

HPre function y is taking

another function x as an

Episade 18 Higher-order functions and functional programming

What is Higher order Functions 9

A function which takes another function as an argument or returns a function from it is known as Higher order function.

example %-

Function X() {

(onsole.log("Namaste");

function y(x) {

orgument, thesefore yis higher order function. (allback function

Introduction to functional programming example 3-

Const tadius = [3,2, const radius = [3,1,2,4];

ronst ralculate Arrea = function (radius) f Const output = [];

for (let i=0; i < radius.length; it+) {
Output.push(Math.PI * radius[i] * radius[i];
}

Page No.
Date
Tetuon output;
3;
(onsole.log((alculateArea(radius));
ronst calculate (is cumprence = function (radius) {
(onst output = []; for (let i=0; i z radius.length; i++)? Output.push (2* Math.PI * radius[i]);
too (let i=0; 12 sadius length; 1++)2
output.push (2* Math.PI * radius (1))
3
return output;
3;
(onsole: log (calculate (iscumfesence (radius));
 J
ronst calculate Diameter = function (radius) 2
const output = [];
for (let i=0; i < radius.length; itt)?
output.push (2* radius [i]);
3
return output;
7:
(onsole.log ((alculate Diameters (rudius));
1 mosteriog (Lated a confine de la lacon many) y
1 output
► (4) [28.2743,3.1415,12.5663,50.2654]
► (4) [
► (4) [6,2,4,8]
- (4) L 0/2/4/03
[this works good, but this is not good way]
1 1/15 WIES 4000 PAT 11/15 1 1/21

	Page No.
-	
	this code is not modulas, not reusable.
r	Danat Yourself
*	DRY Principle - Don't Repeat Yourself
*	Optimized (ode (interview recommended)
.,,	
	for doing this, we will make use of functional
	plogramming
	J
-	const $\gamma a dius = [3,1,2,4]$;
-	return Math.PI * yadius * * * adius;
	3.
	(onst (incumforance = function (radius) {
	return 2* Math.PI * adius;
	3 ;
	const diameter = function (radius) {
	return 2* sadius;
	<i>y</i> ;
	[onst calculate = function (sadius, logic)]
	fox (lahi = D: i cradius langua to i) s
	for (let i=0; i < radius.length; i++) { Output.push (logic (radius [i]));
	3
	betuen output;
	J;
	(on sole. log (calculate (radius, grea));
	(onsole.log((alculate (radius, (iscumfrom(e)); lonsole.log((alculate(radius, diametrs));
	J. wonsole. 10 g ((ulculate (sawus, wometers));
	same output as before

Pa	ge N	o.		
Dat	е			

* Beauty of functional programming

functional Programming is huge in itself but a small past of it says that think or make logic in your head according to functions.

* pollyfil for map function in javascript

- map is higher order function.

Const radius = [3,1,2,4]; (onst grea = function (radius) {

return Math.PI * padius * radius;

Array. prototype. (alculate = function (logic) £

const output = [];

for (let i=0; i < this.length; i+t) }

output. push (logic (this[i]));

r

return output;

(on sole · log (radius · map (grea)); > some output

Higher order functions and functional programming is only possible because, functions are ##