



Webscripting

Hoofdstuk 2

Program structure

DE HOGESCHOOL MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook



Expression

expression stuk code dat een waarde produceert
expressions kunnen sub-expressions bevatten

true

12

12 + 3

12 + 3 * (1 + a)

3 * (2 + Math.sqrt(1 + Math.sqrt(64)))

64 is een expression, maakt de waarde 64

Math.sqrt(64) is een expression, maakt de waarde 8

1 is een expression, maakt de waarde 1

1 + 8 is een expression, maakt de waarde 9

Math.sqrt(9) is een expression, maakt de waarde 3

2 is een expression, maakt de waarde 2

2 + 3 is een expression, maakt de waarde 5

3 * 5 is een expression, maakt de waarde 15



Statement

statement instructie, opdracht
meestal na statement een puntkomma

```
1;           // zinloze statement  
             // produceer de waarde 1  
             // er gebeurt verder niets
```

```
let a = 12 + 3;  
        // maak de variabele a aan geef  
        // deze waarde 15 (resultaat v.  
        // expression 12 + 3)
```

```
a = a + 1;  
        // verhoog a met 1 (resultaat v.  
        // expression a + 1
```



Bindings

Variabele aanmaken via `let`, `const`

```
let i = 1;  
const j = 12;
```

`const`: niet wijzigbaar via assignment operator (`=`, `+=`, `++`, `--`, ...)

`let`: wel wijzigbaar via assignment

Variabele is een voorbeeld van een **binding**:

naam wordt verbonden met een waarde

Later extra voorbeelden van bindings: e.g. function binding

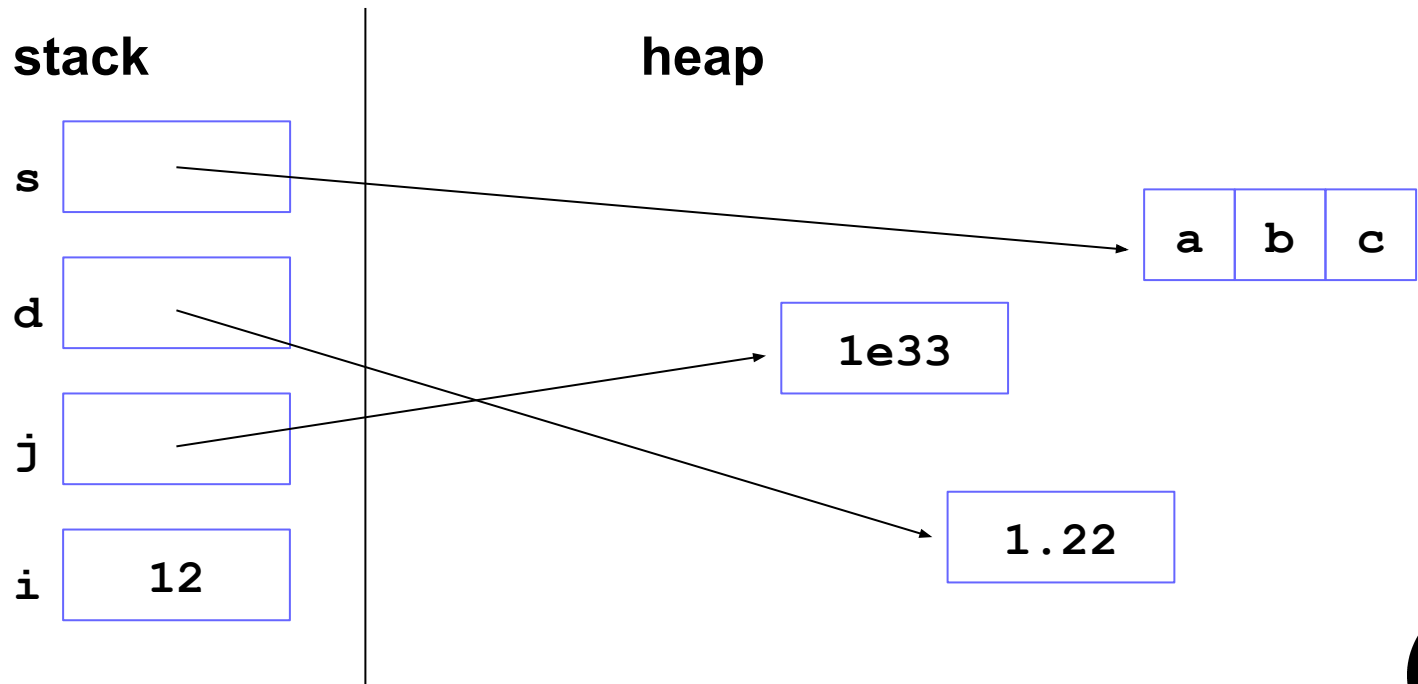
naam van de function wordt verbonden met een uitvoerbaar stuk code



