JavaScript Code examples

JavaScript Arrow function expressions

```
((a)=>console.log(a))('Hello World');
let val = 'test';

const test1 = function (a,b,c){
  return a + b + c;
};
const test2 = (a,b,c) => {
  console.log(a);
  return a + b + c;
}
const test3 = (a,b,c) => a + b + c;

const test4 = a => a + 50 * 10;

const test5 = (c,a=10,b=20)=> a + b + c;

val = test1(5,10,15);
val = test2(5,10,15);
val = test3(5,10,15);
val = test4(5);
val = test5(50);

const myObj = {
  a : 5,
  b : () => console.log('Hello'),
  c : function() {
     console.log(this);
  },
```

```
  d : () => console.log(this)
}

myObj.b();
myObj.c();
myObj.d();

const output = document.querySelector('.output');
output.addEventListener('click',(e)=>{
  console.log('clicked');
})
output.addEventListener('click',function(e){
  console.log('click 2');
})

console.log(val);
output.innerHTML = val;
console.clear();
const arr = [1,2,3,4,5];
const temp = arr.reduce((a,b)=>a+b);
console.log(temp);

const temp1 = arr.filter(a => a > 2);
console.log(temp1);

const temp2 = arr.map(a => a*2);
console.log(temp2);
```

JavaScript Console Object

```
let myStr = "Output String";
const myObj = {prop1:"Value 1",prop2:"Value 2"}
const myArr = ["one","two","three","four"];
const val = myObj;
console.group("logging");

logger('log','blue');
console.log(val);
logger('error','Red');
console.error(myArr);
logger('dir','green');
console.dir(document);
//console.clear();
for(let i=0;i<10;i++){
  logger('count','purple');
  console.count("test");
}
console.count("test");
//console.clear();
logger('info','pink');
console.info(val);
logger('warn','blue');
console.warn(val);

logger('table','green');
console.table(myStr);
console.table(myObj);
```

```
console.table(myArr);
console.table([myArr,myObj,myArr]);

//console.clear();

console.time('test1');
console.time();
let x = 0;
while (x <50000) {x++}
console.timeEnd();
console.timeEnd('test1');

//console.clear();
console.group();
console.log("one");
console.log("two");
console.log("three");
console.groupEnd();

console.groupEnd("logging");

function logger(met,bg){
  //console.count("test");
  console.log(`%c CONSOLE Method ${met.toUpperCase()}
`,`background:${bg};color:white`);
}
```

Object array destructing

```
let a,b,c,d,e,arr1;
const arr = [5,10,15,20,25,30,45,50];
[a,b,c,d,e,...arr1] = arr;
console.log(a,b,c,d,e);
console.log(arr1);
console.log(arr1[0]);
let f,g,myObj;
({f,g,...myObj}={f:100,g:200,h:300,i:400,j:500});
console.log(f,g);
console.log(myObj);

function fun() {
  return [10,20,30,40,50];
}

let h,i,j;
[h,,i,,j] = fun();
console.log(h,i,j);

const people = [{
  first : "Laurence",
  last : "Svekis",
  fav : "JavaScript",
  id : 100
},{
```

```
  first : "Linda",
  last : "Jones",
  fav : "HTML",
  id : 20
}]


const output = document.querySelector('.output');
const jsonFile = 'datajson2.json';
window.addEventListener('DOMContentLoaded',getData);
function getData(){
  fetch(jsonFile )
   .then(rep=>rep.json())
   .then(data=>{
      outputGen(data);
   })
}
function outputGen(people){
  people.forEach((person)=>{
     const {first,last,id} = person;
     const div = document.createElement('div');
     div.textContent = `First : ${first} Last : ${last} ID(${id})`;
     output.append(div);
  })
}
```

JavaScript template literals

```
let val1 = `Hello
        '"World`;
let a = 'Laurence Svekis';
val1 = `Hello ${a}`;
val1 = `Adding ${5+10}`;
val1 = `backtick \` `;
val1 = `Line 1
     Line 2`;
const b = 10;
const c = 33;
val1 = `${b} + ${c} = ${b+c}`;

const first = 'Laurence';
const last = 'Svekis';
function fullName(greeting,fName,lName){
  console.log(greeting);
  console.log(fName,lName);
  return `${greeting[0]} ${fName} ${lName}`;
}
val1 = fullName`Welcome${first}NEW${last}MIDDLE${last}END`;

const game = {
  level : 5,
  name : 'Laurence'
}
```

```
function checker(output,user,level){
  const statusVal = level > 3 ? 'pro' : 'beginner';
  console.log(output);
  return `${output[0]} : ${user} is a ${statusVal} at level "${level}"`;
}

val1 = checker`Player${game.name}A${game.level}B`;
game.level = 2;
val1 = checker`Player${game.name}${game.level}`;

document.querySelector('.output').innerHTML = val1;
console.log(val1);
```

