



samen sterk voor werk

Vlaamse Dienst voor Arbeidsbemiddeling en  
Beroepsopleiding

*Knockout.*

takenbundel

Deze cursus is eigendom van de VDAB©

## Inhoudsopgave

<b>1</b>	<b>WELKOMBOODSCHAP</b>	<b>1</b>
<b>2</b>	<b>BMI</b>	<b>2</b>
<b>3</b>	<b>GALGJE</b>	<b>3</b>
<b>4</b>	<b>PALINDROOM</b>	<b>4</b>
<b>5</b>	<b>STERRENBEELDEN</b>	<b>5</b>
<b>6</b>	<b>CONVERSIE</b>	<b>6</b>
<b>7</b>	<b>FRUIT</b>	<b>7</b>
<b>8</b>	<b>VERKOPEN</b>	<b>8</b>
<b>9</b>	<b>FRITUUR</b>	<b>9</b>
<b>10</b>	<b>VERJAARDAGEN</b>	<b>10</b>
<b>11</b>	<b>PRODUCTEN</b>	<b>11</b>
<b>12</b>	<b>PERSONEEL</b>	<b>12</b>
<b>13</b>	<b>PANNENKOEKEN</b>	<b>13</b>
<b>14</b>	<b>PERSONEEL 2</b>	<b>14</b>
<b>15</b>	<b>WELKOMBOODSCHAP: VOORBEELDOPLOSSING</b>	<b>15</b>
15.1	welkomboodschap.html	15
15.2	welkomboodschap.js	15
<b>16</b>	<b>BMI: VOORBEELDOPLOSSING</b>	<b>16</b>
16.1	bmi.html	16
16.2	bmi.js	16
<b>17</b>	<b>GALGJE: VOORBEELDOPLOSSING</b>	<b>17</b>
17.1	galgje.html	17
17.2	galgje.js	17

<b>18</b>	<b>PALINDROOM: VOORBEELDOPLOSSING</b>	<b>19</b>
18.1	palindroom.html	19
18.2	palindroom.js	19
<b>19</b>	<b>STERRENBEELDEN: VOORBEELDOPLOSSING</b>	<b>20</b>
19.1	sterrenbeelden.html	20
19.2	sterrenbeelden.js	20
<b>20</b>	<b>CONVERSIE: VOORBEELDOPLOSSING</b>	<b>22</b>
20.1	conversie.html	22
20.2	conversie.js	22
<b>21</b>	<b>FRUIT: VOORBEELDOPLOSSING</b>	<b>23</b>
21.1	fruit.html	23
21.2	fruit.js	23
<b>22</b>	<b>VERKOPEN VOORBEELDOPLOSSING</b>	<b>24</b>
22.1	verkopen.html	24
22.2	verkopen.js	24
<b>23</b>	<b>FRITUUR: VOORBEELDOPLOSSING</b>	<b>26</b>
23.1	frituur.html	26
23.2	frituur.js	26
<b>24</b>	<b>VERJAARDAGEN: VOORBEELDOPLOSSING</b>	<b>28</b>
24.1	verjaardagen.html	28
24.2	verjaardagen.js	28
<b>25</b>	<b>PRODUCTEN: VOORBEELDOPLOSSING</b>	<b>30</b>
<b>26</b>	<b>PERSONEEL: VOORBEELDOPLOSSING</b>	<b>31</b>
26.1	personeel.html	31
26.2	personeel.js	31
<b>27</b>	<b>PANNENKOEKEN: VOORBEELDOPLOSSING</b>	<b>32</b>
27.1	pannenkoeken.html	32

---

27.2	pannenkoeken.js	32
<b>28</b>	<b>PERSONEEL 2: VOORBEELDOPLOSSING</b>	<b>33</b>
28.1	personeel.html	33
28.2	personeel.js	33
<b>29</b>	<b>COLOFON</b>	<b>35</b>

## 1 WELKOMBOODSCHAP

Afhankelijk van het uur van de dag toon je de gebruiker een andere boodschap en een andere afbeelding

Uur van de dag	Boodschap	Afbeelding
6, 7, 8, 9, 10	Goede morgen	morgen.jpg
11, 12, 13, 14, 15, 16, 17	Goede middag	middag.jpg
18, 19, 20, 21, 22, 23	Goede avond	avond.jpg
0, 1, 2, 3, 4, 5	Goede nacht	nacht.jpg

Voorbeeld van de pagina als het uur 16 is



Je vindt de nodige afbeeldingen in het materiaal bij de cursus.

## 2 BMI

Je maakt een applicatie om aan de hand van de BMI (Body Mass Index) je “ideale” gewicht te berekenen.

Het ViewModel bevat je huidig gewicht (in kilo) en je huidige lengte (in meter).  
Je beeldt deze data ook af in de View.

Je toont in de View ook het BMI.

Dit is de uitkomst van de formule  $\text{gewicht} / (\text{lengte} * \text{lengte})$

Je toont in de View ook een boodschap, gebaseerd op je BMI.

BMI	Boodschap in de View
minder dan 18,5	ondergewicht
18,5 tot 25	normaal gewicht
25 tot 27	licht overgewicht
27 tot 30	matig overgewicht
30 tot 40	ernstig overgewicht
Meer dan 40	fel overgewicht

De View bevat twee knoppen

- Een knop 1 kg. lichter  
Als de gebruiker daarop klikt, vermindert hij zijn gewicht met één kg.  
Je herberekent het BMI en de boodschap wijzigt eventueel.
- Een knop 1 kg. zwaarder  
Als de gebruiker deze knop aanklikt, vermeerderd het gewicht met één kg.  
Je herberekent het BMI en de boodschap wijzigt eventueel.

