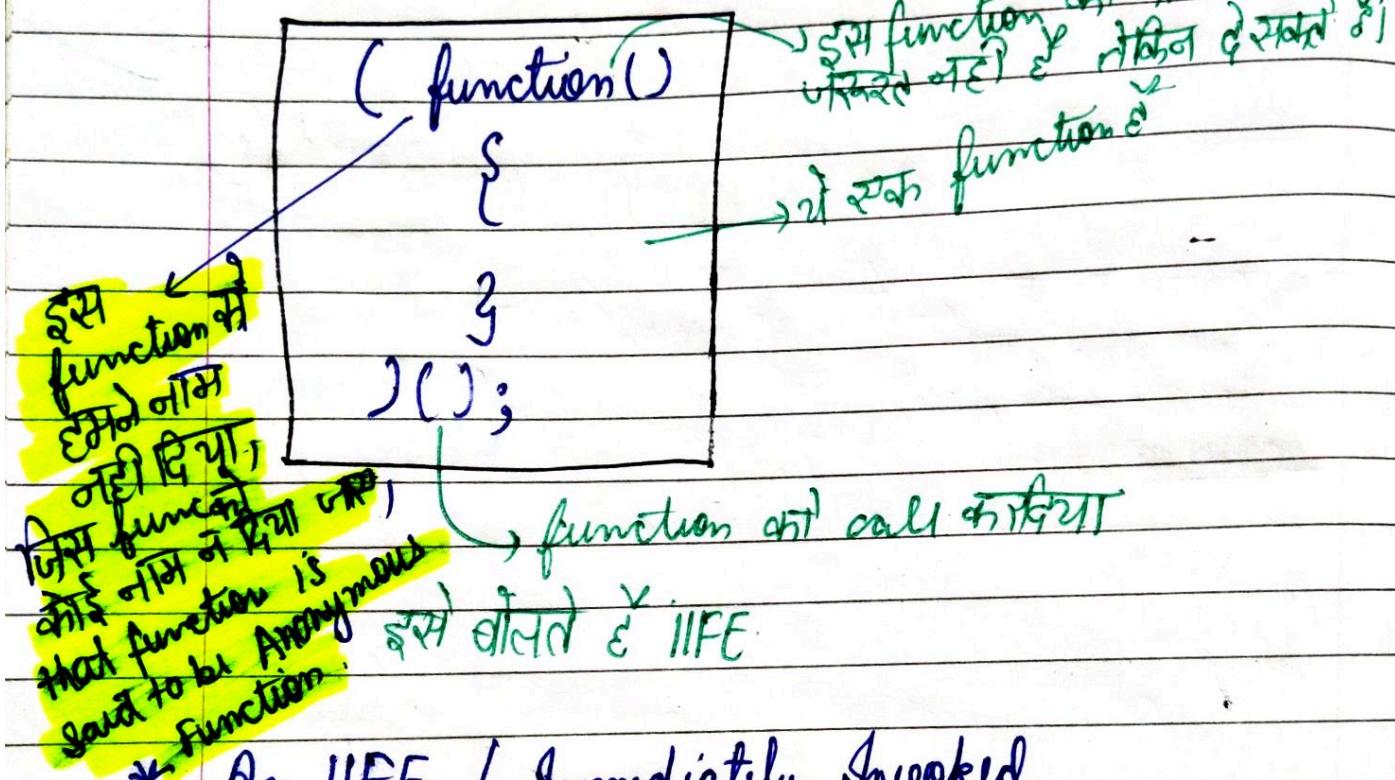


IIFE

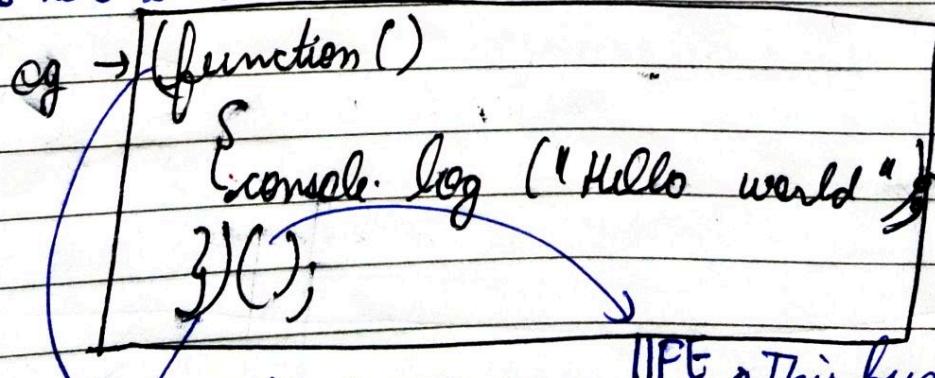
↳ Immediately
Invoked
Function
Expression

IIFE का कार्य function होता है; जिसे बनाते ही
उस call करते हैं।



* An IIFE / Immediately Invoked Function Expression)

is a JavaScript Function that runs as soon as it is defined. It has no name & is not stored in a variable.



normal function is giving a name so that we can call the function

With IIFE().

Page No.	
Date	

function multiply(a, b)

{

return a * b;

};

console.log(multiply(8, 24));

(function(a, b)

{ return a * b;

) (8, 24);

These end
parenthesis
are actually
what calls
the
function

You must
know that
this function
can never
be called
again.

Here is an
example of a
normal
function:

We declare a
function and
then later
recall the function.
Pass the values and
it returns the product
of those values.

