor bedoeld zijn
- De student weet welke arrays er gebruikt worden
- De student kan arrays maken en uitlezen met een loop constructie
- De student snapt de samenhang tussen Javascript en HTML;
- De student weet wat Javascript build-in functions zijn en kan deze toepassen;
- De student snapt wat een Javascript custom-made function is en kan deze maken en toepassen;
- De student snapt wat Javascript events zijn en kan deze toepassen;

### **Te behandelen onderwerpen**

Wat is een array? Toepassingen, Syntax, Voorbeeld Arrays in een Array, Array.length, Indexed arrays (+voorbeelden), Associative arrays (+voorbeelden), multi-dimensional arrays.

Wat is een functie, DOM en javascript, build-functies, paden en elementen, Paden en propertie, Events, Eenvoudige zelfgeschreven functies, Pre-defined functions, Volgorde van lezen van functies.

### **Werkvorm**

De les bestaat voor een groot deel uit klassikale instructie. De docent toont op welke wijze je de verschillende type array's gebruikt en beschrijft de verschillende onderdelen. In een tweede college van 20 minuten worden functies behandeld. Daarna mogen de studenten zelf aan de slag met de opdrachten en is er tijd voor individuele begeleiding van de huiswerkopdrachten.

### **Te behandelen opdracht**

Bij deze les behoort geen nieuwe opdracht.

### **Toetsing**

De leerdoelen van deze les komen terug in de voorgaande opdrachten en worden beoordeeld aan het einde van de periode.

## **Les 7 / Browser compatability, Formvalidatie**

Week 42

### **Leerdoelen**

- De student heeft inzicht in de wijze waarop javascript binnen een browser funtioneeert.
- De student weet dat een script soms aangepast moet worden om in verschillende browsers te kunnen werken.
- De student kan alle formulier elementen uitlezen met javascript en een formulier controleren op volledigheid en juistheid van invoer.

### **Te behandelen onderwerpen**

Javascript werkt in browsers. Dit betekent dat javascript ook samen werkt met HTML. Bijna alle eigenschappen van HTML zijn te bewerken met javascript en dat kan tot leuke dingen leiden. Daarnaast zijn er in de loop van de tijd wat steeds meer features bij gekomen. Helaas zijn er verschillen tussen browsers, maar we gaan met deze lessen niet diep genoeg om daar tegenaan te lopen. We gaan niet in op zeer incurante browsers. Deze les geeft een indruk waar de verschillen zitten.

### **Werkvorm**

De les bestaat voor een groot deel uit college over dit onderwerp. De docent geeft voorbeelden en laat de hindernissen zien. Deze les is ter voorbereiding van de komende periode een geeft een indruk van de voor en nadelen van javascript. De tweede helft van de les is er tijd voor individuele begeleiding bij vragen met betrekking tot het huiswerk.

### **Te behandelen opdracht**

Bij deze les is geen opdracht.

### **Toetsing**

De lesstof van deze les wordt gedeeltelijk getoetst door voorgaande opdrachten. Andere onderdelen zullen in de volgende periode terug komen.

## ***Les 8 / Huiswerkbegeleiding***

Week 43

### **Leerdoelen**

Deze les heeft geen leerdoelen

### **Te behandelen onderwerpen**

In deze les worden geen onderwerpen behandeld

### **Werkvorm**

Tijdens de les is er begeleiding per student bij de realisatie van de opdrachten. Wanneer een student al zijn opdrachten heeft gerealiseerd en deze online klaar heeft staan ter beoordeling voor de docent, kan hij na toestemming van de docent de les eerder verlaten.

### **Te behandelen opdracht**

Er zijn geen opdrachten bij deze les

### **Toetsing**

Deze les is alleen ondersteunend en geeft inzicht in de werkwijze en defficienties bij de studenten. Er wordt verder niet getoest.

## ***Bijlage opdrachten***

### **Opdracht L1:**

Je mag bij deze les kiezen of je oefening 1 en 2 uitvoert of anders alleen oefening 3. Oefening 3 is echter wel iets moeilijker

#### **Oefening 1**

Maak een formulier waarbij een gebruiker zijn naam kan invullen en zijn leeftijd.