Voorbeeld van de pagina

### Body mass index

Lengte  
**1.69**

Gewicht  
**77**

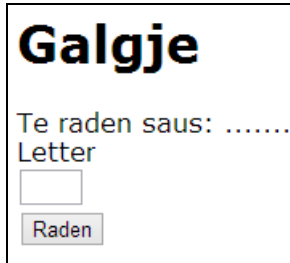
BMI  
**26.959840341724732: licht overgewicht**

1 kg. lichter   1 kg. zwaarder

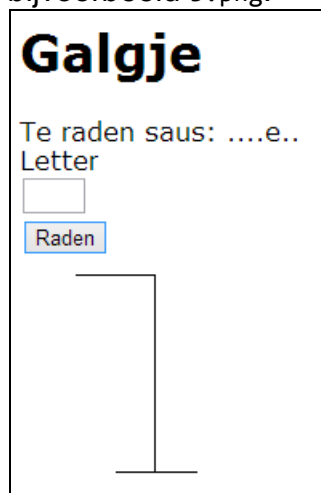
### 3 GALGJE

Je maakt een applicatie waarin de gebruiker de naam van een frietsaus raadt.

Als hij de pagina opent, ziet deze er zo uit:



- Je toont naast Te raden saus: evenveel puntjes als de saus die de gebruiker moet raden. Je kiest de saus willekeurig uit cocktail, mayonaise, mosterd, tartare, vinaigrette
- De gebruiker kan in de textbox een letter intikken.  
De gebruiker ziet daarna het volgende.
  - Als de ingetikte letter voorkomt in de te raden saus, toon je op de plaats van die letter geen puntje, maar de letter. Als de gebruiker bijvoorbeeld de letter *t* tikte en de te raden saus is *cocktail*, toon je Te raden saus: ....t...
  - Als de ingetikte letter niet voorkomt in de te raden saus, verhoog je een teller met het aantal verkeerde pogingen. Je toont deze teller aan de gebruiker via één van de afbeeldingen 0.png ... 9.png bij deze cursus. Als de gebruiker drie verkeerde pogingen deed, toon je bijvoorbeeld 3.png:



- Als de gebruiker 10 verkeerde pogingen deed, verliest hij het spel en toon je volgende tekst: U bent verloren, de saus was cocktail
- Als de gebruiker de saus helemaal raadt en nog geen 10 verkeerde pogingen deed, wint hij het spel en toon je volgende tekst:  
U bent gewonnen, de saus was cocktail

Je vindt de nodige afbeeldingen in het materiaal bij de cursus.

## 4 PALINDROOM

Je maakt een applicatie waarin de gebruiker een woord intikt.

Je toont aan de gebruiker of dit woord een palindroom is.

Standaard is de controle niet hoofdlettergevoelig (1epeL is dan een palindroom).

De gebruiker kan een checkbox aanvinken. Dan is de controle wel hoofdlettergevoelig (1epeL is dan geen palindroom)

### Palindroom

Woord:

  
☐ Hoofdlettergevoelig  
**Dit is een palindroom**



## 5 STERRENBEELDEN

De gebruiker kan in een combobox een dag in de maand kiezen (tussen 1 en 31).  
De gebruiker kan in een tweede combobox een maand kiezen (tussen 1 en 12).

Je toont daarna het sterrenbeeld van de gebruiker en de bijbehorende afbeelding

**Sterrenbeeld**

Dag:

Maand:

**leeuw**



De sterrenbeelden zijn als volgt

Sterrenbeeld	Periode
steenbok	22 december-20 januari
kreeft	22 juni -22 juli
waterman	21 januari - 19 februari
leeuw	23 juli - 23 augustus
vissen	20 februari - 20 maart
maagd	24 augustus - 23 september
ram	21 maart - 20 april
weegschaal	24 september - 23 oktober
stier	21 april - 20 mei
schorpion	24 oktober - 22 november
tweeling	21 mei - 21 juni
boogschutter	23 november - 21 december

Je vindt de nodige afbeeldingen in het materiaal bij de cursus.

## 6 CONVERSIE

De gebruiker kan in een textbox een afstand intikken.

Hij kan met een eerste radiobutton kiezen deze afstand van centimeters naar inches om te zetten.

Hij kan met een tweede radiobutton kiezen deze afstand van inches naar centimeters om te zetten.

### Conversie

Afstand:

☒ Centimeters naar inches  
☐ Inches naar centimeters

**Geconverteerde afstand: 2**

## 7 FRUIT

Je toont een lijst met fruit



Als de gebruiker een fruit aanklikt, toon je onder de lijst de afbeelding van het geselecteerde fruit. Als de gebruiker de muisaanwijzer laat rusten boven de afbeelding, toon je het aantal kilo calorieën van het geselecteerde fruit. Je gebruikt daartoe het `title` attribuut van het HTML element `img`.



Het fruit heeft volgende kilo calorieën

Fruit	Kilo calorieën
Aalbessen	38
Aardbeien	36
Abrikozen	60
Ananas	58

Je vindt de nodige afbeeldingen in het materiaal bij de cursus.

## 8 VERKOPEN

Je maakt een applicatie waarmee de gebruiker per seizoen het bedrag verkoop kan intikken

Verkopen		
Seizoen	Verkoop	Percentage
Lente	1000	12%
Zomer	2000	24%
Herfst	2500	29%
Winter	3000	35%
Totaal	8500	100%

Je toont onder de seizoenen het totale bedrag verkoop.

Je toont naast ieder seizoen hoeveel percentage het bedrag van dat seizoen is ten opzichte van het totale bedrag verkoop.

Je rekt de afbeelding grafiek.png uit om de grafiekbalkjes te tonen.

