Functies

- JavaScript bevat ingebouwde functies (methods), bv. console.**log()**;
- Je kan je eigen functies maken

Enkele ingebouwde string methods

.charAt(positie)

hiermee kan je opvragen welk karakter op een bepaalde positie in een string variabele staat

Opgelet: Javascript begint te tellen vanaf positie **0**

Kat
012

.substring(startIndex,eindIndex)

Hiermee kan je een gedeelte van een string opvragen

startIndex: vanaf welke positie (0 based)

eindIndex: tot **voor** welke positie

```
var test = "0123456789";

document.write(test.substring(4,9) + "<br/>");
```

Oefenen

Maak een var met inhoud JavaScript;

Doe een document.write van enkel de medeklinkers via meerdere keren een substring()

```
var test = "javascript";


document.write(test.substring(0,1) + test.substring(2,3) + test.substring(4,5) +
test.substring(5,6) + test.substring(6,7) + test.substring(8,9) + test.substring(9,10)
);
```

.indexOf(zoekString);

zoek het eerste voorkomen van een tekst binnen een string

resultaat = de index waar de tekst start

of -1 indien de tekst niet gevonden wordt

```
var alfabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

var locatie = alfabet.indexOf("CDE");

var fouteLocatie = alfabet.indexOf("smurf");

document.write('cde begint in het alfabet op index: ' + locatie + "<br>");

document.write('smurf begint in het alfabet op index: ' + fouteLocatie);
```

Oefenen

Maak een var aan voor een voornaam en een tweede voor een achternaam. Voeg deze samen in een derde var . Druk dan de derde var af op het scherm als volgt: De tekst "Hallo" en daarachter de samengestelde naam. Alleen in de derde var mag je spaties toevoegen.
Resultaat:
Hallo Danny Drijvers.

Maak een nieuwe var aan om de lengte van jouw naam te berekenen.
Resultaat:
Jouw naam bevat 13 karakters.

Maak een var aan genaamd filename. Hierin steek je de tekst: javaScriptFrameWork.
Druk deze af op het scherm.
Resultaat (let op de quotes, hiertussen staan vars):
De originele filename is: 'javaScriptFrameWork'

Zet de letters van de var om in kleine letters (zoek de method). Let op, dit is niet enkel tonen, maar echt omzetten.
Resultaat (let op de quotes, hiertussen staan vars):
De filename kan beter in kleine letters getypt worden, dus: 'javascriptframework'.

Maak 2 nieuwe variabelen aan waarin je via de substring() methode de filename opsplitst.
Resultaat (let op de quotes, hiertussen staan vars):
Als je 'javascriptframework' opsplitst, krijg je : 'javascript' en 'framework'.

Maak een nieuwe var en steek het 5e karakter (volgens JS) van de filename var erin.
Resultaat (let op de quotes, hiertussen staan vars):
Het zesde karakter (volgens mensen) in 'javascriptframework' is: 'c'.

Maak een var email met inhoud jan.klaassen@mail.com. Maak een var waarin je de positie van de @ steekt.
Resultaat (let op de quotes, hiertussen staan vars):
In 'jan.klaassen@mail.com' staat de @ op positie: 12

Maak een nieuwe var waarin je de naam opvraagt uit de email. Let op: dit moet altijd kloppen, ook als je een andere naam gebruikt, je mag dus niet een vaste eindpositie gebruiken in de substring(), maar werken met de opvraging van de positie van de @.
Resultaat (let op de quotes, hiertussen staan vars):
De naam is dus: 'jan.klaassen'

Zoek op hoe de opdracht substr() werkt.
Maak een var met de tekst: Deze tekst is te lang en zal worden afgekapt op 8 tekens m.b.v. substr().
Maak een nieuwe var en steek er de eerste 8 karakters in van de vorige var m.b.v. substr().
Resultaat (let op de quotes, hiertussen staan vars):
De originele tekst: 'Deze tekst is te lang en zal worden afgekapt op 8 tekens m.b.v. substr()' wordt dan: 'Deze tek'

Zelf functies schrijven

Wat is een functie?

om aantal statements in een blok te groeperen

in geval code herhaaldelijk gebruikt wordt

kunnen gegevens aanvaarden / aankrijgen

kunnen waarden teruggeven

functie gebruiken :

- **eerst** definiëren
- daarna **aanroepen**

De ingebouwde methods zijn al gedefinieerd, en die roep je enkel aan

eigen functie definiëren

Notatie: gebruik de JavaScript term “function” om aan te geven dan je een functie wil definiëren. Let op de **()** en **{}**

```
function naamVanDeFunctie(){

    opdrachten

    …

}
```