- Voor het veld naam gebruik je een tekstveld van 50 pixels breed, waarbij maximaal 50 characters zijn in te vullen.
- Voor het veld leeftijd maak je een selectbox met daarin een opsomming van 18-65 jaar.

Maak ook gebruik van labels om aan te geven wat er moet worden ingevuld. (naam en leeftijd)

Na het invullen geef je de ingevulde waardes weer zoals in [voorbeeld 2](#) uit de lesbrief.

#### **Oefening 2**

Maak bovenstaand formulier zodanig dat de in te vullen velden worden gecontroleerd:

- naam = verplicht
- leeftijd = verplicht en moet numeriek waarde zijn tussen 18 en 65

Als 1 van beide velden niet of niet goed is ingevuld dan laat je een informatieve foutmelding zien.

#### **Oefening 3**

Maak een formulier waarin je de volgende zaken moet invullen:

- naam (tekstveld 50 pixels max 50 characters, verplicht)
- geboortedatum (aparte selectboxes voor dag, maand en jaar, verplicht)
- geslacht (radiobuttons m/v, verplicht)

Toon de ingevulde waardes na versturen van het formulier in een resultaatscherm.

Reken de leeftijd uit ahv. de ingevulde geboortedatum en toon deze ook in het resultaatscherm.

Toon bij foutief invullen het formulier opnieuw met een foutaanduiding achter het betreffende veld.

Biedt de gebruiker de mogelijkheid om alleen foutief ingevulde velden opnieuw in te vullen.

Dus niet alles opnieuw invullen!

## Opdracht L2

Hieronder 11 opdrachten. Aan elke opdracht zijn punten gekoppeld. In totaal dien je minimaal **10 punten** te behalen. Je mag zelf kiezen welke opdrachten je doet, mits je uiteraard de benodigde punten behaalt. Indien je een beginner bent, begin dan met de eenvoudige opdrachten, maar daag jezelf ook uit en probeer ook de eens moeilijkere opdrachten te volbrengen. De punten geven tevens het niveau aan van de opdracht: 1 punt staat voor het niveau van de beginner, bij 5 punten mag je jezelf guru noemen.

### Opdracht 01 - Smileys (2 punten)

Bij veel fora en in msn wordt gebruik gemaakt van smiley's, bij een bepaalde volgorde van tekens, worden de tekens omgezet in de plaatje. Zorg dat je middels een prompt een smiley kan invoeren. Deze smiley wordt vervolgens in de html weergegeven als een plaatje van een smiley met de juiste uitdrukking. Maak deze opdracht voor minimaal 3 type smiley's werkend.

### Opdracht 02 - Seizoenen (3 punten)

Om een site een persoonlijker karakter te geven, zou je deze kunnen aanpassen aan gegevens die je van de bezoeker weet. Je kan middels een prompt aan de bezoeker zijn geboortemaand in cijfers vragen. Aan de hand van de maand die ingevoerd wordt, wordt er boven in de pagina een plaatje geladen die past bij het seizoen waarin de bezoeker geboren is.

Wanneer er 1,2 of 3 is ingevoerd wordt het een plaatje van de winter, 4,5 of 6 wordt lente, 7,8 of 9 wordt zomer, 10,11 of 12 wordt herfst.

### Opdracht 03 - Boete (1 punt)

Weggebruikers die net geflitst zijn, zijn vaak toch een beetje geïrriteerd. Voor de website [www.flitspaal.nl](http://www.flitspaal.nl) wil je een nieuwe service bieden. De bezoeker kan tijdens het openen van de website meteen de snelheidsovertreding invoeren, hieruit volgt dan de boete. De kan je aanpakken door met behulp van een prompt het aantal kilometers dat er te hard werd gereden in te laten vullen. Aan de hand van de ingevoerde gegevens wordt er een mededeling weer gegeven op de html pagina over de hoogte van de te betalen boete. 5 km te snel geeft 40 euro boete, 10 km te snel geeft 80 euro boete, 20 km te snel geeft 160 euro boete en meer dan 50 km te snel betekend dat het rijbewijs wordt ingenomen.