## 9 FRITUUR

Je maakt een applicatie waarmee je een bestelling voor een frituur kan ingeven.  
Initieel is de bestelling leeg

**Frituur**

Product	Prijs	Aantal
Totaal: € 0		

De gebruiker voegt met de knop Toevoegen een bestellijn toe aan de bestelling

**Frituur**

Product	Prijs	Aantal
Kies een product ▼		1
Totaal: € 0		

Hij kan bij Kies een product een product kiezen.  
Ieder van deze producten heeft ook een prijs, die je zelf verzint.

**Frituur**

Product	Prijs	Aantal
Kies een product ▼		1
Totaal: € 0		

Kies een product  
Kleine friet  
Medium friet  
Grote friet  
Mayonaise  
Stoofvlees  
Kippenboutjes

Nadat hij een product kiest, ziet hij de prijs van het product  
en kan hij het aantal wijzigen

**Frituur**

Product	Prijs	Aantal
Kleine friet ▼	2.5	2
Totaal: € 5		

De gebruiker ziet onder in de tabel de totale waarde van de bestelling.

## 10 VERJAARDAGEN

Je maakt een applicatie waarmee je de verjaardagen van personen bijhoudt.

### Verjaardagen

Naam	Dag	Maand	Vandaag jarig
<input type="text" value="Joe"/>	<input type="text" value="1"/> ▼	<input type="text" value="8"/> ▼	nee
<input type="text" value="Jack"/>	<input type="text" value="16"/> ▼	<input type="text" value="8"/> ▼	nee
<input type="text" value="William"/>	<input type="text" value="14"/> ▼	<input type="text" value="6"/> ▼	nee
<input type="text" value="Averell"/>	<input type="text" value="8"/> ▼	<input type="text" value="10"/> ▼	ja
<input type="text" value="Luke"/>	<input type="text" value="11"/> ▼	<input type="text" value="10"/> ▼	nee
<input type="text" value="Bill"/>	<input type="text" value="7"/> ▼	<input type="text" value="10"/> ▼	nee
<input type="text" value="Jane"/>	<input type="text" value="8"/> ▼	<input type="text" value="10"/> ▼	ja

De gebruiker voegt met de knop **Toevoegen** een nieuwe rij toe aan de tabel.

Hij kan de naam intikken, de dag kiezen uit een lijst 1..31 en de maand kiezen uit een lijst 1..12.

Als een persoon jarig is op de systeemdatum, toon je **ja** in de laatste kolom.

Als de gebruiker de pagina herlaadt of verlaat, sla je de data op in local storage.

## 11 PRODUCTEN

Je breidt de taak Frituur uit.

Tot nu waren de producten hard gecodeerd in de JavaScript van deze taak.  
Je leest nu deze producten uit het bestand `producten.json`.

## 12 PERSONEEL

Je maakt een applicatie waarmee je de data uit `personeel1.json` toont.

Je toont initieel de lijst van het personeel

### Personeel

- [Diane Murphy](#)
- [Mary Patterson](#)
- [Jeff Firrelli](#)
- [William Patterson](#)
- [Gerard Bondur](#)
- [Anthony Bow](#)
- [Leslie Jennings](#)
- [Leslie Thompson](#)
- [Julie Firrelli](#)
- [Steve Patterson](#)
- [Foon Yue Tseng](#)
- [George Vanauf](#)
- [Loui Bondur](#)
- [Gerard Hernandez](#)
- [Pamela Castillo](#)
- [Larry Bott](#)
- [Barry Jones](#)
- [Andy Fixter](#)
- [Peter Marsh](#)
- [Tom King](#)
- [Mami Nishi](#)
- [Yoshimi Kato](#)
- [Martin Gerard](#)

Als de gebruiker een personeelslid aanklikt,  
vervang je de lijst door een detailoverzicht van dat personeelslid

### Personeel

#### Diane Murphy

Nummer  
**1**

Wedde  
**3000**

Job  
**President**

[Terug naar lijst](#)

Als de gebruiker [Terug naar lijst](#) aanklikt,  
vervang je dit detailoverzicht door de lijst.



## 13 PANNENKOEKEN

Je leest in een applicatie de ingrediënten van pannenkoeken en de bereiding van pannenkoeken uit het bestand pannenkoeken.json.

Je toont initieel de ingrediënten

### Pannenkoeken

[Ingrediënten](#) [Bereiding](#)

#### Ingrediënten

- 200 gram bloem
- 4 eieren
- 0,5 liter melk
- 2 eetlepel arachideolie
- 20 gram boter
- 70 gram suiker

Als de gebruiker [Bereiding](#) aanklikt, toon je de bereiding

### Pannenkoeken

[Ingrediënten](#) [Bereiding](#)

#### Bereiding

- Smelt de boter in een pot en doe de bloem, eieren, suiker, melk en de olie erbij.
- Goed roeren tot alle klonters uit het deeg zijn.
- Doe een beetje arachide-olie in de pan en laat goed warm worden.
- Doe een schep van het deeg in de pan en begin onmiddellijk te draaien met de pan, zodanig dat het deeg mooi verdeeld is
- Gebruik een spatel voor het draaien en snij de pannenkoek los langs de randen.

Als de gebruiker [Ingrediënten](#) aanklikt, toon je terug de ingrediënten.

## 14 PERSONEEL 2

Dit is een uitbreiding op de taak Personeel.

Als de gebruiker een hyperlink in de lijst aanklikt, verwijst deze naar een URL fragment dat begint met #personeel1/, gevolgd door het personeelsnummer.

