Functions(Fonksiyonlar)

Bir Javascript fonksiyonu üç farklı yöntemle tanımlanabilmektedir.

Example

//1

function function1(){

}

//2

var function2 = function(){

}

//3.Immediately Invoked Function Expression(IIFE) : Tanımlandığı yerde çalışır.

(function(){

})()

* İç içe fonksiyon tanımlanabilir.
* Fonksiyonlar return deyimi ile geriye değer gönderebilir.

Example

function addNumbers(x, y){

return x + y;

}

function cube(x){

return x \* x \* x;

}

console.log(cube(addNumbers(1, 2))); //27

* Parametresiz fonksiyonlar parametreli olarak çağrılabilmektedir.

Example

function writeNumbers(){

for(index in arguments){

console.log(arguments[index]);

}

console.log(arguments.length); //3

}

//Parametreler "arguments" dizisine aktarılır. Bu diziye fonksiyon içerisinden

//erişebilirsiniz.

writeNumbers(1, 2, 3);

* Fonksiyon parametrelerine default değerler verebilirsiniz.

Example

function addNumbers(x = 1, y = 2){

console.log(x + y);

}

addNumbers(); //3

addNumbers(10); //12

addNumbers(10, 20); //30

* Bir fonksiyon parantezsiz çağrılırsa fonksiyonun kendisi döner.

Example

function function1(){

console.log("buraya birşeyler yaz.");

}

var function2 = function1;

console.log(function2); //ƒ function1(){console.log("buraya birşeyler yaz.");}

function2(); //buraya birşeyler yaz.

call(), apply() ve bind() Fonksiyonları

Example

var obj1 = {number1: 10, number2: 20}

var obj2 = {number1: 30, number2: 40}

//this anahtar kelimesi normalde window objesini gösterir. call sayesinde artık

//obj1 ve obj2'yi gösterecek.

function addNumbers(number3, number4){

console.log(this.number1 + this.number2 + number3 + number4);

}

addNumbers.call(obj1, 100, 200); //330

addNumbers.call(obj2, 100, 200); //370

//apply'nin call'dan farkı parametreleri dizi değişken olarak alması.

addNumbers.apply(obj1, [100, 200]); //330

addNumbers.apply(obj2, [100, 200]); //370

//bind ise fonksiyonun bir kopyasını oluşturup parametre olarak verilen obje ile

//kullanılmasını sağlar.

var copyFunction1 = addNumbers.bind(obj1);

var copyFunction2 = addNumbers.bind(obj2);

copyFunction1(100, 200); //330

copyFunction2(100, 200); //370

Global Functions

String()

Parametre olarak verilen değeri string türüne dönüştürüp geri gönderir.

parseInt()

Parametre olarak verilen değeri number türüne dönüştürüp geri gönderir.

* Tamsayılar için kullanılır. Virgüllü sayılarda virgülden sonrası atılır.

parseFloat(), Number()

Bu fonksiyonlar parseInt()’den farklı olarak virgüllü sayıları da dönüştürebilmektedirler.

isNaN()

Parametre olarak verilen değer sayı ise false, sayı değilse true döner.

Example

console.log(isNaN(1)); //false

console.log(isNaN("1")); //false

console.log(isNaN(NaN)); //true

console.log(isNaN(Infinity)) //false

eval()

Parametre olarak verilen string ifadeyi javascript koduymuş gibi çalıştırır.

Example

console.log(eval("2 + 2")); //4

const number = 1;

console.log(eval("number + 1")); //2

isFinite()

Bir sayının sonlu ve yasal bir sayı olup olmadığını kontrol etmek için kullanılır. Parametre olarak verilen sayı NaN, Infinity veya -Infinity ise false aksi halde true döner.

Example

console.log(isFinite(1)); //true

console.log(isFinite("1")); //true

console.log(isFinite(NaN)); //false

console.log(isFinite(Infinity)); //false

console.log(isFinite(-Infinity)); //false