Date: / /
Closures in JavaScript
function say Word (word) &
24 retwen () => console.log(word);
function say Word (word) & recture () => console log (word); y const cayHello = Say Word ("hello");
SayHello(); 11"hello"
THE PERSON OF MANY AND AND THE PERSON
There's 2 interesting point to notice:
- The returned function from sayword
The returned function from sayword can access the word parameter
The returned function maintain the value of word when SayHello is called outside scope of word.
value of word when SayHello
is called outside scope of
word.
The Livet boint can be explained
The first point can be explained by Lexical Scope.
- Lunction
lexical Scope - The returned thexists
lexical Scope - The returned functions can access word before it exists in the order scope

The second point becor of Closwies A closure is a function combined with references to variables define outside of it. Clasure maintain the variable references, which allow function to access variables outside of their scope. They "enclose" the function and variable in its environment Example of Closures In JavaScript Callbacks - It is common for a callback to reference a variable declared outside of itself eg -> Function get Care By Make (make) & return course, filter (x => x, make = = make) make is available in caleback because of lexical Scoping and make is persisted when filter called becor of closure

Date:/_/_
Storing state > we can use closures
Storing state > Me can use closiones from functions that store states
Let's say a In which retwens an
Let's say a for which retwens an object that can Store and change name.
for function make Person (name) of let -name = name;
let -name = name!
The state of the s
retwen Example of the resiliant
set Name: (new Name) => (_name = new Normanne)
get Name: () \Rightarrow Name,
3
Const me = make Person ("deepa");
Consolitog (me. get Name (1); 11"dep."
me. Set Name !" Deepa Chauravia" 1.
me. Set Name !" Deepa Chaurasia"); console. log (me. get Name ()); 11" Deepa Chaurasia");
11 " Deepa Chaurasio"?
The total spirite in the second
It shows how closure do not freeze
values of variables from function's
outer scope during creation.
Insted they mountain the seferences throughout closwers lifetime
throughout closures libetime

	Date:/_
Private methods	Telephology and the second
O MAN MANUEL MANUEL	
So In cops concept class we have pand expase getter methods public.	, wo In a
class we have	private state
and expose getter	and setter
methods public.	
We can extend this	00/00
Function make Person	name) ?
function make Person (let_name = nan	n'e I man I to be
OLOSVI CE ()	V KUTO VI IND
Function Private Set	Name (newName) of
- name = neu	
- 1 4 poles consol st	nor = see days
and the (almost to be to	ar reportable report
seturn à	
set Name: (new Na	me) => Private Set Name
Andrew Company	(new Name)
get Name: () =>	_name,
99	
3	
	Harry Market
	and the state

			Date: / /	
Porivat	eSet Name esible to	is not a	directly	
acce it	can acce	consumer by	s and	
sta	te vaeviable	2 - name	nivate through cl	084
Closw	us make	it possib	le for '.	
THE RESERVE AND PARTY AND PERSONS ASSESSED.				0
with	side con	aviables	connection g even variables	_
6100		y of the	variasies	
flike	Linkedl	n maybe	2 .)	_
There	ave mas	my nees	of closures	10
of the	it store	state a	like structure	3
Ir	nblement	private n	nethods	
$-\tau$	passing handler	c allback	70 event	
	1.			