Comparing Data Type and automatic conversion of Data Types

```
let a = [10];
let b = 10;
let c = "10";
if(a == b) console.log(`${a} == ${b}`);
if(a == c) console.log(`${a} == ${c}`);
if(a === b) console.log(`${a} === ${b}`);
if(a === c) console.log(`${a} === ${c}`);
if(a != b) console.log(`${a} != ${b}`);
if(a != c) console.log(`${a} != ${c}`);
if(a !== b) console.log(`${a} !== ${b}`);
```

```
if(a !== c) console.log(`${a} !== ${c}`);
let d = [];
let e = 0;
let f = false;
let g = '';
let h = null;
let i = undefined;
let j = NaN;
console.clear();
if(d == e) console.log(`D ${d} == ${e}`);
if(e == f) console.log(`E ${e} == ${f}`);
if(f == g) console.log(`F ${f} == ${g}`);
if(g == h) console.log(`G ${g} == ${h}`);
if(h == i) console.log(`H ${h} == ${i}`);
if(i == j) console.log(`I ${i} == ${j}`);
if(j == d) console.log(`J ${j} == ${d}`);
if(g == d) console.log(`GD ${g} == ${d}`);
if(f == h) console.log(`FH ${f} == ${h}`);
if(f == i) console.log(`FI ${f} == ${i}`);
if(f == j) console.log(`FJ ${f} == ${j}`);
console.clear();
if(!d) console.log(`D Yes ${d}`);
if(!e) console.log(`E Yes ${e}`);
if(!f) console.log(`F Yes ${f}`);
if(!g) console.log(`G Yes ${g}`);
if(!h) console.log(`H Yes ${h}`);
if(!i) console.log(`I Yes ${i}`);
if(!j) console.log(`J Yes ${j}`);
```

```
if(d === e) console.log(`D ${d} === ${e}`);
if(e === f) console.log(`E ${e} === ${f}`);
if(f === g) console.log(`F ${f} === ${g}`);
if(g === h) console.log(`G ${g} === ${h}`);
if(h === i) console.log(`H ${h} === ${i}`);
if(i === j) console.log(`I ${i} === ${j}`);
if(j === d) console.log(`J ${j} === ${d}`);
```

JavaScript Immediately invoked functions coding examples

```
(function(){
  console.log('ready');
})();
(()=>{
  console.log('ready arrow');
})();
((a=1,b=2,c=3)=>{
  console.log(a,b,c);
  const val = a * b * c;
  console.log(val);
})(5,6,7);
```

JavaScript Adding Numbers Array

```
const arr1 = [32,43,556,2,3,4,4345];
let total = 0;
for(let i=0;i<arr1.length;i++){
  console.log(arr1[i],total);
  total += arr1[i];
}
console.log(total);
/*
total = 0;
for(let i in arr1){
  console.log(arr1[i],total);
  total += arr1[i];
}
console.log(total);
*/
total = 0;
for(let i=0,len=arr1.length;i<len;i++){
  console.log(arr1[i],total);
  total += arr1[i];
}
console.log(total);
```

Array Filter Method

```
const arr1 = ["Laurence","Svekis","Hello","World","Cat"];
```

```
const arr2 = arr1.filter((val,ind,arr)=>{
  console.log(val,ind,arr);
  return val.length > 5;
})

const arr3 = arr1.filter(callbackFun);

const arr4 = arr1.filter(val => val.length > 5);

function callbackFun(val,ind,arr){
  console.log(val,ind,arr);
  return val.length > 5;
}

console.log(arr1);
console.log(arr2);
console.log(arr3);
console.log(arr4);
console.clear();

const arr5 = [3223,234,5,23,23333,43,34,34,34,34];

const arr6 = arr5.filter(val => val > 100);
const arr7 = arr5.filter(checker);

function checker(val){
  console.log(`${val} = ${val % 2}`);
```

```
  return val % 2;
}

console.log(arr6);
console.log(arr7);

const arr8 = [0,1,0,1,1,1,1,true,false,true];
const arr9 = arr8.filter(val => val);

console.log(arr9);

const arr10 = arr5.filter(checker2);
function checker2(val,ind,arr){
  console.log(`${arr.indexOf(val)} = ${ind}`);
  return arr.indexOf(val) == ind;
}

console.log(arr10);

const arr11 =
["Laurence","Svekis","Hello","World","Cat","Laurence","Svekis","Cat","Laurence","Svekis","Cat","Laurence","Svekis","Cat","Laurence","Svekis","Dog","Laurence","Svekis",];
console.log(arr11.filter(checker2));

const arr12 = [{name:"Svekis 1",id:10},{name:"Svekis 2",id:1},{name:"Svekis 3",id:100},{name:"Svekis 4",id:50},{name:"Svekis 5",id:20}];
```

```
const arr13 = arr12.filter((val)=>{
  return val.id;
})

console.log(arr13);

const validID = [50,100];
const arr14 = arr12.filter((val)=>{
  return validID.includes(val.id);
})

console.log(arr14);
```

