JavaScript Cheat Sheet

|  |  |  |
| --- | --- | --- |
| **Serial (video number)** | **Code/Topic** | **Explanation** |
|  | Datatypes | Javascript has 5 data types.   * Numbers (whole/fractional/negative) * Strings ( “ ” ) * Booleans (True/ false) * Null (explicitly nothing) * Undefined (doesn’t have a value yet) |
|  | jS escape characters | \ (backslash)  Ex: “Hello I am \“ maruf\” ” |
|  | Variable creation | To make a variable we use  *var* VariableName = value  However, we have something new that was introduced in 2015. Let and const.  Var – global scope  LET and CONST – block scope   * Const is more like final variable in java. For example,   If we have, const a = 25;  Then a = 24; Would give an error.   * The location is defined it cant be changed however when It comes to reference we cant change it. * Let can be updated. Similar to var however but it’s a block object. * Var and let can be undefined const cant. * Const and let cant be redeclared. |
|  | Variable hoisting | Variable are created first then they are initialized. That is why  Console.log(cat); creates an error where as,  Console.log(cat);  Var cat = “abc”  This doesn’t throw an error. Just prints undefined. |
|  | alert(value) | Pop up msg |
|  | Promt(“value”) | Users can give input |
|  | == | Compare values |
|  | === | Compare values and types |
|  | Function | Function giveName (arguemnts){  }  giveName(argument);  ex: function area (len, width){  console.log(len\*width);  }  Area(5,6);   * 30   We can also return same as java..  Function declaration: normal creation of function  Fucntion expression: assigning the function to a variable. Thus reassigning the variable would end delete the function. |
| 1. 153 | SetInterval(method, time in ms) | Set interval calls the corresponding method. Here we aren’t the ones calling the method that’s why we are not putting any parenthesis here.  This is called a higher order function.  If we write a function inside a higher order function like,  setInterval(function(){  },2000); |
| 1. 155 | Arrays | Same as java. However,  It supports any type of data inside.  it supports dynamic addition. 2 legnth array. Want to add a new item?  Array[2] = “mango”;  Array is now of length 3.  For more info go to MDN |
| 1. 156 | Array addition | Array.push(“content”) <= same as list  Array.pop(“content”) <= same as list  Array.shift(); <= pops the leftmost element and returns it.  Array.unshift(“value”); <= pushes the value in the leftmost element (array[0]) |
| 1. 163 | forEach(parameters) | If you only give one argument, then it's the element, every time.  If you give two, then it's the element, then the index, in that order.  If you give three, then it's element, index, and array, in that order.  Ex:  Array.forEach(a){  Console.log(a);  } |
| 1. 166 | Function name VS function name () | Just calling the function wouldn’t execte the code however when we say function()then the code is ececuted. |
| 1. 166 | Creating our own method on array | Array.prototype.functionName = function(parameter){  Function content  //this refers to the Array on which the method/function is called on.  } |
| 1. 167 | JS objects | Array isn’t the best option all the time. We can use JS objects which stores values in a key-value pair.  Var person ={  name: “”,  id: 21,  city: “NYC”  } |