## 15 WELKOMBOODSCHAP: VOORBEELDOPLOSSING

### 15.1 welkomboodschap.html

```
<!doctype html>
<html lang="nl">
<head>
<meta charset="UTF-8">
<title>Welkomboodschap</title>
<link rel="stylesheet" href="default.css">
</head>
<body>
  <h1>
    Goede <span data-bind="text:boodschap()"></span>
  </h1>
  <img data-bind="attr:{src: 'images/'+boodschap()+' .jpg', alt:boodschap()}">
  <script
    src="http://cdnjs.cloudflare.com/ajax/libs/knockout/3.0.0/knockout-min.js">
</script>
  <script src="welkomboodschap.js"></script>
</body>
</html>
```

### 15.2 welkomboodschap.js

```
var viewModel = {
  boodschap : function() {
    switch (new Date().getHours()) {
      case 6:
      case 7:
      case 8:
      case 9:
      case 10:
        return "morgen";
      case 11:
      case 12:
      case 13:
      case 14:
      case 15:
      case 16:
      case 17:
        return "middag";
      case 18:
      case 19:
      case 20:
      case 21:
      case 22:
      case 23:
        return "avond";
      default:
        return "nacht";
    }
  }
};
ko.applyBindings(viewModel);
```

## 16 BMI: VOORBEELDOPLOSSING

### 16.1 bmi.html

```
<!doctype html>
<html lang="nl">
<head>
<meta charset="UTF-8"><title>BMI</title>
<link rel="stylesheet" href="default.css">
</head>
<body>
  <h1>Body mass index</h1>
  <dl>
    <dt>Lengte</dt>
    <dd data-bind="text:Lengte"></dd>
    <dt>Gewicht</dt>
    <dd data-bind="text:gewicht()"></dd>
    <dt>BMI</dt>
    <dd data-bind="text:bmi()+': '+boodschap()"></dd>
  </dl>
  <button data-bind="click:eenKiloLichter">1 kg. lichter</button>
  <button data-bind="click:eenKiloZwaarder">1 kg. zwaarder</button>
  <script
src="http://cdnjs.cloudflare.com/ajax/libs/knockout/3.0.0/knockout-min.js">
  </script>
  <script src="bmi.js"></script>
</body>
</html>
```

### 16.2 bmi.js

```
var viewModel = {
  lengte : 1.69,
  gewicht : ko.observable(77),
  bmi : function() {
    return this.gewicht() / (this.lengte * this.lengte);
  },
  boodschap : function() {
    var bmi = this.bmi();
    if (bmi < 18.5)
      return "ondergewicht";
    else if (bmi <= 25)
      return "normaal gewicht";
    else if (bmi <= 27)
      return "licht overgewicht";
    else if (bmi <= 30)
      return "matig overgewicht";
    else if (bmi <= 40)
      return "ernstig overgewicht";
    else
      return "fel overgewicht";
  },
  eenKiloLichter : function() {
    this.gewicht(this.gewicht() - 1);
  },
  eenKiloZwaarder : function() {
    this.gewicht(this.gewicht() + 1);
  }
};
ko.applyBindings(viewModel);
```

## 17 GALGJE: VOORBEELDOPLOSSING

### 17.1 galgje.html

```
<!doctype html>
<html lang="nl">
<head>
<meta charset="UTF-8">
<title>Galgje</title>
<link rel="stylesheet" href="default.css">
</head>
<body>
  <h1>Galgje</h1>
  <div>Te raden saus:
    <span data-bind="text:puntjes()"></span></div>
  <label>Letter
  <input data-bind="value:letter" maxlength="1" size="1" autofocus id="letter">
  <button data-bind="click:raden">Raden</button>
</label>
  <div data-bind="text:boodschap()"></div>
  <div><img data-bind="attr:{src:'images/'+aantalVerkeerdePogingen()+'.png',
    alt:aantalVerkeerdePogingen()+ ' verkeerde pogingen'}"></div>
  <script src="http://cdnjs.cloudflare.com/ajax/libs/knockout/3.0.0/knockout-
min.js"></script>
  <script src="galgje.js"></script>
</body>
</html>
```

### 17.2 galgje.js

```
function maakTeRadenSaus() {
  var sausen = [ "cocktail", "mayonaise", "mosterd", "tartare", "vinaigrette" ];
  var randomIndex = Math.floor(Math.random() * sausen.length);
  return sausen[randomIndex];
}

function maakPuntjes(saus, geradenLetters) {
  var puntjes = "";
  var index;
  var letter;
  for (index = 0; index !== saus.length; index++) {
    letter = saus.charAt(index);
    if (geradenLetters.indexOf(letter) === -1) {
      puntjes += ".";
    } else {
      puntjes += letter;
    }
  }
  return puntjes;
}

var teRadenSaus = maakTeRadenSaus();
var viewModel = {
  geradenLetters : "",
  letter : ko.observable(),
  puntjes : ko.observable(maakPuntjes(teRadenSaus, "")),
  aantalVerkeerdePogingen : ko.observable(0),
  boodschap : ko.observable(""),
}
```

```
raden : function() {
    if (teRadenSaus.indexOf(this.letter()) === -1) {
        this.aantalVerkeerdePogingen(this.aantalVerkeerdePogingen() + 1);
    }
    this.geradenLetters += this.letter();
    this.puntjes(maakPuntjes(teRadenSaus, this.geradenLetters));
    this.letter("");
    document.getElementById("letter").focus();
    if (this.aantalVerkeerdePogingen() === 10) {
        this.boodschap("U bent verloren, de saus was " + teRadenSaus);
    } else if (teRadenSaus === this.puntjes()) {
        this.boodschap("U hebt gewonnen, de saus was " + teRadenSaus);
    }
}
};
ko.applyBindings(viewModel);
```