Array includes method for arrays

```
const arr = ["Svekis","Laurence",1000,20,300,true,323,"Svekis"];
const arr1 = arr.map(String);
console.log(arr1);
const myInput = document.querySelector('input');
const btn = document.querySelector('button');
const output = document.querySelector('.output');
btn.addEventListener('click',(e)=>{
 const val = myInput.value;
 const result = arr1.includes(val);
 let message;
 if(result){
    message= `${result} is found in the array content`;
```

```
 }else{
    message = `${result} is NOT found`;
 }
 output.textContent = message;
 console.log(result);
})
let val = arr.includes("Svekis");
console.log(val);
val = arr.includes("Svekis",-1);
console.log(val);
```

How to remove and update array items

```
const arr1 = ['FIRST','Laurence','Svekis',100,false];
const arr2 = ['SECOND',2332,true,'Hello','Svekis',400,false];
const arr3 = arr1.concat(arr1,arr2);
const arr4 = arr1;
arr2.push(arr1);
console.log(arr2);
console.log(arr3);
Array.prototype.push.apply(arr1,arr2);
console.log(arr1);
arr2.push('NEW ITEM');
arr1.push('Arr 1 New');
//delete arr1[12];
```

```
//delete arr2[7];
//arr1.length = 0;
arr1.splice(12,1,'REMOVED','SECOND');
arr2.splice(3);
console.log(arr1);
console.log(arr2);
console.log(arr3);
console.log(arr4);
```

Every Method for array

```
const arr1 = [1,4,54,23,5,7,34,66,1000];
const arr2 = ["Svekis","a","World","He"];
const arr3 = ["Svekis","Laurence","Cat","Hello","a","World"];
let val = arr1.every((ele)=>{
  console.log(ele);
  return ele < 10;
})
val = arr1.every(callBackFun);
function callBackFun(ele,ind,arr){
  console.log(`${ele} ${ind}`);
  return ele < 1000;
}
console.log(val);
console.log(arr1.every(x => x < 10000));
console.log(arr1.every(x => x < 1000));
console.clear();
val = arr2.every((ele)=>{
```

```
 console.log(ele);
 return ele.length < 15;
})
console.log(val);
console.clear();
val = arr2.every((ele)=>{
 console.log(ele);
 return arr3.includes(ele);
})
console.log(val);
```

JavaScript Logical Conditions

```
let val = 5;
val = 6;
if(val == 5) {fun1(1);}
if(val == 5) fun1(2);
val == 5 && fun1(3);
val == 5 || fun1(4);
if(val != 5) fun1(5);
fun1();
let val2;
val2 = val2 || 100;
console.log(val2);
function fun1(v=1){
 v = v || 1;
 console.log(`Hello ${v}`);
```

```
}
```

Array map

```
const arr1 = [1,5,7,8,23,342,2,3,4];
const arr2 = arr1.map((val,ind,arr)=>{
 console.log(val,ind,arr);
 return val * val;
})
const arr3 = arr1.map(val => val*val);
const arr4 = arr1.map(callbackFun);
function callbackFun(val){
 return val * val;
}
console.log(arr1);
console.log(arr2);
console.log(arr3);
console.log(arr4);
```

