Block, scope and shadowing in javascript		
min min min min de		
block is also called as compound statement:		
-block is defined by curly braces. { }		
we group multiple statements together in a block		
so that we can use it where javascript expects		
one statement.		
if (true) of		
var a=10; block (=) 10		
(onsoll·log(a);		
3		
fonst are block scoped		
֡֡֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜		

	Page No. Date
	- black scope means what an variables and functions
	we can access inside block.
-	Chapal scoped
	VOT a = 10;
	let b = 20; > block scoped
	[0][3t c = 30]
	$\begin{array}{ccc} (\text{on sole.log (a)}; & & & & \\ (\text{on sole.log (b)}; & & & \\ & & & & \\ & & & & \\ \end{array}$
	$(on sole, log(b); \longrightarrow 20$
	(on sole log (c);
	3
	(onsole log (a); gefrence exxx: b is not
	(onsole, log(b); defined
	(on sole log (1);
-	- Var a=100;
	shadowing shadowing
	VOB (1 = 10)
	let b = 20; => 10 Hore vax a = 10 shadows
	(onst 0 = 30°, Var a = 100
	(onsole log(a);
	3
	d = 100;
	$vor a = 10; \qquad \Longrightarrow 10$
	let b = 20;
	(onsole loy (a);

Shadou	Page No.
•	a cope globa galo
	has it's set
	let b=100; This b' has its block scope 1 has its block scope
	1 von a=10; This b has its block scope (in our case local)
1,	von a=10; This b (in our case local)
This'b' is	← le+ b = 20; ←
shauding	[Opst c = 30;
110 pp 165	(on sole. log (b); => 20
Jalla Jalla	100
	(on sole: loy (b);
	The state of the s
	=7 i.e. let b = 20; is shadwing let b=100;
	man a man of an army
	· · · · · · · · · · · · · · · · · · ·
	const c=100;
	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
	(onst c=10; =>) 10 const c=1
	(01) - ((10)(())
	3
	(on sole log(c);
3	
	=> 'let' and 'const' same things applied
	Vin our court our fin william
	(onst (= 100;
	function x() {
	(onsole.log(c); (=>) 10 (example)
	3 100
	(onsolelog(c); (onst c=10) shadows (onst c=10)
	(onsole.log(1);
	10102 (21)

we can shadow let using let 7 we cannot shadow let using yar we can shadow var using var we can shadow var using let Page No. Date Hegal shadowing - let a = 20; vos a = 20; > syntaxerror identifies 'a' has already been declored alid shadowing let a = 10; VOS a=10; > valid shadowing " UON a = 20; van a=10; alid shadowing let a = 20°, - let a=20; This is valid because von a=10; function x() { has its boundary or block scope VOY 0=10; const a = 10; > valid (onst a=20%

T blo	ck scope also follows lex	ical scope]
[lex	in Liscope works in same	Date
	inside block a	SO
	(00st 20 = 00)	
	Const a=20;	
	(onst a = 100;	[lasteal support
	1	[lexical scope]
	(ons+ a = 200;	(=)100
	Z	
	$logsole.log(a); \leftarrow \Gamma.$	this 'a' will get access from its nemest 'a'
	3	from its nemest 'a'
		· · · · · · · · · · · · · · · · · · ·
-	- lonst a=20;	
		in an and
	(onst a=100;	[lexical scope]
	1 '	
	(onst a=200;	(=)200
	1 on sole log (a); 2	[this 'a' will get access from its nearest 'at
	4	from its neasest at
	Janes Chillip Land	
	-(onst a=20);	
	(1
-	(anst a = 100;	>100 [lexical scope]
	e ye of borners	(=>)100 [[(A)
1 7	(on sole log (a);	
	3	[this a will get access] from its nearest a!
	3	Trom Ho Wholes a
	$\frac{\text{const} \alpha = 20;}{4}$	[lexical scope]
		[[PXI (WI ST]]
	(onst 0=100; 1	20 and access
	(onst a=200;	this a will get access from its nearest at
	3, , , , , , ,	from its neares
	console. log(a);	

Page No. 1 Hormal Function scope rules which work on functions once Hy same on moon function also.