## 18 PALINDROOM: VOORBEELDOPLOSSING

### 18.1 palindroom.html

```
<!doctype html>
<html lang="nl">
<head>
<meta charset="UTF-8">
<title>Palindroom</title>
<link rel="stylesheet" href="default.css">
</head>
<body>
  <h1>Palindroom</h1>
  <label>Woord: <input
    data-bind="value:woord, valueUpdate:'keyup'" autofocus></label>
  <label> <input type="checkbox"
    data-bind="checked:hoofdLetterGevoelig"> Hoofdlettergevoelig
  </label>
  <h2 data-bind="visible:woord().length !== 0" style="display: none"> (1)
    Dit is <span data-bind="text:isPalindroom() ? 'een' : 'geen'"></span>
    palindroom
  </h2>
  <script
    src="http://cdnjs.cloudflare.com/ajax/libs/knockout/3.0.0/knockout-
min.js"></script>
  <script src="palindroom.js"></script>
</body>
</html>
```

- (1) Met `style="display: none"` verhinder je dat bij het laden van de pagina deze h2 een fractie van een seconde getoond wordt, vooraleer Knockout zijn databinding kan doen.

### 18.2 palindroom.js

```
var viewModel = {
  woord : ko.observable(""),
  hoofdLetterGevoelig : ko.observable(false),
  isPalindroom : function() {
    if (this.hoofdLetterGevoelig()) {
      return this.woord().split("").reverse().join("") === this.woord();
    } else {
      return this.woord().toUpperCase().split("").reverse().join("") ===
        this.woord().toUpperCase();
    }
  }
};
ko.applyBindings(viewModel);
```

## 19 STERRENBEELDEN: VOORBEELDOPLOSSING

### 19.1 sterrenbeelden.html

```
<!doctype html>
<html lang="nl">
<head>
<meta charset="UTF-8">
<title>Sterrenbeelden</title>
<link rel="stylesheet" href="default.css">
</head>
<body>
  <h1>Sterrenbeeld</h1>
  <label>Dag: <select data-bind="options:dagen, value:dag,
    optionsCaption:'Maak een keuze'"></select>
  </label>
  <label>Maand: <select data-bind="options:maanden, value:maand,
    optionsCaption:'Maak een keuze'"></select>
  </label>
  <h2 data-bind="text:sterrenbeeld()"></h2>
  <img
    data-bind="attr:{src: 'images/'+sterrenbeeld()+' .png', alt:sterrenbeeld()},
    visible:dag() !== undefined && maand() !== undefined"
    style="display: none">
  <script
    src="http://cdnjs.cloudflare.com/ajax/libs/knockout/3.0.0/knockout-
min.js"></script>
  <script src="sterrenbeelden.js"></script>
</body>
</html>
```

### 19.2 sterrenbeelden.js

```
function vulArray(van, tot) {
  var array = [];
  var index;
  for (index = van; index <= tot; index++) {
    array.push(index);
  }
  return array;
}

var viewModel = {
  dagen : vulArray(1, 31),
  maanden : vulArray(1, 12),
  dag : ko.observable(),
  maand : ko.observable(),
  sterrenbeeld : function() {
    if (this.dag() === undefined || this.maand() === undefined) {
      return "";
    }
    switch (this.maand()) {
      case 1:
        return this.dag() >= 21 ? "waterman" : "steenbok";
      case 2:
        return this.dag() >= 20 ? "vissen" : "waterman";
      case 3:
        return this.dag() >= 21 ? "ram" : "vissen";
      case 4:
        return this.dag() >= 21 ? "stier" : "ram";
    }
  }
}
```



```
    case 5:
        return this.dag() >= 21 ? "tweeling" : "stier";
    case 6:
        return this.dag() >= 22 ? "kreeft" : "tweeling";
    case 7:
        return this.dag() >= 24 ? "leeuw" : "kreeft";
    case 8:
        return this.dag() >= 24 ? "maagd" : "leeuw";
    case 9:
        return this.dag() >= 24 ? "weegschaal" : "maagd";
    case 10:
        return this.dag() >= 24 ? "schorpioen" : "weegschaal";
    case 11:
        return this.dag() >= 23 ? "boogschutter" : "schorpioen";
    default:
        return this.dag() >= 23 ? "steenbok" : "boogschutter";
    }
}
};
ko.applyBindings(viewModel);
```

## 20 CONVERSIE: VOORBEELDOPLOSSING

### 20.1 conversie.html

```
<!doctype html>
<html lang="nl">
<head>
<meta charset="UTF-8">
<title>Conversie</title>
<link rel="stylesheet" href="default.css">
</head>
<body>
  <h1>Conversie</h1>
  <label>Afstand:
  <input type="number" data-bind="value:afstand, valueUpdate:'keyup'"
    autofocus></label>
  <div><label>
  <input type="radio" name="conversie" value="cmInch"
    data-bind="checked:conversie">Centimeters naar inches</label></div>
  <div><label>
  <input type="radio" name="conversie" value="inchCM"
    data-bind="checked:conversie">Inches naar centimeters</label></div>
  <h2>Geconverteerde afstand:
  <span data-bind="text:geconverteerd()"></span></h2>
  <script src="http://cdnjs.cloudflare.com/ajax/libs/knockout/3.0.0/knockout-
min.js"></script>
  <script src="conversie.js"></script>
</body>
</html>
```

### 20.2 conversie.js

```
var viewModel = {
  afstand : ko.observable(),
  conversie : ko.observable("cmInch"),
  geconverteerd : function() {
    if (this.afstand() === undefined || this.afstand() === ""
      || isNaN(this.afstand())) {
      return "";
    }
    if (this.conversie() === "cmInch") {
      return this.afstand() / 2.54;
    }
    return this.afstand() * 2.54;
  }
};
ko.applyBindings(viewModel);
```