### Opdracht 04 - Boete uitbreiding (4 punt)

Zie bovenstaande opdracht, maar nu wordt er middels een prompt gevraagd hoe hard er werd gereden in te vullen, en een tweede prompt vraagt wat de maximum snelheid is. Aan de hand van de ingevoerde gegevens wordt er uitgerekend hoeveel er te hard is gereden. Daarna wordt er aan de hand van dit resultaat een mededeling weer gegeven op de html pagina over hoe hard er gereden werd, hoe hard je zou mogen rijden en de hoogte van de te betalen boete. 5 km te snel geeft 40 euro boete, 10 km te snel geeft 80 euro boete, 20 km te snel geeft 160 euro boete en meer dan 50 km te snel betekend dat het rijbewijs wordt ingenomen.

### Opdracht 05 - Empty user-name (1 punt)

Bij het openen van de pagina, toon je een prompt() aan de bezoeker om zijn/haar naam in te vullen. Nadat de gebruiker zijn naam heeft ingevuld, wordt de volgende tekst in het HTML document getoond, bijvoorbeeld: 'Welkom Bas, Leuk dat je er bent'. Indien de bezoeker geen naam invult, dan wordt de volgende boodschap in het HTML document getoond: 'Je hebt je naam niet ingevuld!'. Tevens wordt deze boodschap in een alert getoond (let op, als de gebruiker wel een naam heeft ingevuld, wordt er geen alert getoond).

### Opdracht 06 - Prototype: Pinautomaat 01 (2 punten)

Stel je voor dat je een prototype moet maken voor een pinautomaat. Het betreft het opvragen van een bedrag, er dient voldoende saldo zijn. Definieer een variabele voor het saldo. Prompt de gebruiker (bij het openen van de pagina) om een bedrag in te vullen, met de vraag: 'Welk bedrag wil je opnemen?'. Controleer vervolgens of het saldo toereikend is. Indien het saldo voldoende is, toon dan een plaatje met geldbiljetten. Indien het saldo niet voldoende is, toon dan een bericht (alert): 'Sorry, onvoldoende saldo,

je hebt momenteel € 211,45 op je rekening staan.' (het saldo wat hier vermeld staat, dien je dus te vervangen voor het saldo dat jij hebt opgegeven).

#### **Opdracht 07 - Prototype: Pinautomaat 02 (5 punten)**

Je hebt de het vorige prototype met succes aan je klant gepresenteerd. De product manager heeft zelfs nog een leuke feature bedacht: 'een persoonlijke welkomst boodschap'. Om de werking van deze feature te demonstreren, dien je een prototype te maken. Bij het openen van de pagina, wordt de gebruiker eerst gevraagd zijn/haar rekeningnummer en pincode in te voeren (dus twee prompts) in te voeren. Vervolgens verschijnt een prompt met de tekst 'Welkom Bob, Hoeveel geld wil je opnemen? (je kunt maximaal € 231.45 opnemen)' (het bedrag in dit voorbeeld vervang je uiteraard voor het saldo van de gebruiker). Voer een bedrag in:.

Uiteraard controleer je nog even of de gebruiker wel voldoende saldo heeft. Indien het saldo voldoende is, toon dan een plaatje met geldbiljetten. Indien het saldo niet voldoende is, toon dan een bericht (alert): 'Sorry, onvoldoende saldo, je hebt momenteel € 211,45 op je rekening staan.' (het saldo wat hier vermeld staat, dien je dus te vervangen voor het saldo dat jij hebt opgegeven).

Zorg dat het prototype voor 3 gebruikers werkt:

Rekeningnr	Pin	Naam	Saldo
3421	21	Bob	231.45
8900	89	Diana	11.91
7731	73	James	890.34

Tip: gebruik variabelen als account01, account02, account03 en daarbij horen dan userName01, userName02, userName03 en een saldo.

