DOMKNIĘCIA

Mechanizm korzystania ze ZMIENNEJ POZA jej ZASIĘGIEM leksykalnym(=używa lokalizacji zdefiniowania zmiennej w kodzie źródłowym aby określić gdzie ta zmienna jest dostępna. Zagnieżdżone funkcje mają dostęp do zmiennych zdeklarowanych w ich zewnętrznym zasięgu(przykład z królem))). (=zapamiętuje i korzysta; np. zmienna jest w funkcji, po wykonaniu funkcji kiedy teoretycznie już nie istnieje chce dalej z niej korzystać bo funkcja zapamiętuje i uzyskuje dostęp do zasięgu leksykalnego nawet po wywołaniu tej f na zewnątrz jej zasięgu; nie tylko zapamiętuje ale też wykonuje operacje np. inkrementację);

Dostęp do zmiennej utworzonej we zakresu INNEJ funkcji, która zakończyła swoje działania;

1. RETURN

function inside(){

let insideVar = "insideVar";

return insideVar;

}

let outsideVar = inside();// korzystam ze zmiennej spoza jej zasięgu tj. f inside

console.log(outsideVar)

1. JAK LOKALNIE BRAK PRZYPISANEJ WARTOŚCI TO „SZUKA” wyżej w zmiennej globalej

Zasięg leksykalny chociaż nie do końca…..

// zasięg leksykalny

const name = "Iga";

function getUser(){

let age = 13;

return `imię: ${name} wiek: ${age}`;

}

let userData = getUser();

console.log(userData)

// nie znajduje w swoim zasięgu zmiennej więc idzie wyżej do globalnej

3. przechowanie w pamięci zmiennej po wykonaniu funkcji:

dostęp po wykonaniu funkcji

function hello(name){

return function(day){

let age = 13;// nie będzie tej zmiennej w closure bo nie używam jej poniżej w return!!!

//jak by był link do age ${age} to by zapamiętał

return function(day){

console.log(`hej ${name} ${age}, jest ${day}`)

}

}

const greeting = hello("Iga");

console.log(greeting);

console.dir(greeting);//[[Scopes]]: Scopes[3] 0: Closure (hello) {name: "Iga"} => w konsoli

console.log(greeting("wtorek"));//name po wykonaniu f hello cały czas zapamiętany

console.log(greeting("poniedziałek"))// name po wykonaniu f hello cały czas zapamiętany

// funkcja hello się wykonała i js usuwa z pamięci ALE przechowuje wartość zmiennej "name" PO wykonaniu funkcji bo WIE, że

// będzie z niej klorzystał w funkcji anonimowej (z param (day), którą przypisuję do zmiennej greeting później);

Domknięcie jest kombinacją funkcji i leksykalnego środowiska w którym ta funkcja została zdeklarowana. To środowisko zawiera każdą zmienną lokalną która była w zasięgu w momencie kiedy domknięcie zostało stworzone. W tym przypadku, myFunc jest referencją do instancji funkcji displayName stworzonej w momencie działania makeFunc. Instancja displayName zarządza referencją do jej leksykalnego środowiska, w którym istnieje zmienna. Dlatego, kiedy myFunc jest uruchomiona, zmienna pozostaje dostępna do użycia i "Mozilla" może być przekazane do alert.

function makeFunc() {

var name = "Mozilla";

function displayName() {

alert(name);

}

return displayName;

}

var myFunc = makeFunc();

myFunc();

JAKAS DZIWNA AKCJA ZAMAIST LETOW W PWTLACH SETTIME ITD. UZYWAC IIFE (funkcji od razu wykujących się, którą opatulasz () i dodajesz, () -wywowłąnie pod koniec) wtf nie wiem czy to potrzebne jak są lety etf?

CONST, VAR, LET