## 21 FRUIT: VOORBEELDOPLOSSING

### 21.1 fruit.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Fruit</title>
<link rel="stylesheet" href="default.css">
<style>
li {
  cursor:pointer;
}
</style>
</head>
<body>
<h1>Fruit</h1>
  <ul data-bind="foreach:fruit">
    <li data-bind="text:naam, click:$root.detail"></li>
  </ul>
  <div data-bind="with:gekozenFruit(), visible:gekozenFruit() !== undefined"
  style="display:none">
    <img data-bind="attr:{src:'images/'+naam+'.jpg',
                        alt:naam,
                        title:kcal+' kilo calorieën'}">

  </div>
  <script
    src="http://cdnjs.cloudflare.com/ajax/libs/knockout/3.0.0/knockout-
min.js"></script>
  <script src="fruit.js"></script>
</body>
</html>
```

### 21.2 fruit.js

```
var viewModel = {
  fruit : [ {naam : "Aalbessen", kcal : 38},
             {naam : "Aardbeien",kcal : 36},
             {naam : "Abrikozen", kcal :60},
             {naam : "Ananas", kcal : 58} ],
  detail:function(fruit) {
    viewModel.gekozenFruit(fruit);
  },
  gekozenFruit:ko.observable()
};
ko.applyBindings(viewModel);
```

## 22 VERKOPEN VOORBEELDOPLOSSING

### 22.1 verkopen.html

```
<!doctype html>
<html>
<head>
<meta charset="UTF-8">
<title>Verkopen</title>
<link rel="stylesheet" href="default.css">
</head>
<body>
<h1>Verkopen</h1>
<table>
<thead>
<tr>
<th>Seizoen</th>
<th>Verkoop</th>
<th>Percentage</th>
</tr>
</thead>
<tbody data-bind="foreach:seizoenen">
<tr>
<td data-bind="text:naam"></td>
<td><input data-bind="value:verkoop" type="number"/></td>
<td>
<span data-bind="text:percentage()"></span>%</td>
</tr>
</tbody>
<tfoot>
<tr>
<td>Totaal</td>
<td data-bind="text:totaal()"></td>
<td>
100%</td>
</tr>
</tfoot>
</table>
<script
src="http://cdnjs.cloudflare.com/ajax/libs/knockout/3.0.0/knockout-
min.js"></script>
<script src="verkopen.js"></script>
</body>
</html>
```

### 22.2 verkopen.js

```
var viewModel = {
  seizoenen : [
    {naam : "Lente", verkoop : ko.observable(), percentage : ko.observable(0)},
    {naam : "Zomer", verkoop : ko.observable(), percentage : ko.observable(0)},
    {naam : "Herfst", verkoop : ko.observable(), percentage : ko.observable(0)},
    {naam : "Winter", verkoop : ko.observable(), percentage : ko.observable(0)}
  ],
  totaal : function() {
    var totaal = 0;
    var index;
    var verkoop;
```

```
    for (index = 0; index !== this.seizoenen.length; index++) {
        verkoop = this.seizoenen[index].verkoop();
        if (verkoop !== undefined && (!isNaN(verkoop)) && verkoop !== "") {
            totaal += parseInt(verkoop);
        }
    }
    for (index = 0; index !== this.seizoenen.length; index++) {
        verkoop = this.seizoenen[index].verkoop();
        if (verkoop !== undefined && (!isNaN(verkoop)) && verkoop !== ""
            && totaal !== 0) {
            this.seizoenen[index].percentage(Math.round(verkoop * 100
                / totaal));
        }
        else {
            this.seizoenen[index].percentage(0);
        }
    }
    return totaal;
}
};
ko.applyBindings(viewModel);
```

## 23 FRITUUR: VOORBEELDOPLOSSING

### 23.1 frituur.html

```
<!doctype html>
<html>
<head>
<meta charset="UTF-8">
<title>Frituur</title>
<link rel="stylesheet" href="default.css">
</head>
<body>
<h1>Frituur</h1>
<table>
<thead>
<tr>
<th>Product</th>
<th>Prijs</th>
<th>Aantal</th>
</tr>
</thead>
<tbody data-bind="foreach:bestelling(">
<tr>
<td><select data-bind="options:$root.producten, value:product,
optionsText:'naam', optionsCaption:'Kies een product'"></select></td>
<td data-bind="with:product"><span data-bind="text:prijs"></span></td>
<td><input data-bind="value:aantal" type="number"></td>
</tr>
</tbody>
<tfoot>
<tr>
<td colspan="3" data-bind="text:'Totaal: &euro; '+bestellingWaarde()'"></td>
</tr>
</tfoot>
</table>
<button data-bind="click:toevoegen">Toevoegen</button>
<script
src="http://cdnjs.cloudflare.com/ajax/libs/knockout/3.0.0/knockout-
min.js"></script>
<script src="frituur.js"></script>
</body>
</html>
```

### 23.2 frituur.js

```
var viewModel = {
  producten: [
    {naam:"Kleine friet", prijs:2.5},
    {naam:"Medium friet", prijs:2.8},
    {naam:"Grote friet", prijs:3.2},
    {naam:"Mayonaise", prijs:0.8},
    {naam:"Stoofvlees", prijs:2.8},
    {naam:"Kippenboutjes", prijs:2.5}
  ],
  bestelling : ko.observableArray(),
  toevoegen : function() {
    this.bestelling.push({product:ko.observable(), aantal:ko.observable(1)});
    document.querySelector("tbody tr:last-child td:first-child select").focus();
  },
}
```