Bindings

Primitive

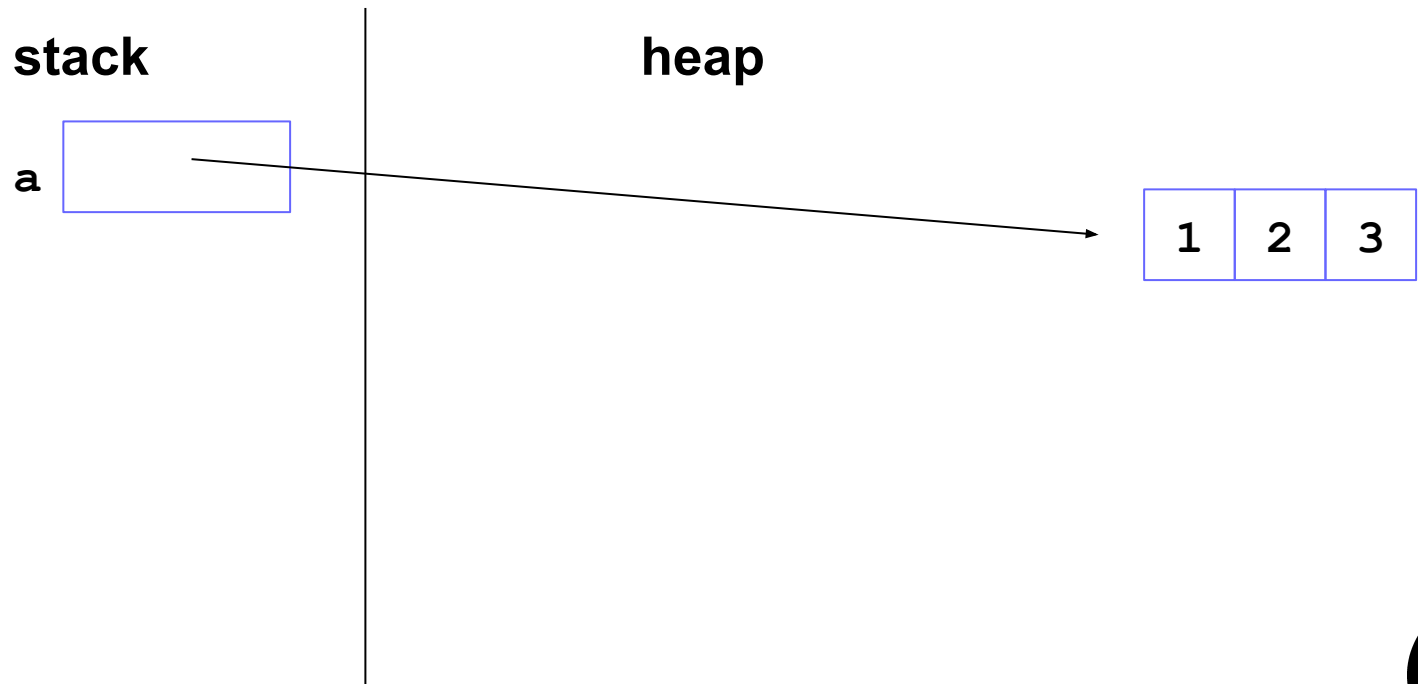
```
let i = 12;           // small integer (smi)
let j = 1e33;
let d = 1.22;
let s = "abc";
```



