

Tentamen Programmeren in JavaScript

Vrijdag 11 januari 2002, 9.30–12.30 uur

Universiteit Leiden — Informatica

De vijf opgaven tellen even zwaar mee. Geef overal duidelijke uitleg. Veel succes!

- Leg uit hoe je in JavaScript een plaatje op een bepaalde plek kan zetten. Waarom bestond deze mogelijkheid niet in eerdere browsers?
 - Schrijf een JavaScript-functie `menu ()` die het volgende doet. Aan de gebruiker wordt een klein menu met twee keuzes gepresenteerd, S en Y. Bij S stopt de functie, en bij Y verschijnt een alert-box met de tekst YyyyY. Bij de keuze Y wordt daarna opnieuw het menu gepresenteerd.
 - Het volgende stukje JavaScript gebruikt een timer:

```
function update ( ) {  
    ... doe iets ...  
    naamtimer = setTimeout ("update ( );", 2000);  
} // update  
  
update ( );  
<... onClick="clearTimeout (naamtimer); update ( );">
```

Leg uit wat er gebeurt, en geef aan waarom de timer “gecleard” wordt.

- We hebben een array A met `A.length` positieve gehele getallen. Dit zijn de lengtes van een aantal personen.
 - Geef een JavaScript-functie `buren (A)` die een array-element uit het array A retourneert, waarvoor geldt dat het gelijk is aan zijn directe buur. (Staan er twee even lange mensen direct naast elkaar — en hoe lang zijn ze?) Als zo’n array-element niet te vinden is, moet `-1` worden teruggegeven. Mochten er meerdere getallen gelijk zijn aan hun buurman, dan mag het eerste het beste worden geretourneerd.
Voorbeeld: het array A (ter lengte 8) bevat 1, 7, 0, 2, 9, 9, 0, 6; de functie geeft dan 9 terug, want op plek 4 en op plek 5 staat 9.
 - Idem, maar nu een array-element van het array A retourneren, waarvoor geldt dat het gelijk is aan een ander array-element. (Zijn er twee even lange mensen — en hoe lang zijn ze?)
Voorbeeld: het array A (ter lengte 8) bevat 1, 7, 0, 2, 9, 8, 0, 6; de functie geeft dan 0 terug, want op plek 2 en op plek 6 staat 0.
 - Geef een JavaScript-functie `sorteer (A)` die A *aflopend* sorteert. Pas bijvoorbeeld bubblesort aan.

- Bij een functie kun je te maken hebben met *locale* en *globale* variabelen. Verder heb je ook nog variabelen (parameters) die tussen haakjes worden meegegeven. Leg deze begrippen aan de hand van een *klein* voorbeeld duidelijk uit.
 - Een zeker JavaScript-programma bevat de volgende functies:

```
function aap ( ) { var temp = 3*u; return temp; }  
function noot (u,v) { u = 3*u; v = 2*v; return u + v; }  
function mies ( ) { u = 3*u; v = 2*v; return u + v; }
```

Wat gebeurt er bij het volgende stukje JavaScript? NB *u*, *v* en *w* zijn globale variabelen.

```
u = 7; v = 2; w = aap ( ); alert (u); alert (v); alert (w);
u = 7; v = 2; w = noot (u,v); alert (u); alert (v); alert (w);
u = 7; v = 2; w = noot (v,u); alert (u); alert (v); alert (w);
u = 7; v = 2; w = mies ( ); alert (u); alert (v); alert (w);
```

Wat wordt er in de alert-boxen afgedrukt? Geef hierbij uiteraard uitleg.

4. Deze opgave gaat over een *m* bij *n* array *A* met gehele getallen tussen 0 en 99. Een voorbeeld, met *m* = 3 rijen en *n* = 5 kolommen:

```
10 20 0 40 50
0 0 88 33 55
31 0 13 91 1
```

a. Schrijf een JavaScript-functie *drukaf* (*A*) die het array *A* precies als boven afdruckt, dus *n* getallen per regel, en de kolommen rechts uitgelijnd.

b. Schrijf een stukje JavaScript dat een *m* bij *n* array *A* maakt (denk aan *new*), en de array-elementen als volgt met getallen tussen 0 en 99 vult. In de *i*-de rij, *j*-de kolom komt de som van *i* en *j* te staan, en 0 als deze som groter dan 99 is.

c. Schrijf een JavaScript-functie *hoeveel* (*A*) die bepaalt hoeveel array-elementen er zijn waarvan *alle burens* ongelijk aan 0 zijn. Een buur van een array-element is een array-element dat er horizontaal, verticaal of diagonaal direct aan grenst. In het voorbeeld zijn het er 4, namelijk 1, 50, 55 en 91.

5. De functie *HaalUit* retourneert het eerste element van een array, verwijdert dit element uit het array en schuift alle volgende elementen eentje naar voren: als *A* het array (0,1,2,3) is, dan retourneert *HaalUit* (*A*) de waarde 0, en is de waarde van *A* na de aanroep (1,2,3).

a. De functie *StopIn* doet precies het tegenovergestelde: het schuift alle elementen van het array eentje naar achteren en stopt een nieuw element vooraan in het array: stel dat *A* het array (1,2,3) is; na aanroep *StopIn* (0, *A*) is de waarde van *A* het volgende: (0,1,2,3). Schrijf deze functie in JavaScript.

b. Functie *RaRa* krijgt als argument een array. Leg uit wat de functie afdruckt (dat mag aan de hand van een voorbeeld: neem *A* is (0,1,2,3)).

```
function RaRa (A) {
  var B = new Array ( );
  var x;
  var i;
  while ( A.length != 0 ) {
    x = HaalUit (A);
    StopIn (x, B);
  } // while
  for ( i = 0; i < B.length; i++ )
    window.document.write (B[i] + " ");
} // RaRa
```