A Short Analysis of Java Script

Mini Assignment -3

Kartik Srinivas - ES20BTECH11015
Ojjas Tyagi - CS20BTECH11060

A Course Homework Assignment

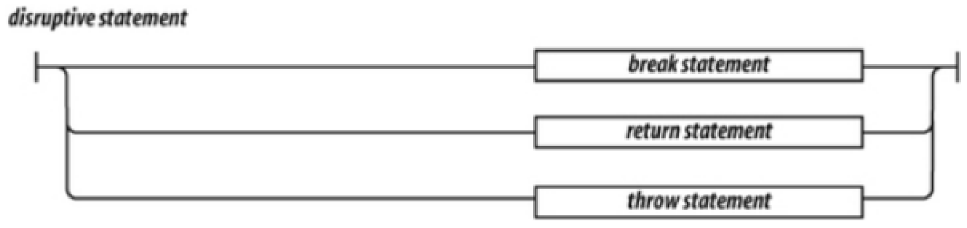
Indian Institute of Technology
Hyderabad

April 23, 2022

1 Grammatical analysis

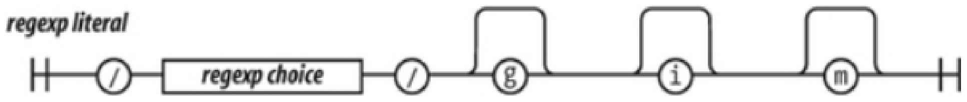
1.1 Labelled breaks

Breaks can take a optional argument to label where in C we have to use goto. Note that even though goto is a reserved word in JS , there is no usage of it! There are only three Disruptive Statements



1.2 RegExp Literals

JS has a regexp Object literal, we can use this to find certain patterns in strings or files etc. (pretty much like the grep command) There are also additional flags to search globally (g), Case insensitive Search (i) and Multiline search for (m). The regular expression is written between to "/" characters.



1.3 One var to rule them all

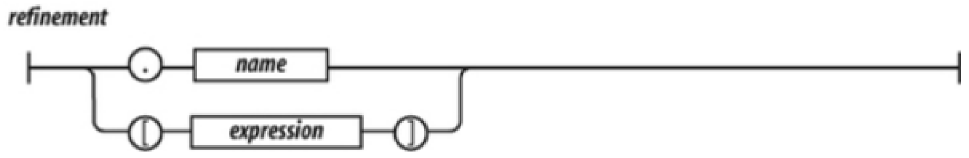
No type specifier is asked in grammar as JS is a dynamically typed language.Hence after declaring a variable its type is decided by value assigned to it and can later be subsequently changed.

1.4 Characterless Language

There is no character type in js only strings. In order to define a character I actually need to define a string with a single character : \

1.5 Refinement using [ ] instead of .

We can access a field in object using obj[fieldname] as well as obj.fieldname.This allows for cases when which field is to be accessed is unknown at compile time.

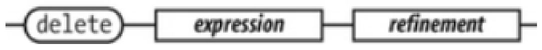


1.6 No Numeric Nightmares

All numbers in js are internally stored as a 64 bit floating point and hence there is no int,float etc types and no cast/rounding errors as well.

1.7 Delete properties

Properties of objects such as some specific field can be deleted in javascript using delete keyword. This is clearly a very unsafe thing to do!



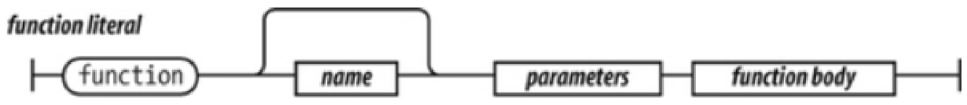
2 Bizzare Stuff About JS

2.1 Objects, Objects and Objects

In Java Script arrays are objects, regular expressions are objects, functions are objects. Even numbers are object like,i.e. they can have methods. Also objects in JS are **Class free** i.e you can define an object without even having a class for it

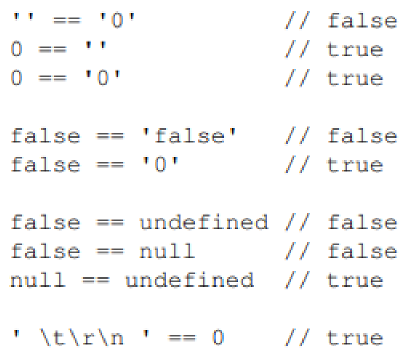
2.2 Nameless Functions!

Functions can be anonymous in JavaScript! The railroad diagram can skip the ”name” and move to parameters and the function body!



2.3 ”===” and the Evil Twins

Java Script uses equality operators of the form === and !== ,they behave normally. But there are other ”evil twins” which attempt to make both the values same , here is a glimpse of how weird it can get:-



```
'' == '0'           // false
0 == ''             // true
0 == '0'            // true

false == 'false'    // false
false == '0'        // true

false == undefined  // false
false == null       // false
null == undefined   // true

' \t\r\n ' == 0     // true
```

2.4 Scopes

Blocks in JavaScript do not create scopes! , unlike how C works. The curly braces in C allow us to define new variables with the same name as outside ones within it. But this will merge into a single variable in Java Script.

2.5 Primitive and Reference Value type

JS has both primitive types of the form :- String , boolean, number , undefined and null; and it has a reference type called ”Object” however consider the following code :-

```
let k = null;
typeof(k) -----------"THIS YIELDS OBJECT :("
```

Despite null being a primitive value type, we get k as an object type.

2.6 \ operator

The \ division operator can return a non integer answer even if both the input types are integers (this is because JS has a single number literal. 1 and 1.0 are interpreted the same)

2.7 Bitwise Operators

In Java Script the bitwise operators operate by converting their operands to integers , then perform the operations and then convert them back to the ”number form” (Java script has only a single number form of double precision type) No errors are raised when we use bitwise operators and therefore an ”&” which may have been mis-typed for ”&&” will hide like a pin in the haystack of errors