var promise = new Promise(function(resolve, reject) {

 // do a thing, possibly async, then…

 if (1===1) {

 resolve("Stuff worked!");

 }

 else {

 reject(Error("It broke"));

 }

});

promise.then(r => console.log(r));

=> Stuff worked!

var promise = new Promise(function(resolve, reject) {

 resolve(1);

});

promise.then(function(val) {

 console.log(val); // 1

 return val + 2;

}).then(function(val) {

 console.log(val); // 3

})

1

3

function doubleAfter2Seconds(x) {

 return new Promise(resolve => {

 setTimeout(() => {

 resolve(x \* 2);

 }, 2000);

 });

}

doubleAfter2Seconds(10).then((r) => {

 console.log(r);

});

=> 20

function addPromise(x){

 return new Promise(resolve => {

 doubleAfter2Seconds(10).then((a) => {

 doubleAfter2Seconds(20).then((b) => {

 doubleAfter2Seconds(30).then((c) => {

 resolve(x + a + b + c);

 })

 })

 })

 });

}

addPromise(10).then((sum) => {

 console.log(sum);

});

=> 130

async function addAsync(x) {

 const a = await doubleAfter2Seconds(10);

 const b = await doubleAfter2Seconds(20);

 const c = await doubleAfter2Seconds(30);

 return x + a + b + c;

}

addAsync(10).then((sum) => {

 console.log(sum);

});

=> 130