Vergelijk met installeren van programma op je PC

```
function groet() {

    alert(“Goedendag”);

}
```

sleutelwoord **function**
naam van de functie : groet
na de naam van de functie komen ronde **haakjes**.
statements staan tussen **accolades**
de namen van functies zijn hoofdlettergevoelig, standaard moeten ze met kleine letters beginnen, functies met een hoofdletter hebben een speciale betekenis

aanroepen = gebruik maken van functie

vergelijk met opstarten van geïnstalleerd programma

gebruik naam van functie als statement in script

met haakjes

```
groet();
```

Oefening

maak een functie “eersteCharacter”

deze bevat een var naam met inhoud “Javascript”

verder schrijft deze in het document het volgende:

```
Javascript : naam.charAt(0) : J
```

```
function eersteCharacter() {

    var naam = "Javascript";

    document.write(naam + " : naam.charAt(0) : " +
naam.charAt(0));

}


eersteCharacter();
```

Oefening

Maak een functie hallo

In de functie vraag je via een prompt de naam van de gebruiker

Vervolgens schrijf je in het document: “Hallo ” en de naam

Argumenten doorgeven

Soms is het nodig dat je gegevens doorstuurt naar de functie, zodat die er iets kan mee doen:

function naamVanDeFunctie(**argument1, argument2,...**){

    opdrachten

    …

}

De functie definitie bevat dus een soort variabele, argument genoemd dus, waarin de doorgegeven waardes opgevangen worden.

Bij het oproepen van de functie geeft je dan ook waardes door.

Vergelijking met water aan een werker doorgeven

Stel dat een hoeveelheid water moet bewerkt worden, bv. heel eenvoudig er moet grenadine aan toegevoegd worden.

Hiervoor stellen we een werker aan, de function grenadineToevoegen();

Vanuit het hoofdscript willen we dus water doorgeven aan de functie.

In het hoofdscript wordt het water bewaard in de variabele “fles”

Bij het aanroepen van de functie moet het water doorgegeven worden, dus niet de var “fles” op zich wordt doorgegeven, maar de inhoud, het water dus.

In de functie definitie is een argument voorzien, een container, om het water in op te vangen, bv. “opvangContainer”. Hiermee zal die functie dan kunnen werken.

```
var fles="water";

function grenadineToevoegen(opvangContainer){

    //voeg grenadine toe aan opvangContainer;

    document.write("grenadine is toegevoegd aan het " +
opvangContainer);

}

grenadineToevoegen(fles);
```

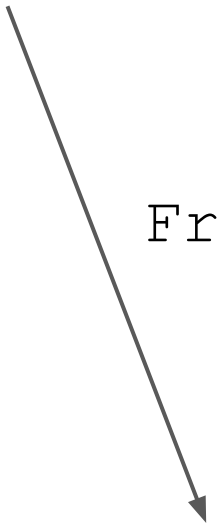
De inhoud van de fles is dus overgegoten in de opvangContainer

Voorbeeld2

```
function groet(naamPersoon) {
   alert(“Goedendag, ” + naamPersoon);
}


groet(“Fred”);
groet(“Ethel”);
```

```
groet("Fred");
```




```
function groet(naamPersoon) {
   alert("Dag, " + naamPersoon);
}
```

Best pratice

normaal wordt het ontvangen argument binnen de functie bewaard in een variabele als het argument meerdere keren gebruikt wordt in de functie (performanter)

```
function groet(naamPersoon) {

    var naam= naamPersoon;

    alert("Goedendag, " + naam);

    alert("Alles goed met de kinderen " + naam + "?")

}
```

Oefening

Pas de oefening van eerste karakter aan:

de naam wordt nu niet meer in de functie eersteCharacter gedefinieerd, maar wordt telkens doorgestuurd als de functie opgeroepen wordt

De oproepen van de functie:

```
eersteCharacter("javascript");

eersteCharacter("html");

eersteCharacter("css");
```

```
function eersteCharacter(naam) {

   document.write(naam + " : naam.charAt(0) : " +
naam.charAt(0) + "<br>");

}


eersteCharacter("javascript");

eersteCharacter("html");

eersteCharacter("css");
```

Oefening

Maak een functie definitie : hoofdletters()
deze krijgt een argument: tekst

in de definitie voeg je code toe om de aangekregen tekst om te zetten naar hoofdletters (zoek JS method op)
druk deze dan af in het document

hoofdletters(“JavaScript”);
hoofdletters(“css”);

geeft dan

JAVASCRIPT
CSS

```
function hoofdletters(tekst) {
   document.write(tekst.toUpperCase() +
"<br>");
}

hoofdletters("javascript");
hoofdletters("css");
```

Functie die een waarde teruggeeft

Functies kunnen een waarde teruggeven aan het script dat de functie heeft aangeroepen.

```
return waarde;
```

gemiddelde berekenen van 4 getallen :

in het hoofdscript hebben we een var om een berekend gemiddelde waarde in te steken

```
var score = gemiddelde(3,4,5,6);
```

en dit drukken we dan af

```
document.write("Het gemiddelde : " + score);
```

roep functie gemiddelde op en stuur 4 getallen naar de functie

```
gemiddelde(3,4,5,6);
```

