**Javascript esc6**

**//escopo global**

**{**

**//escopo de bloco**

**}**

**Function test() {**

**//escopo de função**

**}**

**// Geratators**

**function\* hello() {**

**console.log('Hello');**

**yield 1;**

**console.log('from');**

**var value = yield 2;**

**console.log(value);**

**}**

**const it = hello()**

**console.log(it.next());**

**console.log(it.next());**

**console.log(it.next('outside'));**

**//generetor**

**function\* naturalNumbers() {**

**let number = 0;**

**while (true) {**

**yield number;**

**number++;**

**}**

**}**

**const it = naturalNumbers();**

**console.log(it.next());**

**console.log(it.next());**

**console.log(it.next());**

**console.log(it.next());**

**console.log(it.next());**

**const iniqueId = Symbol('Hello');**

**// Well Known symbols**

**Symbol.iterator;**

**const arr = [1, 2, 3, 4];**

**const str = 'Digital Innovation One';**

**//generator**

**const obj = {**

**values: [1, 2, 3, 4],**

**\*[Symbol.iterator]() {**

**for (var i = 0; i < this.values.length; i++) {**

**yield this.values[i];**

**}**

**}**

**};**

**for (let value of obj) {**

**console.log(value);**

**}**

**Função com pausa**

**// funçoes de callback**

**function doSomething(callback) {**

**setTimeout(function () {**

**//did something**

**callback('fish data');**

**}, 1000);**

**}**

**function doOtherThing(callback) {**

**setTimeout(function () {**

**// did other thing**

**callback('second data');**

**}, 1000);**

**}**

**function doAll() {**

**try {**

**doSomething(function (data) {**

**var processoData = data.split('');**

**try {**

**doOtherThing(function (data2) {**

**var processoData2 = data2.split('');**

**try {**

**setTimeout(function () {**

**console.log(processoData, processoData2);**

**}, 1000);**

**} catch (err) {**

**//haldle error**

**}**

**});**

**} catch (err) {**

**// handle error**

**}**

**});**

**} catch (err) {**

**// handle error**

**}**

**}**

**doAll();**

**// promises**

**const doSomethingPromise = () =>**

**new Promise((resolved, reject) => {**

**setTimeout(function () {**

**//did something**

**resolved('First data');**

**}, 1000);**

**});**

**const doOtherThingPromise = () =>**

**new Promise((resolved, reject) => {**

**setTimeout(function () {**

**// did other thing**

**resolved('second data');**

**}, 1000);**

**});**

**doSomethingPromise()**

**.then(data => {**

**console.log(data.split(''));**

**return doOtherThingPromise();**

**})**

**.then(data2 => console.log(data2.split('')));**

**//Pending**

**console.log(doOtherThingPromise());**

**// fufilled**

**//reject**

**// promises**

**const doSomethingPromise = () =>**

**new Promise((resolved, reject) => {**

**setTimeout(function () {**

**//did something**

**resolved('First data');**

**}, 1500);**

**});**

**const doOtherThingPromise = () =>**

**new Promise((resolved, reject) => {**

**setTimeout(function () {**

**// did other thing**

**resolved('second data');**

**}, 1000);**

**});**

**Promise.race([doSomethingPromise(), doOtherThingPromise()]).then(data => {**

**console.log(data);**

**});**