



# Oefenopgaven Objecten

## Opgave 1

- Maak een constructor-functie om persoon-objecten te maken, met verschillende attributen en methoden. Geef de attributen en methoden Nederlandse namen: voornaam, achternaam, geboortejaar. Voeg de methoden `wijzigVoornaam`, `wijzigAchternaam` toe en een functie die alle gegevens van een persoon als string aflevert.
- Maak twee objecten, een voor uzelf en een voor een familielid.
- Maak een webpagina waarin de objecten en hun methoden getest worden.
- Zet het script in een apart `.js` bestand. Test de code in verschillende browsers.

## Opgave 2

Maak een webpagina waarin het hiervoor uitgelegde verschil tussen de wijze waarop JavaScript met objecten en variabelen van een primitief type omgaat duidelijk wordt.

Zet het script in een apart `.js`-bestand.

## Opgave 3

Maak een constructor-functie voor een kassa-object. Het object heeft een property `saldo`, een methode `telOp(bedrag)` die de waarde van `bedrag` optelt bij `saldo` en een methode `geef Totaal()` die het totaal ingevoerde bedrag aflevert.

Maak vervolgens een webpagina met een invoervak waarin de gebruiker bedragen kan invoeren. Zorg dat het invoervak leeggemaakt wordt, nadat de gebruiker op Enter heeft gedrukt. Gebruik het kassa-object voor het optellen van de bedragen en maak een knop die het totaal van de ingevoerde bedragen op de webpagina toont.

## Opgave 4

- Maak een webpagina waarop de inhoud van een array wordt getoond.
- Voeg een knop toe waarmee u een getal aan de array kunt toevoegen.
- Voeg een knop toe waarmee u een getal uit de array kunt toevoegen.
- Voeg een knop toe waarmee u de getallen in de array kunt sorteren.
- Voeg een knop toe waarmee u de getallen in de array in de omgekeerde volgorde kunt sorteren.
- Laat na elke handeling de inhoud van de array zien.

### Opgave 5

Maak een webpagina waarop na het drukken op de knop de tafel van 10 netjes uitgelijnd verschijnt.  
Aanwijzing: zet zo nodig een spatie voor het betreffende getal; gebruik de pre-tag om de spatie daadwerkelijk te laten zien.

Tafel van 10				
1	*	10	=	10
2	*	10	=	20
3	*	10	=	30
4	*	10	=	40
5	*	10	=	50
6	*	10	=	60
7	*	10	=	70
8	*	10	=	80
9	*	10	=	90
10	*	10	=	100

### Opgave 6

Maak een webpagina waarop na het invoeren van een getal in een tekstvak een tabel met acht regels als in de afbeelding hieronder verschijnt. De getallen aan de rechterkant verschillen telkens 0.25 van elkaar.

100	
1	100.00
2	100.25
3	100.50
4	100.75
5	101.00
6	101.25
7	101.50
8	101.75

### Opgave 7

- Schrijf een script dat de som van de getallen van 1 tot en met 100 berekent. Maak in het script gebruik van een `for`-statement. Het juiste antwoord is 5050.
- Dezelfde opgave, maar gebruik nu een `while`-statement.

### Opgave 8

Initialiseer twee arrays `a` en `b` met evenveel getallen.

Schrijf code waarmee u in een derde array `c` de som van de overeenkomstige elementen van de eerste twee arrays opbergt, dus:

```
c[0] = a[0] + b[0]
c[1] = a[1] + b[1]
et cetera.
```

Laat de inhoud van de drie arrays op een webpagina zien.

### Opgave 9

Schrijf een functie `keerom()` die de volgorde van de elementen in een array omdraait.

Maak geen gebruik van de methode `reverse()`.

Stel `a = [1, 4, 3, 2, 5, 10]`.

Na de aanroep `keerom(a)` geldt: `a = [10, 5, 2, 3, 4, 1]`.

Test een en ander via een webpagina.

### Opgave 10

Vul twee arrays met evenveel getallen in oplopende grootte, dus bijvoorbeeld:

```
[1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
en
[2, 4, 8, 16, 32, 64, 128, 256, 512, 1024]
```

Schrijf een stukje code dat de twee arrays samenvoegt tot een derde array waarin de getallen opnieuw in oplopende grootte staan, dus:

```
[1, 2, 2, 3, 4, 5, 8, 8, 13, 16, 21, 32, 34, 55,
 64, 89, 128, 256, 512, 1024]
```

Schrijf een methode met twee parameters voor de arrays. De methode voegt de arrays samen als hierboven en levert de derde array af.

### Opgave 11

- Vraag de gebruiker een datum in te voeren in de vorm dd-mm-jjjj. Controleer of de invoer inderdaad deze vorm heeft, dat wil zeggen dat u in eerste instantie niet hoeft te controleren of het een geldige datum is, alleen of de vorm klopt.
- Controleer vervolgens of de ingevoerde string een geldige datum kan zijn. Spreek met uzelf af tussen welke jaartallen de data mogen liggen, bijvoorbeeld tussen 1900 en de huidige datum. U hoeft geen rekening te houden met schrikkeljaren.
- Zoals bekend, heeft een schrikkeljaar een dag meer: 29 februari.  
Een jaar is een schrikkeljaar als het jaartal deelbaar is door 4, maar niet deelbaar door 100 óf als het jaartal deelbaar is door 400.  
Voorbeelden van schrikkeljaren zijn: 1600, 1980, 2000, 2012, 2016.  
Voorbeelden van niet-schrikkeljaren zijn: 1800, 1900, 1981, 2015.

Met de operator `%` kunt u onderzoeken of een bepaald geheel getal deelbaar is door een ander geheel getal. Immers, als twee getallen op elkaar deelbaar zijn, is de rest van de deling gelijk aan nul. En als ze niet deelbaar zijn is de rest ongelijk aan nul.

```
var getal = ...  
  
if( getal % 10 == 0 )  
    alert( "het getal is deelbaar door 10" );  
else  
    alert( "het getal is NIET deelbaar door 10" );
```

Eventueel kunt u de uitkomst van de deelbaarheid opslaan in een (boolean) variabele. Bijvoorbeeld zo:

```
var getal = ...;  
var even = (getal % 2 == 0);
```

Als de uitdrukking `getal%2` de waarde 0 heeft, is het getal even en krijgt `even` de waarde `true`. Als het getal oneven is, krijgt `even` de waarde `false`.