***JavaScript - Verkefni 2***

1. ECMAScript er undirhópur af JavaScript. JavaScript er basically ECMAScript í grunnin en byggir svo upp á það.
2. JavaScript er Loosely Typed af því að það þarf ekki variable til vera definað.
3. Þú þarft að nota ‘’ þar sem þú vilt að “” birtast innan strengsins sem er gott þannig að þú þurfir ekki að forðast það að setja gæsalappir í textan þinn eða öfugt. Annars er enginn munur.
4. Undefined er hlutur sem ekki hefur verið gefið gildi, en svo er Null það er með gildi en gildið er ekkert. Það er í sjálfum sér engin munur á þeim það er eiginlega bara formsatriði.
5. Munurinn er sá að === passar að þetta séu sömu data týpur eins og 3===3 er true en “3”===3 er false. En == er alveg sama hvaða data týpa þetta er eins og 3==3 er true og “3”==3 er true.
6. Munurinn er “scope-ið”. Var er scopað í næsta function blockið á meðan let er scopað í næsta enclosing block, sem getur verið minni en function blokk. Bæði var og let eru global breytur ef fyrir utan einhvað block.

## Function Declarations:

function myFunction(a, b) {

return a \* b;}

Function Expression:

var x = function (a, b) {return a \* b};

var z = x(4, 3);

Arrow Functions:

var x = (i) => i+2

x(2);

1. Strict mode er mjög gott ef þú vilt hafa alveg villu lausan kóða, þegar þú ert í Strict mode þá þarf kóðinn þinn að vera 100% málfræðilega réttur. Til dæmis venjulega kæmistu upp með að gera semí kommu fyrir aftan þar sem það þarf ekki semí kommu en í Strict mode gefur það þér villu og þá virkar scriptið ekki.
2. Þessi kóði virkar afþví að það er búið að defina með x(); , ef það er tekið út þá virkar kóðinn ekki því þá er þetta undefined.
3. Alert-ið birtir 8 afþví að það keyrir fyrst functionin og svo er returnið keyrt

Keyrslu röð:

1 function foo(){

2 function bar() {

return 3; }

4 return bar();

3 function bar() {

return 8; }

}

5 alert(foo());

1. Ef þú ert inn í glopal scope-i þá er engin munur að að sleppa var/let en ef þú ert t.d inni í function þá er var/let local en það sem er það ekki leitar það upp scope keðjuna þar til það finnur global breytu. Þannig í sjálfu sér gerist ekkert sérstakt.
2. Það sem þetta gerir er að það birtir “alert” sem í stendur Hello World.

Þessi svigi inniheldur expressions sem á að framkvæma => (function(þetta er function svigin þannig ef þú vilt fá t.d object í functionið þá er það gert hér) { þessir slaufusvigar sína þér hvar functionið er alert(hér er það sem kemur í alertið 'Hello World'); })

(þessi svig framkvæmir expressionin );

1. JavaScript hefur tvö “scopes” það eru global og local. Variable fyrir utan function er global sem þýði það að það er hægt að nota þetta variable allstaðar. Ef variableið er búið til inni í function er Local sem er þannig að það er ekki hægt að nota hana fyrir utan þetta function.
2. Call Stack er gagna grind sem skráir function köllin. Í hvert skipti sem við köllum á functionið þá er því bætt á “stack”-inn þess vegna er það Call stack.