

## ■教育実施記録帳

管理No.	FND27	Day11	Page1
実施内容	while ループと for ループ		
場所	5F	設備・工室	
実施日	2024/5/16	実施時間	9:00 ~ 12:00
実施者	Masaさん	受講者	
	DJさん		
作成者	本田	作成日(開始)	2024/5/16
		作成日(完了)	2024/5/16
<Time>	<Contents>		
	★2つのfor…ループ		
	「for…ofループ」		
	・配列の各要素ごとに反復処理を行う		
	「for…inループ」		
	・オブジェクトの各キーごとに反復処理を行う		
	★for…ループは何のためにあるか		
	・「for…ループ」は「繰り返し同じ処理を行う」		
	・「for…ループ」は「配列の要素やオブジェクトキー1つずつに同じ処理を繰り返す」		
	★while…ループ（回数指定での反復処理）		
	・配列やオブジェクトが無くても反復処理を行える		
	[. js]		
	let counter = 1; //変数counterを初期化(初期値: 1)		
	while ( counter <= 10) { //counterの値が10以下である限り		
	console. log (counter); //counterの値をコンソールに表示		
	counter++; //counterの値を+1		
	} ↑ whileの条件が「いつかfalseになる」ように書く //無限ループ回避のため重要		
	[DevTool Console]		
	1～10を1つずつ表示		
	☆10が表示された時のcounterの値は「11」 //whileが「false」になり反復処理を抜ける値		
	★ループは「i」で回す		
	・由来はiteration（反復）やindex（添え字）		リーダブルコード
	・ただ昔からの習慣というものもあるらしい		クリーンコード
	★while…ループをfor…ループへ		
	[. js] while		
	let i = 1; //初期化		
	while ( i <= 10) { //条件		
	console. log (i); //反復処理内容		
	i++; //条件がいずれ「false」になる処理		
	}		
	[. js] for		
	for ( let i = 1; i <= 10; i++) {		
	console. log (i);		
	}		
	[構文]		
	for ( 初期化 ; 条件 ; 条件がいずれ「false」になる処理 ) {		
	反復処理		
	}		
	☆回数指定での反復処理は「while…ループ」でも「for…ループ」どちらでも書ける		
	★では何故while…ループをfor…ループへするのか？		
	・配列やオブジェクトで回数指定での反復処理が行えるから		

## ■教育実施記録帳

管理No.	FND27	Day11	Page2
実施内容	while ループと for ループ		
場所	ライオン	設備・工室	
実施日	2024/5/16	実施時間	9:00 ~ 12:00
実施者	Masaさん	受講者	
	DJさん		
作成者	本田	作成日(開始)	2024/5/16
		作成日(完了)	2024/5/16
<Time>	<Contents>		
	★配列での回数指定の反復処理		
	・「 i 」を配列のインデックスとすれば、配列の全要素に順番にアクセスできる		
	[. js]		
	const cars = ["Toyota", "Volvo", "BMW", "Honda"];		
	インデックス ⇒ [ 0 , 1 , 2 , 3 ];		
	for ( let i = 0; i < 4; i++) {		
	console.log(cars[i]);	i の値は「for..ループ」の反復処理で決まる	
	}	[ i ] は「carsのインデックス」	
	[DevTool Console]		
	Toyota		
	Volvo		
	BMW		
	Honda		
	[. js] 条件の回数をcars.lengthへ		
	const cars = ["Toyota", "Volvo", "BMW", "Honda"];		
	要素 ⇒ [ 1 , 2 , 3 , 4 ];		
	インデックス ⇒ [ 0 , 1 , 2 , 3 ];		
	for ( let i = 0; i < cars.length; i++) {		
	console.log(cars[i]);	「cars.length」は配列の要素数を返す	
	}	[ i ] は「carsのインデックス」	
	} 混同注意		
	[DevTool Console]		
	Toyota		
	Volvo		
	BMW		
	Honda		
	[. js] 条件の比較演算子を小なり＝へ		
	const cars = ["Toyota", "Volvo", "BMW", "Honda"];		
	要素 ⇒ [ 1 , 2 , 3 , 4 , 5 ];		
	インデックス ⇒ [ 0 , 1 , 2 , 3 , 4 ];		
	for ( let i = 0; i <= cars.length; i++) {		
	console.log(cars[i]);	「<=」は4以上になるため4でfalseにならず5まで行く	
	}	[ i ] は「carsのインデックス」	
	[DevTool Console]		
	Toyota		
	Volvo		
	BMW		
	Honda		
	undefined		
	[. js] cars.length-1とすることで対応可能		
	const cars = ["Toyota", "Volvo", "BMW", "Honda"];		
	要素 ⇒ [ 1 , 2 , 3 , 4 , 5 ];		
	インデックス ⇒ [ 0 , 1 , 2 , 3 , 4 ];		
	for ( let i = 0; i <= cars.length -1; i++) {		
	console.log(cars[i]);	「<= cars.length -1」は「要素数5-1」	
	}	[ i ] は「carsのインデックス」	