```
bestellingWaarde:function() {  
    var waarde = 0;  
    var index;  
    for (index = 0; index !== this.bestelling().length; index++) {  
        var bestellijn = this.bestelling()[index];  
        if (bestellijn.product() !== undefined) {  
            waarde += bestellijn.product().prijs * bestellijn.aantal();  
            waarde = Math.round(waarde *100) / 100; // afrondingsproblemen vermijden  
        }  
    }  
    return waarde;  
}  
};  
ko.applyBindings(viewModel);
```

## 24 VERJAARDAGEN: VOORBEELDOPLOSSING

### 24.1 verjaardagen.html

```
<!doctype html>
<html>
<head>
<meta charset="UTF-8">
<title>Verjaardagen</title>
<link rel="stylesheet" href="default.css">
</head>
<body>
<h1>Verjaardagen</h1>
<table>
<thead>
<tr>
<th>Naam</th>
<th>Dag</th>
<th>Maand</th>
<th>Vandaag jarig</th>
</tr>
</thead>
<tbody data-bind="foreach:personen()">
<tr>
<td><input data-bind="value:naam"></td>
<td><select data-bind="value:dag, options:$root.dagen,
optionsCaption:'Kies een dag'"></select></td>
<td><select data-bind="value:maand, options:$root.maanden,
optionsCaption:'Kies een maand'"></select></td>
<td data-bind="text:jarig() ? 'ja' : 'nee'"></td>
</tr>
</tbody>
</table>
<button data-bind="click:toevoegen">Toevoegen</button>
<script
src="http://cdnjs.cloudflare.com/ajax/libs/knockout/3.0.0/knockout-
min.js"></script>
<script src="verjaardagen.js"></script>
</body>
</html>
```

### 24.2 verjaardagen.js

```
function maakPersoonMetObservables(naam, dag, maand) {
    var persoon = {
        naam : ko.observable(naam),
        dag : ko.observable(dag),
        maand : ko.observable(maand)
    };
    persoon.jarig = function() {
        var vandaag = new Date();
        return persoon.dag() === vandaag.getDate() &&
            persoon.maand() === (vandaag.getMonth() + 1);
    };
    return persoon;
}
function opslaan() {
    window.localStorage.setItem("personen", ko.toJSON(viewModel.personen));
}
```



```
function laden() {
    var index;
    var personen = window.localStorage.getItem("personen");
    var persoon;
    if (personen !== null) {
        personen = JSON.parse(personen);
        for (index = 0; index != personen.length; index++) {
            persoon = personen[index];
            viewModel.personen.push(maakPersoonMetObservables(persoon.naam,
                persoon.dag, persoon.maand));
        }
    }
}

function vulArray(van, tot) {
    var index;
    var array = [];
    for (index = van; index <= tot; index++) {
        array.push(index);
    }
    return array;
}

var viewModel = {
    personen : ko.observableArray(),
    dagen : vulArray(1, 31),
    maanden : vulArray(1, 12),
    toevoegen : function() {
        this.personen.push(maakPersoonMetObservables("", null, null));
        document.querySelector("tbody tr:last-child td:first-child input")
            .focus();
    },
};

laden();
ko.applyBindings(viewModel);
window.onbeforeunload = function() {
    opslaan();
};
```

## 25 PRODUCTEN: VOORBEELDOPLOSSING

```
var viewModel = {
  bestelling : ko.observableArray(),
  toevoegen : function() {
    this.bestelling.push({product:ko.observable(), aantal:ko.observable(1)});
    document.querySelector("tbody tr:last-child td:first-child select").focus();
  },
  bestellingWaarde : function() {
    var waarde = 0;
    var index;
    for (index = 0; index !== this.bestelling().length; index++) {
      var bestellijn = this.bestelling()[index];
      if (bestellijn.product() !== undefined) {
        waarde += bestellijn.product().prijs * bestellijn.aantal();
        waarde = Math.round(waarde * 100) / 100; // afrondingsproblemen vermijden
      }
    }
    return waarde;
  }
};

var xmlHttpRequest = new XMLHttpRequest();
xmlHttpRequest.open("GET", "producten.json", true);
xmlHttpRequest.onreadystatechange = function() {
  if (xmlHttpRequest.readyState === 4 && xmlHttpRequest.status === 200) {
    viewModel.producten = JSON.parse(xmlHttpRequest.responseText);
    ko.applyBindings(viewModel);
  }
};
xmlHttpRequest.send();
```

## 26 PERSONEEL: VOORBEELDOPLOSSING

### 26.1 personeel.html

```
<!doctype html>
<html lang="nl">
<head>
<meta charset="UTF-8">
<title>Personeel</title>
<link rel="stylesheet" href="default.css">
</head>
<body>
  <h1>Personeel</h1>
  <div data-bind="template:template()"></div>
  <script id="Lijst" type="text/html">
    <ul data-bind="foreach:personeel">
      <li><a href="#" data-bind="text:voornaam+' '+familienaam,
        click:$root.detail"></a></li>
    </ul>
  </script>
  <script id="detail" type="text/html">
    <h2 data-bind="with:personeelslid"><span data-bind="text:voornaam"></span>
    <span data-bind="text:familienaam"></span></h2>
    <dl data-bind="with:personeelslid">
      <dt>Nummer</dt>
      <dd data-bind="text:nummer"></dd>
      <dt>Wedde</dt>
      <dd data-bind="text:wedde"></dd>
      <dt>Job</dt>
      <dd data-bind="text:job"></dd>
    </dl>
    <a href="#" data-bind="click:lijst">Terug naar lijst</a>
  </script>
  <script src="http://cdnjs.cloudflare.com/ajax/libs/knockout/3.0.0/knockout-
min.js"></script>
  <script src="personeel.js"></script>
</body>
</html>
```

### 26.2 personeel.js