#### **Opdracht 08 - Cijferlijsten (2 punten)**

Veel studenten blijven altijd omrekenen van het Europese beoordelingssysteem naar het Amerikaanse systeem (voor de docenten die nog Europese cijfers geven). Maak voor deze studenten een omreken calculator die de Europese cijfers omzet naar de Amerikaanse equivalenten. De verdeling van de cijfers is als volgt.

0 - 1 E  
2 - 3 D  
4 - 5 C  
6 - 7 B  
8 - 9 A

#### **Opdracht 09 - Check age (1 punt)**

Bij het openen van de pagina, wordt de bezoeker gevraagd zijn/haar leeftijd in te voeren. Indien de bezoeker jonger is dan 18, toon dan het bericht 'Acces denied, grow up!' zowel in een alert als in het HTML document.

#### **Opdracht 10 - Check username & password (2 punten)**

Bij het openen van de pagina, wordt de bezoeker gevraagd zijn/haar user-id en password in te voeren. Bedenk zelf een user-id en password, indien de gebruiker een verkeerd user-id of password invoert, toon het bericht: 'Acces denied! Invalid user-id or password!' (in een alert en in het HTML document), anders toon het bericht 'Welkom Clark!' (waarbij de 'Clark' vervangen wordt door de gebruikers naam die is ingevoerd).

#### **Opdracht 11 - Check username, password, age (3 punten)**

Bij het openen van de pagina, wordt de bezoeker gevraagd zijn/haar leeftijd in te voeren. Indien de bezoeker jonger is dan 18, toon dan het bericht 'Acces denied, grow up!' zowel in een alert als in het HTML document. Indien de bezoeker ouder is dan 18 prompt dan de gebruiker voor zijn user-id en password (let op: user-id en password kunnen pas ingevoerd worden als de bezoeker 18 jaar of ouder is). Indien de gebruiker een verkeerd user-id of password invoert, toon het bericht: 'Acces denied!'

Invalid user-id or password!' (in een alert en in het HTML document), anders toon het bericht 'Welkom Clark!' (waarbij de 'Clark' vervangen wordt door de gebruikers naam die is ingevoerd). De volgende 3 gebruikers hebben toegang tot de website:

User-id	Password
Clark	Lane
Lois	Kent
Steve	Jobs

**Experts:** Is dit allemaal 'piece of cake' ? Probeer deze opdracht op te lossen door een array te gebruiken, of wellicht een multi-dimensional array.

### Opdracht L3: Flowchart

- installeer VISIO op je computer (kun je gratis downloaden bij MSDN Academic Alliance, zie eerder emailtje wat je hebt ontvangen van systeembeheer)
- installeer de JJG shape libraries die je kunt downloaden via <http://jgg.net/ia/visvocab/#download> in Visio en Powerpoint.
- maak een flowchart (stroomdiagram) in Visio van je portfolio website met deze gedownloade set van symbolen. Voor uitleg betekenis symbolen zie lesbrief; opslaan flowchart op je computer als .vsd (net zoiets als psd in Photoshop); nog eens opslaan als webpage, welke je kunt publiceren op je webspace. (geen vsd bestanden op de server, alleen lokaal)
- maak van punt 1 tm 3 een reflectieverslag en zet alles online op octagon (CMD server) vóór de 2e les in de lente via <jouw cmd webspace>/jaar1/lente/mme/index.htm = opleverpagina lente)

### Opdracht I4: MiniQuiz

Maak een mini-quiz waarbij je een 5-tal vragen stelt aan een gebruiker.

Verplichte invoervelden zijn:

- naam
- leeftijd
- man/vrouw

Daarna toon je een 5-tal vragen. De antwoorden hierop worden als multiple-choice antwoorden (a/b/c/d) getoond, elk voorzien van een radio-button. Het aanklikken van een radiobutton bij iedere vraag is verplicht. Kenmerkend van een radio-button is dat er altijd maar 1 goed antwoord mogelijk is.