Serialization and deserialization

```
const output = document.querySelector('.output');
const arr1 = ["Svekis","Laurence",1000,true];
const obj1 = {
 first : "Laurence",
 last : "Svekis",
 id : 1000,
 num : 55,
```

```
  mes : "Hello",
  boo : false
};
output.textContent = arr1;
output.textContent = obj1;
console.log(arr1);
console.log(obj1);
let val = JSON.stringify(obj1)
val = JSON.stringify(arr1);
console.log(val);
output.textContent = val;
output.textContent += arr1;
console.clear();
val = JSON.stringify(obj1,rep);
function rep(key,value){
  console.log(key,value);
  if( typeof value === 'number'){
     return undefined;
  }
  return value;
}
output.textContent = val;
val = JSON.stringify(obj1,['num','first']);
output.textContent = val;
val = JSON.stringify(obj1,null);
output.textContent = val;
console.log(val);
val = JSON.stringify(obj1,null,' ');
```

```
output.textContent += val;
console.log(val);
const str1 = JSON.stringify(obj1,['num','first']);
const str2 = JSON.stringify(arr1);
console.clear();
console.log(str1);
console.log(str2);
const ob1 = JSON.parse(str1);
const ob2 = JSON.parse(str2);
console.clear();
console.log(ob1);
console.log(ob2);
```

Storing to local storage

```
<!DOCTYPE html>
<html>
 <head>
    <title>JavaScript Course</title>
 </head>
 <body>
    <input type="text">
    <button>Check</button>
    <div class="output">Output</div>
    <script src="code6.js"></script>
 </body>
</html>
```

```
const str1 = "Laurence Svekis";
const myObj = {
  first : "Laurence",
  last : "Svekis",
  id : 100
};
const myInput = document.querySelector('input');
const btn = document.querySelector('button');
const output = document.querySelector('.output');
document.addEventListener('DOMContentLoaded',init);
btn.addEventListener('click',btnClicker);
function btnClicker(){
  const val = myInput.value;
  if(val.length > 0){
      let user = JSON.parse(localStorage.getItem('user'));
      user.first = val;
      console.log(user);
      localStorage.setItem('user',JSON.stringify(user));
  }
}
function init(){
  console.log('ready');
  output.innerHTML = '';
  let val = localStorage.getItem('user');
  let user = JSON.parse(val);
  if(!user.first){
      let storeObj = JSON.stringify(myObj);
      localStorage.setItem('user',storeObj);
```

```
  }else{
     output.textContent = `${user.first} ${user.last}`;
  }
  console.log(val);
}
```

Math object

```
const arr = [];
const arr1 = [];

for(let i=0;i<10;i++){
   const val = ran(0,1000);
   arr.push(val.toString());
}

for(let i=0;i<20;i++){
   const ind = Math.floor(Math.random()*arr.length);
   const val = arr[ind];
   console.log(ind,val);
}

function ran(min,max){
   return Math.floor(Math.random()*(max-min+1))+min;
}

for(let i=1;arr1.push(i++)<50;);
```

```
console.log(arr1);

const arr2 = arr.sort();
const arr3 = arr.reverse();
arr.sort(()=>{
  return Math.random() - 0.5;
});
console.log(arr[0]);
arr.sort(()=>{
  return Math.random() - 0.5;
});
console.log(arr[0]);
```

Object constructor Object Construction with JavaScript

```
function FullName(firstName,lastName){
 this.firstName = firstName;
 this.lastName = lastName;
 this.full = `${firstName} ${lastName}`;
}
const person1 = new FullName("Laurence","Svekis");
console.log(person1.full);
const person2 = new FullName("John","Svekis");
```

```
console.log(person2.full);
const person3 = new FullName("Mike","Smith");
console.log(person3.full);
let val = `${person1.firstName} ${person1.lastName}`;
console.log(val);
```

String Whitespace Cleaner and Remover

```
String.prototype.cleaner = function(){
  return this.replace(/\s+/g,' ').trim();
}
const myStr1 = "  Hello    World   ";
console.log(myStr1.cleaner());
let val = myStr1;
val = trimMyString(val);
console.log(val);
document.querySelector('.output').textContent = val;
function trimMyString(str){
  let val = str.replace(/\s+/g,' ').trim();
  //val = str.trimLeft();
  //val = str.trimRight();
  return val;
}
```

Random Numbers and Random Array items

```
const arr = [];
const arr1 = [];
for(let i=0;i<10;i++){
  const val = ran(0,1000);
  arr.push(val.toString());
}
for(let i=0;i<20;i++){
  const ind = Math.floor(Math.random()*arr.length);
  const val = arr[ind];
  console.log(ind,val);
}
function ran(min,max){
  return Math.floor(Math.random()*(max-min+1))+min;
}
for(let i=1;arr1.push(i++)<50;);
console.log(arr1);
const arr2 = arr.sort();
const arr3 = arr.reverse();
arr.sort(()=>{
  return Math.random() - 0.5;
});
console.log(arr[0]);
arr.sort(()=>{
  return Math.random() - 0.5;
});
```

```
console.log(arr[0]);
```