管理No.	FND27	Day9	Page1							
実施内容	オブジェクト入門									
場所	542									
実施日	2024/5/10	実施時間	9:00	~ 12:00						
実施者	DJ さん	受講者	0.00	1		i				
	Ai さん			·+ !	 	-} 				
 作成者	本田	!	作成日(開始)	2024/5/14	作成日(完了)	2024/5/14				
<time></time>	<contents></contents>		ТРЖЦ (МЖ)	202 1/ 3/ 1 1	TENNIA (201)	202 1/ 3/ 1 1				
(Tillie)		(object)とは								
	★オブジェクト (object)とは [.js] const 変数名 = {									
	[.js] const 変数名 = { +-: 値,									
	·									
	}; ・どのようなデータ型の値でも格納できる									
	- ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・									
	・「key-valueペア」(キーと値のペア)の順番は保証されない [.js] 例:オブジェクト									
	[.]S] 19月: オノンエクト const info = {									
	const into = { name : "Hana",									
	dog : true,									
	age : 12, } ;									
	1 .									
	★オブジェクト参照(ブラケット記法)									
	★オノシエクト参照(ノラグット記法) [.js] 変数名["キー"]									
	・オノンエクトの中の個を参照するとさば「キー」を指定することで個を参照できる									
	・「・・」に文字列で書く 「・is】例:ブラケット記法で参照									
	[.]S] 例: ノフグット記法で参照 console. log(info["name"]); // ⇒ Hana									
	console. log(info ["dog"]);									
	console. log(info ["age"]); // \Rightarrow 12									
	★オブジェクト参照(ドット記法)									
	「.js」 変数名.キー									
	[.Js]									
		の中の値で多点。)のまま書いてok	(9CG(9) + 1G1F)		># CG 9					
		ブラケット記法で参	·····································							
		g(info. name)				// ⇒ Hana				
	console. log(info. dog); // ⇒ tr									
	console. $\log(\inf \log)$; // \Rightarrow 12									
	// → 12									
	★オブジェクトの値を変更・追加									
	★ オノシェクトの値を変更・追加 [.js] 変数名 ["+-"] = "変更後の値";									
	[.]S] 多数名[4-]= 多更後の値 ; ・追加と変更の方法は同じ									
	☆キーが既にあれば値が更新され、キーが元から存在しなければ「key-valueペア」が追加される									
	「.js] 例:オブジェクト									
	const lesson = { //オブジェクトの宣言									
	instructor: "Eriko",									
	coding : true,									
	} ;									
	-	tructor = "Tmg	aroh".		//値の効	重(ドット記注)				
	lesson. instructor = "Tmaroh";									
	lesson. school = "DIG"; //値の追加(アッケット記法)									
	console. log		// → Sinctructor	"Tamaroh" and	ng:true, place:"Zoo					
	COHSUTE. 108	5 (1600011),	// → {IIISTructor	. Tarriaron , codii	ig.true, place: 200	iii , School: DIG }				

	加記球帳				¬						
管理No.	FND27	Day9	Page2		4						
実施内容	オブジェクト入門	り、人コーフ - 調整・工程									
場所		!									
実施日	2024/5/10	実施時間	9:00	<u>~ 12:00</u>		•					
実施者	DJ さん	受講者		¦ ∤	¦ 	¦ -}					
	Ai さん										
作成者	本田	<u> </u>	作成日(開始)	2024/5/14	作成日(完了)	2024/5/14					
<time></time>	★スコープと	<contents> ★スコープとは ・ある区切られた範囲を指し、スコープの中で宣言された変数はその中でしかアクセスできない</contents>									
		家の外									
	(ロー あた	あなたの家 (ローカルスコープ) あなたの部屋 (ローカルスコープ)									
		☆あなたには <mark>自分の部屋の中</mark> にあるラグビーボールが見えますが、それはあなたしか使えません。 ☆あなたには <mark>自分の家の中</mark> にあるサッカーボールも見えますし、それは家族で使えます。									
		☆家の外にあるバレーボールも見えますが、それは地域の皆さんで使えます。									
		☆でも他の人の家にあるシャトルや、部屋にあるボーリングの球は見えませんし、使えません。									
		☆ 他の人も <mark>あなたの家の中</mark> にあるサッカーボールや、 <mark>部屋にある</mark> ラグビーボールは見えませんし、									
		☆他の人も <mark>のなたの家の中</mark> にめるケッカーホールや、耐量にあるプクヒーホールは兄えませんし、 使えません。									
	** *	・ボールは「変数」を表しており、使える範囲を「スコープ」という									
		・ホールは「変数」を表しており、使える範囲を「スコーノ」という ・すべての変数が、コードの全ての場所で使えるわけではない									
	- 97((0)	・9/11にの変数が、コートの主にの場別で使えるわりではない									
	→・眼巻・レフェ	★関数とスコープ									
	let a = 5;	J-/									
			 ↓これがスコープ								
	let $b = 3$;	ما ما ۷ (۱۰۰۰ ۱۰۰۰ ۱۰۰۰ ۱۰۰۰ ۱۰۰۰ ۱۰۰۰ ۱۰۰۰			•						
		ddX(num) {			let num = 2;						
	let x =5				x = 5;						
		um + x;			return num +	- x;					
	` ' '	addX(2); //関数呼出し ⇒ JavaScriptがコンピュータの中に呼出す区画を作る									
	☆関数を定 新しいス	・関数の処理が終了すると、スコープもその中の変数もコンピュータ上から消える ☆ 関数を定義する ということは 処理をまとめる というだけではなく、 変数が有効な範囲を決める 新しいスコープを作っていると言えます(抜粋: JavaScript Primer)									
	★スコープの	★スコープの内と外									
	let $a = 5$;										
	let $b = 3$;			スコー	スコープの内からは変数xにアクセス可能						
	function a	ddX(num) {		let num = 2;							
	let x = 5			x = 5;							
		um + x;	return num + x;								
	}	·									
		addX(2); //関数呼出し ⇒ JavaScriptがコンピュータの中に呼出す区画を作る									
	console.lo	a(x):		•	arenceError: x i						
	33113010110	5(*)/	11		O外からは変数×に						
				N_	ンバルンの父女人(・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・					