Om elke vraag met bijbehorende antwoorden-set maak je een kader met een legenda, waarin de tekst vraag staat gevolgd door het vraagnummer. Je mag zelf bepalen wat de vragen en de antwoorden zijn, maak er maar iets moois van! Maak gebruik van een html formulier om de invoer te verzorgen. Bevestig de invoer met een submit button.

Bouw de volgende controles in het formulier in:

- Geldige invoer bij naam: een tekst met een lengte van minimaal 3 karakters. Het is niet toegestaan om nummers en leestekens in te vullen. Ook is het niet toegestaan niets in te vullen, of om alleen maar spaties in te vullen.
- Geldige invoer bij leeftijd: een getal groter dan 18, maar niet groter dan 65. Dus van 18 tot en met 65 is een geldig getal. Het is niet toegestaan om leestekens of letters in te vullen. Ook is het niet toegestaan niets in te vullen of om spaties in te vullen.
- Controleer of alle vragen zijn beantwoord. Bij elke vraag moet er een keuze worden gemaakt.



- Controleer hoeveel punten je hebt gescoord. Ieder goed antwoord is 1 punt waard. Aan de hand van het totaal bepaal je het nivo van de gebruiker:
  - o Bij een getal tussen de 0 en 1 is het level: looser
  - o Bij een getal tussen de 2 en 3 is het level: rookie
  - o Bij een getal tussen de 3 en 4 is het level: pro
  - o Bij een getal tussen de 4 en 5 is het level: guru
- Als het nivo pro of guru is bereikt, ben je geslaagd voor de quiz. Je maakt nu een samenvatting van alle invoer en deze presenteert je in de vorm van een html pagina aan de gebruiker, waarbij een diploma wordt gegenereerd welke je via een button 'print' kunt afdrucken op een printer. Dit diploma ontwerp je zelf, hoe origineler, hoe beter!  
Als het nivo looser of rookie is bereikt, ben je gezakt. Dit laat je aan de gebruiker duidelijk zien.
- In beide gevallen toon je een html pagina: het level-lijstje en je geeft hierin grafisch (bijv. met een pijl) weer op welk nivo de gebruiker is geëindigd.
- een button 'opnieuw' om de quiz opnieuw te starten, waarbij je met een nieuw vragenformulier begint. Naam, leeftijd en geslacht hoeft je hierbij niet opnieuw in te vullen!

ALLE HTML OPMAAK BRENG JE ONDER IN EEN EXTERN CSS STYLESHEET.

Alle controles doe je met behulp van javascript. Demonstreer hierbij het if...else statement. Maak ook (op een juiste manier) gebruik van variabelen.

## Opdracht L5: Random foto in webpagina

Maak een webpagina waarbij een willekeurige foto wordt getoond bij het laden van de webpagina. Volg onderstaande stappen:

- verzamel of maak 10 foto's en plaats deze in een map
- hernoem de foto's naar foto1.jpg tm foto10.jpg
- maak een html pagina en noem deze foto.htm
- plaats foto.htm in dezelfde map als waar de foto's staan
- maak een array aan en vul deze met de fotonamen. Dit kan op 2 manieren:
  - makkelijk: iedere array-insert uitschrijven (dus 10 keer)
  - iets moeilijker: met behulp van een for loop
- bepaal een random nummer (van 0 tot 10) en bewaar deze in een variabele
- toon met behulp van de random variabele uit stap 6 de bijbehorende foto uit de array met document.write

## Opdracht L6: Dropdownmenu

Maak een webpagina met twee dropdown menus boxes.

Beiden wordt tijdens het laden (onLoad) van de pagina gevuld met gegevens.