Here is the
same function
in IIFE (Immediately
Invoked
Function
Expression)

Here, we

1) remove the
function name,

2) wrap the function
in parenthesis,

3) and then add a
parenthesis to
the end with the
values we want
to pass

तो अब हम Function calling के 5 ways का स्तर

- 1) normal call
- 2) call () function
- 3) apply () function
- 4) Bind () function.
- 5) IIFE

↳ IIFE से function को भी विद्युत में अप्लाई कर सकते हैं।

Eg. of IIFE

↳ एक HTML page बनाओ → पृष्ठा का webpage open
 (page load होता है), ask the user for
 his name & then print
 filename : basic.html hello username

```

<html>
  <head>
    <title> for IIFE Example </title>
  </head>
  <body>
    <script src = "IIFE.js"></script>
  </body>
</html>
  
```

↓
This is the
source
file

Filename → life.js

```
(function () {
    let uname = prompt("What's your name?");
    alert("Hello " + uname);
})();
```

↑
uname
↓
start
username

सुनिए दी
page load होता

visit webpage.

Open it, then

1) there will be an input box prompted on the screen.

2) we have to give the username as input

3) & then print Hello username.

webpage को
inspect करता
अपके console
में हमें output
मिलता

whatsoever.
username we
give, will be
printed.

Eg 2) of IIIFE → एक web page (basic html page)
बनाओ, पैसे ही web page
open होता, तभी ask the user for the timer

Page No.	
Date	

जिस तरीफ
seconds का
timer चलाना है?

→ let's say user of $\frac{10}{\text{seconds}}$ submit किया
→ जो यह 10 seconds
का timer चलाना है?

→ After every second, print फ़ख़्री की
timer i.e. (seconds तक रह दें).:

for eg. 10
10 तक 1 तक
counting
print
करो

9
8
7
6
5
4
3
2
1

After the counting from 10 to 1 is
done / completed

→ एक alert खोलें
"Counted 10 seconds"

अब ये Example के लिए हमें IIFE,
prompt,
Alert,
Set Interval.
की knowledge होनी
चाही।

Page No.	
Date	

function()

let timeUnits = prompt ("How much to count");

After time

(After seconds)

After counting ~~timer~~ E^l

(For timer E^l).

let interval = prompt ("Log after how much interval");

setInterval

if it
function
pass E^l

at E^l
interval
happened

function
call

All this
is handled
by setInterval
function

After ~~E^l~~ the print ~~E^l~~ function passed E^l

setInterval (handleCall,
interval * 1000

milliseconds
per second of convert
E^l to E^l

function handleCall()

{ console.log (timeUnits + " left ");

timeUnits -= interval;

if (timeUnits == 0)

setInterval
E^l

timeseconds
at setInterval
subtract
E^l

{ clearInterval (uid);

· alert (" Done ");

charInterval
interval
uid pass
call E^l
function
call E^l

3TR
timer
get E^l E^l
(start time units
3TR 0 E^l)

3()3

Output → वेब पैज़ ही html page load से तो, तभी console में timeUnits left print होगा।

10 left
9 left
8 left
7 left
6 left
5 left
4 left
3 left
2 left
1 left

This page says:
Done

OK

Ans 2.

Always
function
object

WAY2

calls the handleCalls
function
after every interval

Functional object example
function →
3rd EIT
properties की रख
2nd EIT functions
can be used as
a store of
properties (much
like objects)

Original
Time Unit की
Time करती रही
Save कर रही थी
(10 की रखी रही)
check भी

अब original Time unit
की print करती रही

function () {

baseInt() प्रोटोटोट फंक्शन
to convert int from string

→ let timeUnits = prompt("How much do you want?");

→ let interval = prompt("Log after how much interval?");

→ let iid = setInterval('handleCalls', interval * 1000);

handleCalls: OrigTu = timeUnits → return
an id

function handleCalls () →
function के अपर एक static property के रूप में रखा गया है

function handleCalls () →
if (timeUnits >= 0) →
{ console.log(timeUnits + " left"); →
timeUnits -= interval; →
if (timeUnits == 0) →
{ clearInterval(iid); →
alert(handleCalls.OrigTu + "

" has been counted."); →

Eg of Closure.

Page No.

Date

```
function powerCreator(exp)
```

```
{ let fun = function(base)
```

```
{ let res = Math.pow(base, exp)  
return res;
```

```
}
```

```
return fun;
```

```
}
```

```
let square = powerCreator(2);
```

```
let val = square(8);
```

```
console.log(val);
```

Change powerCreator function to make it a producer of such functions whose exponent we can change later on.

Solution → ^{powerCreator function को} ^{^ एक exponent को} ^{Simply base करेंगे}
^{एक exp को pass करेंगे एक object की, उसकी}
^{exponent property करेंगे जिसका base करेंगे pow() function}
^{और उस object की property की value}
^{गो set है, उसी तो ही एक वाले में भी change}
^{hो सकते हैं।}

object pass first
as a argument

Page No.

Date

function powerCreator(obj)

{ let fun = function (base)

{ let res = Math.pow(base, obj.exp);
return res;

}

return fun;

}

let o1 = {

exp: 2,

}

let csquare = powerCreator(o1);

let val = square(8);

console.log(val);

→ 8 is the base.

change the value of exp property of object 1
from 2 to 3.

o1.exp = 3;

let val = square(7);

console.log(val2);