Functions-Tryout

Functions

Problem Statement

The given code shows how we can create the functions and how the function can be called

Code in Plain JavaScript

function myFunction(num1,num2){

num3=num1\*num2;

return num3

}

mulVal=myFunction(10,20);

console.log(mulVal); //200

# Functions as Objects-Tryout

Functions as Parameter

Problem Statement

The given code demonstrates the code where function is passed  as a parameter

// function passed as a paramter

function welcome(){console.log("Hello World");}

function goodbye(){console.log("See you later");}

function greet(choice){

choice();

}

greet(welcome);

greet(goodbye);

Arrow Functions -Tryout 1

Arrow Functions

Problem Statement

The given code shows the usage of different scenario of arrow functions. There are follwoing  types of syntax:

* Multi parameter, multi line code:
* No parameter, single line code:

// Multi parameter, multi line code

calculateCost = (ticketPrice, noOfPerson)=>{

noOfPerson= ticketPrice + noOfPerson;

return noOfPerson;

}

console.log(calculateCost(500, 2)); //1000

//No parameter,single line code

trip = () => "Lets go to trip."

console.log(trip()); //// Lets go to trip.

Arrow Functions-Tryout 2

Arrow Functions

Problem Statement

The given code demonstrates the two syntax of the arrow functions, which are given below:

* One parameter, single line code
* One parameter, single line code:

//One parameter, single line code

trip = place => "Trip to " + place;

console.log(trip("Paris")); // Trip to Paris

//One parameter, single line code:

trip = \_ => "Trip to " + \_;

console.log(trip("Paris")); // Trip to Paris

Anonymous function-Tryout

Anonymous Functions

Problem Statement

The given code shows the creation  of the function without the name

function greet(choice){

choice();

}

greet(function(){ console.log("Hello World")}); // Hello World