Bindings

Reference

```
let a = [1,2,3];
```

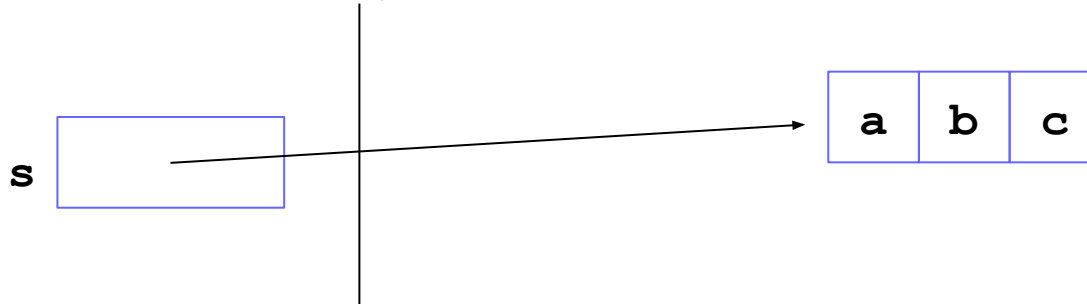


Bindings

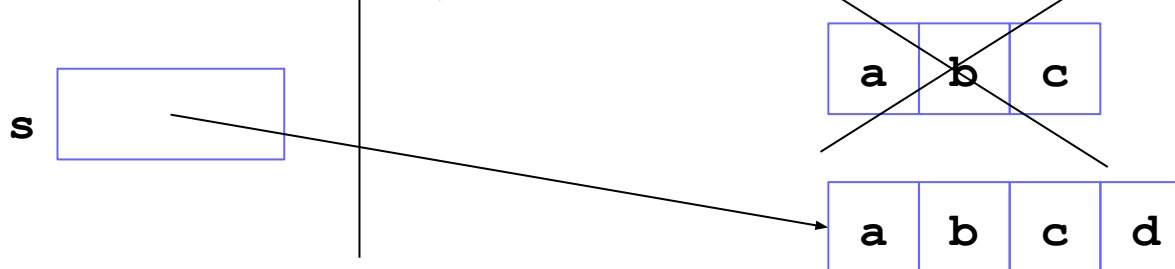
Garbage collection

wanneer er geen referentie naar een geheugenlocatie is, wordt de locatie vrijgegeven

```
let s = "abc";
```



```
let s = s + "d";
```



primitive datatypes: immutable, reference datatypes: mutable



Binding names

- letters, hoofdletters
- getallen: catch22
(niet beginnen met een getal)
- dollar-teken \$ en underscore _
- geen keywords:

break case catch class const continue debugger default
delete do else enum export extends false finally for
function if implements import interface in instanceof let
new package private protected public return static super
switch this throw true try typeof var void while with yield



Environment

- verzameling van bindings
- nooit leeg, zelfs als we zelf geen variabelen / functies / objecten maken

bijvoorbeeld in browser:

functies `alert`, `prompt`, `console.log(...)`



Sequentie



```
let theNumber = Number(prompt("Pick a number"));  
  
console.log("Your number is the square root of " +  
    theNumber * theNumber);
```

prompt: geeft een string terug

Number(...): omzetting naar number



if



```
let num = Number(prompt("Pick a number"));  
if (num < 10) {  
    console.log("Small");  
} else if (num < 100) {  
    console.log("Medium");  
} else {  
    console.log("Large");  
}
```



if



```
let theNumber = Number(prompt("Pick a number"));
if (!Number.isNaN(theNumber)) {
    console.log("Your number is the square root of " +
        theNumber * theNumber);
}

let theNumber = Number(prompt("Pick a number"));
if (!Number.isNaN(theNumber)) {
    console.log("Your number is the square root of " +
        theNumber * theNumber);
} else {
    console.log("Hey. Why didn't you give me a number?");
}
```



switch



```
let answer = prompt("What is the weather like?");
switch ( answer ) {
  case "rainy":
    console.log("Remember to bring an umbrella.");
    break;
  case "sunny":
    console.log("Dress lightly.");
  case "cloudy":
    console.log("Go outside.");
    break;
  default:
    console.log("Unknown weather type!");
    break;
}
```



while



```
let number = 0;
while (number <= 12) {
  console.log(number);
  number = number + 2;
}
console.log(number);
```

```
let result = 1;
let counter = 0;
while (counter < 10) {
  result = result * 2;
  counter++;
}
console.log(result);           // → 1024
```



do while



```
let yourName;  
do {  
    yourName = prompt("Who are you?");  
} while (!yourName);  
console.log(yourName);
```



for



```
for (let number = 0; number <= 12; number += 2) {  
  console.log(number);  
}
```

```
for (let current = 20;      ; current++) {  
  if (current % 7 == 0) {  
    console.log(current);  
    break;  
  }  
}
```



Besluit

Variabele is binding van naam met waarde

Programma bestaat uit statements

Meestal bevatten statements expressions

Expressions kunnen bestaan uit sub-expressions

Sequentie

Opsplitsing: if-else, switch

Looping: while, do while, for