■ 秋月天』		D= 0		22			¬				
管理No.	FND27	Day9	-	Page3			4				
実施内容	オブジェクト入門	スコープ : 386-73									
場所							_				
実施日	2024/5/10	実施時間		9:00	~	12:00					
実施者	DJ さん	受講者			_i		_ <u>j</u>	į 			
	Ai さん		[]				
作成者	本田		1	作成日(開始)	202	4/5/14	作成日(完了)	2024/5/14			
<time></time>	<contents></contents>										
		★関数スコープ									
	let a = 5;										
	let $b = 3$; \downarrow これが関数スコープ										
	function addX(hum) { let num = 2;										
	let x = 5; x = 5;										
	return nu	m + v:					return num +	. V'			
	}	штх,					^	· ,			
				//88米40式(1)	\ 1 =	- C: +-h"-	 				
	addX(2);		/	/関数呼出し =	⇒ Java	aScript///_	コンピュータの中に呼	出り区凹で作る			
	console.log	(x);		/,			arenceError: x i				
)外からは変数 x に	アクセス出来ない			
				子で呼び出され							
	・関数内で宣	言された「変数	汝・仮弓	数」は、 関数ス	ユープア	勺でのみア	クセス可能				
	☆仮引数は、	関数スコーフ	内で宣	言され、引数に	よって初	刀期化され	る				
	☆仮引数は、	関数に引数を	を取り込	むのに使用され	ເ ວロ− :	カル変数(物	持定のスコープ内だ	けの変数)			
	★ブロックスコ	コープ									
	・ {} で囲んだ範囲をブロックと呼び、ブロックもスコープを作成し、それをブロックスコープと呼ぶ										
	・ブロック内で気	E義した変数	は、スコ	ープ内でのみ参	照可能						
	{										
	const x =	1:									
	console.lo							// ⇒ 1			
	}	79(77)						77 - 2			
	console.ld	u(x).				// ⇒ Ref	arenceError: x	is not defined			
	CONSOICHE	79(^);					プの外から x を参照				
						\	ノのハガラスで多点	(CE/80/1/0)IJ			
	★ブロックスコープを使うもの ·if文やwhile文										
	let num = 2	•									
	if (num > 0										
		= "正の数だ									
	console.log(resalt); //スコープの内側 // ⇒ "正の数だ										
	}										
	console.ld	g(resalt);	//スコ-	-プの外側			arenceError: x				
						↑スコー	プの外からxを参照	できないためエラー			
	★for文のブロックスコープ										
	・ループごとに新しいブロックスコープを作成し、処理が完了すると破棄される										
	☆ループするた	びに新しいブ	ロックス	コープ内で変数	が宣言	され、初期	化されているからcc	nstで			
	変数を定義	しても重複エ	ラーにな	よらない							
	★ローカルスコ	1 ープ									
			_プを全	カサてローカル	スコープ	と呼ぶ					
-	・関数スコープとブロックスコープを合わせてローカルスコープと呼ぶ ・ローカルスコープの外側をグローバルスコープと呼ぶ										
		ノッハーはごでン	, ц. 77,	ックコ ノビザか	•						

■ 秋日 巻 管理No.	:他記球帳 FND27	Day9	Page/		\neg						
実施内容	オブジェクト入門	•	Page4		- 						
場所	37) 37)	<u> </u>			- 						
実施日	2024/5/10	_ <u>i</u> 実施時間	9:00	~ 12:00	-						
実施者	DJ さん	受講者	9.00	~ 12.00 T		1					
大旭日	Ai さん	文冊日		 		-}					
 作成者	本田		作成日(開始)	2024/5/14	作成日(完了)	2024/5/14					
<time></time>		<u> </u>		2024/3/14	TFJXLI(TLJ)	2024/3/14					
\Time/		<contents></contents>									
	★グローバルスコープ ・ローカルフコープで写言まれた恋粉は、グローバルフコープからアクセフできない。										
		・ローカルスコープで宣言された変数は、グローバルスコープからアクセスできない・グローバルスコープで宣言された変数は、コードの内のどこからでもアクセス出来る									
	70 7007	・グローバルスコープで宣言された复数は、コートの内のとこからでもアクセス出来る const greeting = "Hello"; グローバルスコープ									
		function greet(person) {									
			greeting + " " + p		ローカルスコーフ	0					
		}									
		console.log(greet("Urara"));									
		// ⇒ "Hello Urara"									
		// -> Fiello Orara									
	★コンピュー	タの挙動									
			からない場合、一致す	する名前の変数が	・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・						
					770773 00. 01						
		内側のスコープから外側のスコープへと 順に探す ☆変数の値を外側に探しに行く									
			eeting = "Hello";			ブローバルスコーフ					
			et person) {							
		return greeting + " " + person; ローカルスコープ									
		}									
		console.log(greet("Urara"));									
		// ⇒ "Hello Urara"									
		77									
	★スコープ中	★スコープ中のスコープ(スコープがネストされている場合)									
	・外部スコープの変数は、内部スコープからアクセス可能										
	・内部スコー	・内部スコープ内の変数は、外部スコープからアクセスできない									
		function sum(arrOfNums) {									
	let total = 0;										
	for (const number of arrOfNums) { total += number ;										
		}									
	console.log(number);										
			urn total;								
)	-									