```
var viewModel = {template:ko.observable("lijst"), personeelslid:ko.observable(),
  detail:function(gekozenPersoneelslid) {
    viewModel.personeelslid(gekozenPersoneelslid);
    viewModel.template("detail");
  },
  lijst:function() {
    viewModel.template("lijst");
  }
};
var xmlHttpRequest = new XMLHttpRequest();
xmlHttpRequest.open("GET", "personeel.json", true);
xmlHttpRequest.onreadystatechange = function() {
  if (xmlHttpRequest.readyState === 4 && xmlHttpRequest.status === 200) {
    viewModel.personeel = JSON.parse(xmlHttpRequest.responseText);
    ko.applyBindings(viewModel);
  }
};
xmlHttpRequest.send();
```

## 27 PANNENKOEKEN: VOORBEELDOPLOSSING

### 27.1 pannenkoeken.html

```
<!doctype html>
<html lang="nl">
<head>
<meta charset="UTF-8">
<title>Pannenkoeken</title>
<link rel="stylesheet" href="default.css">
</head>
<body>
  <h1>Pannenkoeken</h1>
  <a href="#ingredienten">Ingrediënten</a>
  <a href="#bereiding">Bereiding</a>
  <div data-bind="template:template()"></div>
  <script id="ingredienten" type="text/html">
    <h2>Ingrediënten</h2>
    <ul data-bind="foreach:ingredienten">
      <li data-bind="text:$data"></li>
    </ul>
  </script>
  <script id="bereiding" type="text/html">
    <h2>Bereiding</h2>
    <ul data-bind="foreach:bereiding">
      <li data-bind="text:$data"></li>
    </ul>
  </script>
  <script src="http://cdnjs.cloudflare.com/ajax/libs/knockout/3.0.0/knockout-
min.js"></script>
  <script src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
  <script src="sammy-latest.min.js"></script>
  <script src="pannenkoeken.js"></script>
</body>
</html>
```

### 27.2 pannenkoeken.js

```
var viewModel = {template : ko.observable()};
Sammy(function() {
  this.get("#ingredienten", function() {
    viewModel.template("ingredienten");
  });
  this.get("#bereiding", function() {
    viewModel.template("bereiding");
  });
  this.get("", function() {
    viewModel.template("ingredienten");
  });
}).run();
var xmlHttpRequest = new XMLHttpRequest();
xmlHttpRequest.open("GET", "pannenkoeken.json", true);
xmlHttpRequest.onreadystatechange = function() {
  if (xmlHttpRequest.readyState === 4 && xmlHttpRequest.status === 200) {
    var responseObject = JSON.parse(xmlHttpRequest.responseText);
    viewModel.ingredienten = responseObject.ingredienten;
    viewModel.bereiding = responseObject.bereiding;
    ko.applyBindings(viewModel);
  }
};
xmlHttpRequest.send();
```

## 28 PERSONEEL 2: VOORBEELDOPLOSSING

### 28.1 personeel.html

De eerste <a> tag wijzigt

```
<a data-bind="text:voornaam+' '+familienaam, attr:{href:'#personeel/'+nummer}">
```

De laatste <a> tag wijzigt

```
<a href="#">Terug naar lijst</a>
```

Extra script verwijzingen

```
<script  
src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>  
<script src="sammy-latest.min.js"></script>
```

### 28.2 personeel.js

```
var viewModel = {template : ko.observable(), personeelslid : ko.observable()};  
Sammy(  
  function() {  
    this.get("#personeel/:nummer", function() {  
      var nummer = parseInt(this.params["nummer"]);  
      var xmlHttpRequest;  
      if (viewModel.personeel === undefined) {  
        xmlHttpRequest = new XMLHttpRequest();  
        xmlHttpRequest.open("GET", "personeel.json", true);  
        xmlHttpRequest.onreadystatechange = function() {  
          if (xmlHttpRequest.readyState === 4  
            && xmlHttpRequest.status === 200) {  
            viewModel.personeel = JSON.parse(xmlHttpRequest.responseText);  
            zoekPersoneelslid(nummer);  
            viewModel.template("detail");  
            ko.applyBindings(viewModel);  
          }  
        };  
        xmlHttpRequest.send();  
      } else {  
        zoekPersoneelslid(nummer);  
        viewModel.template("detail");  
      }  
    });  
    this.get("", function() {  
      if (viewModel.personeel === undefined) {  
        var xmlHttpRequest = new XMLHttpRequest();  
        xmlHttpRequest.open("GET", "personeel.json", true);  
        xmlHttpRequest.onreadystatechange = function() {  
          if (xmlHttpRequest.readyState === 4  
            && xmlHttpRequest.status === 200) {  
            viewModel.personeel = JSON.parse(xmlHttpRequest.responseText);  
            viewModel.template("lijst");  
            ko.applyBindings(viewModel);  
          }  
        };  
        xmlHttpRequest.send();  
      }  
      else {  
        viewModel.template("lijst");  
      }  
    });  
  }  
);
```

```
    }  
    });  
  }).run();  
function zoekPersoneelslid(nummer) {  
  var index;  
  var personeelslid;  
  for (index = 0; index !== viewModel.personeel.length; index++) {  
    personeelslid = viewModel.personeel[index];  
    if (personeelslid.nummer === nummer) {  
      viewModel.personeelslid(personeelslid);  
      break;  
    }  
  }  
}  
}
```

## 29 COLOFON

**Domeinexpertisemanager:** Rita Van Damme

**Moduleverantwoordelijke:** Hans Desmet

**Medewerkers:** Hans Desmet

**Versie:** 28/10/2013

**Nummer dotatielijst:**