JavaScribt behind the scene

Abdelrahman Shaheen

OUTCOMES

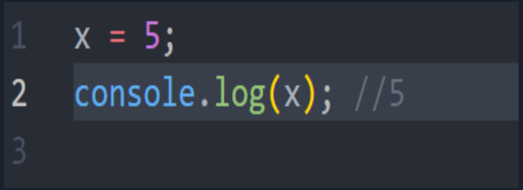
- Strict mode
- "This" keyword *extremely important*
- Regular Functions VS Arrow Functions : (When to use or avoid each of them)
- How Primitives and Objects are stored in memory ?
- Primitives and Objects in the context of functions.

Strict mode

- You’ll be strict when writing code ☺
- Strict mode makes it easier to write "secure" JavaScript.
- Keywords reserved for future JavaScript versions can NOT be used as variable names in strict mode.
- Strict mode changes previously accepted "bad syntax" into real errors.
- We use it to prevent some weird behavior when working with **“this”**

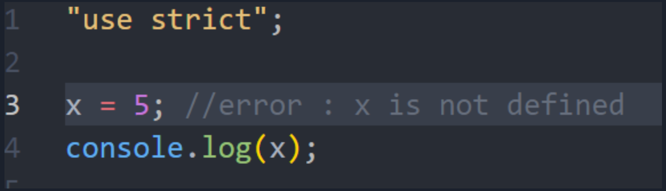
Strict mode code example

Bad Syntax but works



```
x = 5;
console.log(x); //5
```

Not allowed in strict mode



```
"use strict";

x = 5; //error : x is not defined
console.log(x);
```

- For more information read this : strict-mode

Execution context

Every EC has these :

- ✓ Variable environment. (Discussed)
- ✓ Scope Chain. (Discussed)
- ❑ This key word (I’ll talk about it today)

This key word

- Every EC has it’s own “this” keyword
- In JS “this ” refers/points to the object who owns the function/EC

This key word

- Not static ? Depends on how the function called it ! How ? :

0. In global context this points to window object. (see code 0)

1. In a method → this points to the object how called it. (see code 1)

2. In simple function call (regular function/function expression) → this points to undefined (in strict mode) ,otherwise

   It points to the window object. (see code 2)

3. In Arrow functions → it does not have it's own "this" keyword ,it takes it's this from the first outer scope. (see code 3) .Note : Do not use arrow functions inside and object or a class when you use "this".

4. In event listener → it points to the object(DOM element) who attached to the event listener. (see code 4)

This key word

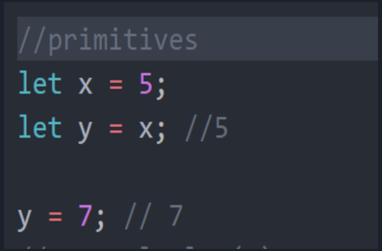
For more information : this key word

Objects/Reference Types VS Primitive Types

- Primitive types
  - Number , string ,boolean ,null ,undefined ,null
  - Stored in the call stack
- Objects/Reference Types
  - Object literal .arrays ,functions , …..etc.
  - Stored in heap . Why ? Take more memory and the heap is unlimited.

See the code example

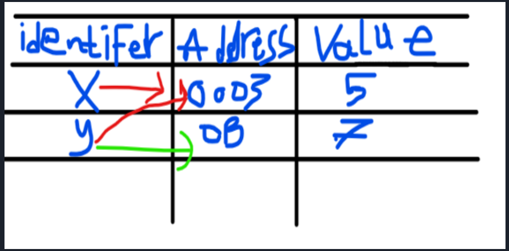
Primitives Vs Reference Types in Memory ?

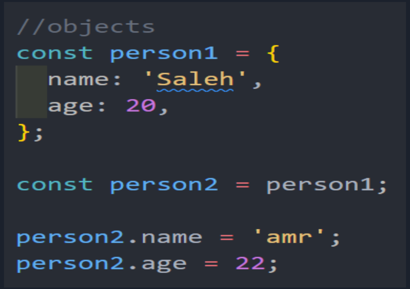


```
//primitives
let x = 5;
let y = x; //5

y = 7; // 7
```

Call Stack





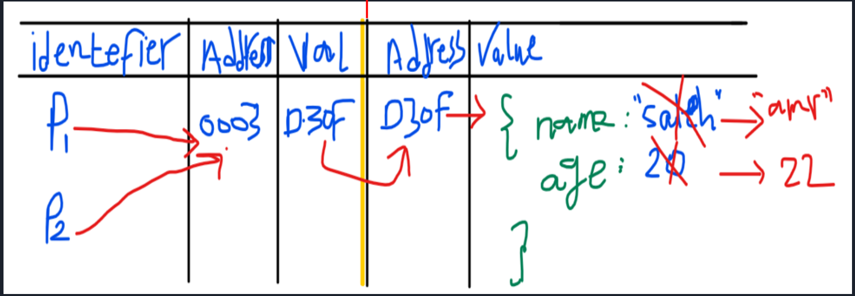
```
//objects
const person1 = {
  name: 'Saleh',
  age: 20,
};

const person2 = person1;

person2.name = 'amr';
person2.age = 22;
```

Call stack

Heap



Resources to study

- strict mode
- this keyword
- Play with my code examples
- https://www.youtube.com/watch?v=gvicrj31JOM
- https://www.youtube.com/watch?v=eOI9GzMfd24
- https://www.youtube.com/watch?v=QCRpVw2KXf8

Tasks

- Play with my code ,Study the Session well.
- Search for “this in event handler” and you will demonstrate the concept to me.