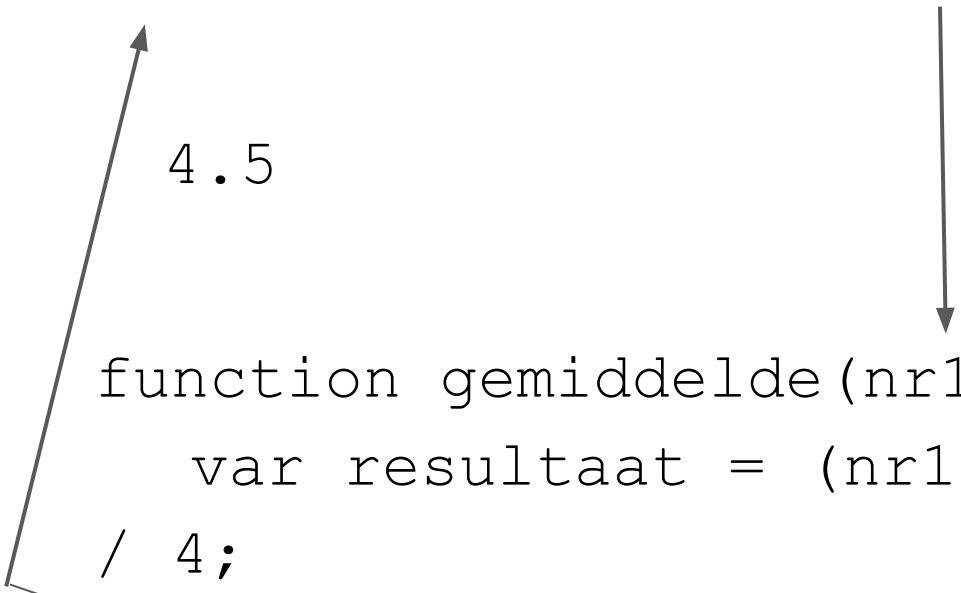
in functie worden deze getallen opgeteld, gedeeld door 4 en resultaat wordt terug gestuurd

```
function gemiddelde(a,b,c,d) {
   var resultaat = (a + b + c + d) / 4;

   return resultaat;

   }
```

```
var score = gemiddelde(3,4,5,6);
```




```
function gemiddelde(nr1,nr2,nr3,nr4) {
   var resultaat = (nr1 + nr2 + nr3 + nr4)
/ 4;
   return resultaat;
}
```

Voordeel

functie kan herhaaldelijk gebruikt worden en wat er dan juist met het resultaat moet gebeuren kan telkens anders zijn want dit staat niet in de functie

```
var score2= gemiddelde(8,5,15,16);

document.write("de score is: " + score2);
```

Oefening product

Voorzie de function "product". Deze krijgt 3 argumenten aan. Dit zijn getallen, je moet geen controle uitvoeren of het wel getallen zijn enz.

De function product berekent het product van de 3 argumenten en geeft dit terug.

Het hoofdscript:

Je maakt een var score die de function product oproept en 3 getallen doorgeeft.

Deze var score druk je af in de html pagina.

```
function product(a,b,c) {
   var resultaat = (a * b * c);
   return resultaat;
}
```

```
var score = product(3,4,5);

document.write("Het product is : " + score);
```

Oefening echo

Je bent een ervaren PHP programmeur en bent het zo gewoon dat je echo gebruikt i.p.v. document.write() dat je hiervoor een eigen functie gaat maken.

Voeg een functie definitie voor echo toe zodat volgende code werkt:

```
var tekst = "Deze tekst wordt afgedrukt op het scherm";

echo(tekst);
```

```
function echo(ontvangenTekst){

    document.write(ontvangenTekst);

}
```

Oefening

var getal = 5;

doe een document write:

Het getal 5 in het kwadraat is : 25

het kwadraat roept je op via een nieuwe functie kwadraat()

deze krijgt een argument aan: ontvangenGetal

in de functie wordt een var uitkomst gemaakt met de ontvangenGetal * ontvangenGetal

deze uitkomst wordt gereturnd

```
var getal = 5;


function kwadraat(ontvangenGetal){
    var uitkomst = ontvangenGetal * ontvangenGetal;
    return uitkomst;
}


document.write("Het getal " + getal + " in het kwadraat is :
" + kwadraat(getal));
```

Oefening

var naam = prompt("Geef je naam:");

We gaan de controles automatiseren

maak een functie controle

deze krijgt een argument aan: waarde

als waarde null is of als de lengte van waarde 0 is -> return "error"

anders return "ok"

document write : de prompt geeft error/ok

```
var naam = prompt("Geef je naam:");
function controle(waarde){
var waardeInFunction = waarde;
if(waardeInFunction == null || waardeInFunction.length==0){
return "error";
}
else{
    return "ok";
}
}
document.write("De prompt geeft : " + controle(naam));
```

Oefening

Via een prompt vraag je een geboortejaar op

Dit jaar wordt doorgegeven aan een functie leeftijdBerekenen()

Controleer in de functie op annuleren en niets ingegeven.

Controleer vervolgens op het een nummer is

Controleer vervolgens of de lengte 4 is

Als alles goed is, vraag dan het huidige jaar op (zoek op)

en bereken hoeveel jaar de persoon dit jaar wordt

Roep een tweede functie op: schrijf() en geef de leeftijd door

In de functie schrijf wordt in het document geschreven:

Dit jaar wordt u x jaar.