Gebruik als inhoud voor de dropdowns een multidimensionele array met daarin de studiejaren en bijbehorende projecten.

studiejaar  (id="years")

De tweede dropdown wordt opnieuw gevuld als de eerste dropdown (years) wordt gewijzigd, dus met het onChange event. Onderstaande dropdown (select) bevat de projecten van het eerste jaar. Deze zijn hier in deze pagina al uitgeschreven mbv. de option tag.

project  (id="project")

Probeer dit nu eens op een andere manier, met javascript, te bewerkstelligen. Maak hierbij gebruik van de DOM structuur. Het is een lastig klusje maar als je de volgende stappen volgt dan is het te doen.

### Maak eerst een multidimensionale array.

Noem deze yearsprojects:

```
var yearsprojects = new Array();
```

**Vul de array met gegevens.** Deze gegevens bevatten nog geen echte projectnamen, maar laten wel zien wat er moet staan. Dit kun je later wijzigen.

```
yearsprojects[0] = new Array("jaar 1", "jaar1project1", "jaar1project2",  
"jaar1project3", "jaar1project4");  
yearsprojects[1] = new Array("jaar 2", "jaar2project1", "jaar2project2",  
"jaar2project3", "jaar2project4");  
yearsprojects[2] = new Array("jaar 3", "jaar3project1", "jaar3project2",  
"jaar3project3", "jaar3project4");  
yearsprojects[3] = new Array("jaar 4", "jaar4project1", "jaar4project2",  
"jaar4project3", "jaar4project4");
```

**Test nu eerst met een geneste loop of je de inhoud van de arrays zichtbaar kunt maken:**

```
for (i=0; i<yearsprojects.length;i++) {  
  for (j=0;j<yearsprojects[i].length;j++) {  
    document.write ('entry: ' + i + ' inhoud: ' + j + ': ' +  
yearsprojects[i][j] + '<br>' );  
  }  
}
```

**Maak 2 dropdown selectboxes, 'years' en 'projects' in een <form> ergens in de <body>.** In deze <select> boxes hoef je géén <option> tags te plaatsen, dit ga je straks met javascript doen. De <form> hoef je ook geen action en naam te geven.

```
<select id="years"></select>  
<select id="projects"></select>
```

**Maak voor het verwijderen en toevoegen van options in een <select> 4 functies:**

```
//1. toevoegen options  
function appendOptions(key,id) {  
  var elOptNew = document.createElement('option');  
  elOptNew.text = key;  
  elOptNew.value = key;  
  var elSel = document.getElementById(id);  
  try {  
    elSel.add(elOptNew, null); // standards compliant; doesn't work in IE  
  }  
  catch(ex) { //catch exeptions  
    elSel.add(elOptNew); // IE only  
  }  
}  
  
//2. verwijderen options  
function removeOptions(id) {  
  var elSel = document.getElementById(id);  
  for (i = elSel.length - 1; i>=0; i--) {  
    elSel.remove(i);  
  }  
}  
  
//3. maak options voor years
```

```

function makeYears() {
    removeOptions('years');
    for (i=0;i<yearsprojects.length;i++) {
        appendOptions(yearsprojects[i][0], 'years');
    }
}

//4. maak options voor projects van selected year
function makeProjects() {
    removeOptions('projects');
    var elSel = document.getElementById('years');
    for (i=0; i<elSel.length; i++) {
        if (elSel.options[i].selected) {
            for (j=1;j<yearsprojects[i].length;j++) {
                appendOptions(yearsprojects[i][j], 'projects');
            }
        }
    }
}

```

**Bovenstaande functies kun je nu aanroepen.**

Zo kun je bij het laden van de pagina 'years' en 'projects' al vullen.

Hierbij maak je gebruik van het onLoad event:

```
<body onLoad="javascript:makeYears();makeProjects();">
```

**Daarna kun je, iedere keer als de selectbox 'years' wijzigt, de bijbehorende projecten er bij halen.**

Dit doe je mbv. het onChange event:

```
<select id="years" onChange="javascript:makeProjects();"></select>
```

Probeer met behulp van de bovenstaande informatie een werkende pagina te maken. (2 punten)

Verklaar de door mij verstrekte code met behulp van commentaar regels in de code. Dit is een reflectie van de les die aan deze opdracht is gekoppeld. Gebruik hierbij je eigen woorden! (0-5 punten)

Als je les 5 hebt gemist, dan verwacht ik toch dat je aan de hand van de lesbrief over arrays de werking van deze workshop opdracht probeert uit te leggen.

Je scoort een voldoende als je minimaal 5 punten hebt gehaald. (2 voor de werking en minimaal 3 voor de uitleg). Heb je een super goede uitleg, dan kun je  $2 + 5 = 7$  punten halen. (gáán voor die A!!)

## Stappen plan debuggen: In geval van “Het werkt niet!”

### Stap 1: open Javascript console Firefox erbij, of debug window.

<b>Vraag:</b> Roep je wel de goede pagina op?
<b>Actie:</b> Open de pagina vanuit het programma waar je mee aan het editen bent bijv. F12 in Dreamweaver.
<b>Actie:</b> Refresh browser window

### Stap 2: Eerste blik op de code:

<b>Vraag:</b> Klopt de color syntax (moet je wel dreamweaver of notepad ++ hebben).
<b>Actie:</b> Zie je iets raars pas het aan!

### Stap 3: Goed (hardop) lezen foutmelding

<b>Vraag:</b> Ga naar de plaats van de fout melding en lees het script goed (hardop)
<b>Actie:</b> Let bij het lezen op: <ol style="list-style-type: none"><li>1. Hoofdletters en kleine letters goed gebruikt?</li><li>2. Leestekens controleren ( ; ' " )</li><li>3. Naamgeving consequent?</li><li>4. Voor gebruik altijd inspringen /tabs binnen code<ol style="list-style-type: none"><li>a. Wordt elke haak die je opent ook afgesloten wordt.</li><li>b. Zo ook voor accolades</li><li>c. Zo ook voor commentaar</li></ol></li></ol>

### Stap 4: Goed doorlezen van het gehele script...

<b>Actie:</b> Doe stap 3 voor het gehele script.
<b>Vraag:</b> Wil je weten wat er gebeurd door het script heen:
<b>Actie:</b> In bouwen van alerts om tussentijds waardes/uitkomsten weer te geven op de plek waar de fout mogelijk zou kunnen staan. Weet je niet precies waar dat is, begin dan bovenaan je script.

### Stap 5: Functies Controleren

<b>Vraag:</b> Wordt de functie aangeroepen?
<b>Actie:</b> Bij gebruik van functions, if of for (dus er moet iets worden uitgevoerd al dan niet conditioneel) plaats een alert als eerste statement. Als deze wordt uitgevoerd dan ben je al een heel eind
<b>Vraag:</b> Is de volgorde van de code goed?
<b>Actie:</b> Als je functies gebruikt: worden deze gedeclareerd VOORDAT je deze aanroept?
<b>Vraag:</b> Vraag:Bestaan de elementen of objecten die je aanroept wel?
<b>Actie:</b> Maw. heb je deze een NAAM of een ID gegeven en klopt dit met wat je roept in javascript?

### Stap 6: Tests / Loops Controleren

<b>Vraag:</b> Heb je wel een dubbele == gebruikt binnen je if?
<b>Vraag:</b> Bestaan de elementen of objecten die je aanroept wel?
<b>Actie:</b> Maw. heb je deze een NAAM of een ID gegeven en klopt dit met wat je roept in javascript?
<b>Vraag:</b> Eeuwige loop? Maw. zijn de tests wel goed binnen de loop?
<b>Actie:</b> Misschien groter dan > ipv. kleiner dan < of andersom gebuiken?

### **Stap7: Geheel debuggen met alerts of trace (Flash)**

**Actie:** Inbouwen van alerts om tussentijds waardes/uitkomsten weer te geven op de plek waar de fout mogelijk zou kunnen staan.  
Weet je niet precies waar dat is, begin dan bovenaan je script.

### **Stap 8: De laatste vraag**

**Vraag:** Was deze oplossing voor u bevredigend? Heeft de oplossing geholpen?  
**Actie:** ZO NEE! roep docent en geef stapnummer aan.