block/scope

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/block>

Scopes er et konsept som går igjen i de fleste programmeringsspråk. Det er vanligst at de avgrenses med en krøllparentes {}, men de kan også avgrenses med innrykk. Dette oppgavesettet skal hjelpe deg med å forstå problemstillingene og mekanismene som utgjør en block/scope.

**Oppgave 1**

//Oppgave 1

{

const i = 10;

console.log(`i = ${i}`);

}

Hva blir utskriften her? Forklar hvorfor du får det resultatet du får.

I = 10

Fordi const er deklarert innenfor blokken

**Oppgave 2**

**//Oppgave 2**

**const x = 10;**

**{**

**console.log(`i = ${x}`);**

**}**

Hva blir utskriften her? Forklar hvorfor du får det resultatet du får.

**I = 10**

**Cosnt er deklarert utenfor blokken og har global scope så kan derfor bli skrevet ut innenfor en annen blokk**

**Oppgave 3**

//Oppgave 3

{

const y = 10;

}

console.log(`y = ${y}`);

Hva blir utskriften her? Forklar hvorfor du får det resultatet du får.

Error

Variabelen y er deklarert med const innenfor blokken og er ikke tyilgjengelig utenfor blokken derfor får jeg en error (refrence error)

**Oppgave 4**

//Oppgave 4

let z;

{

z = 10;

//utskrift a)

console.log(`z = ${z}`);

}

//utskrift b)

console.log(`z = ${z}`)

Hva blir utskrift a) ? z = 10

Hva blir utskrift b) ? z = 10

Kan du forklare sammenhengen mellom de to variablene og hvorfor utskriften blir som den blir?

Variabelenb z er deklarert med let utenfor blokken og kan derfor ha tilgang innenfor en annen blokk og den kan endres fordi det ikke er en konstant.

**//Oppgave 5**

//Oppgave 5

let a;

fubar();

console.log(`a = ${a}`);

function fubar(){

a =2;

}

Forklar hvordan denne koden vil kjøre. Hvilken rekkefølge vil kodelinjene kjøre, og hva blir utskriften fra console.log ?

A = 2

Funkjsonen fubar blir kalt før konsole.log og den tildeler verdien 2 til variabelen derfor vil console.log som kommer etter fubar() vise verdien a som er satt inne i funkjsonen som er 2

**//Oppgave 6**

**//Oppgave 6**

**let b = 10;**

**add(b, 5);**

**console.log(b);**

**function add(c, d){**

**let result = c + d;**

**console.log(result);**

**}**

Hvor mange utskrifter vil denne koden gi, og hva vil utskriften bli? Forklar hvordan dette foregår, og hvorfor du får det resultatet du får.

15 og 10

Add funkjsonen blir først kalt også logger det ut svaret som er 15, også blir variabelen b logga, men verdien inni variabelen endrer seg ikke så den forholder seg 10

**//Oppgave 7**

//Oppgave 7

console.log(`e = ${e}`);

var e = 2;

Hva blir utskriften her? Hvordan er dette uvanlig atferd, og hva sier dette om forskjellen, på let/const og var?

E = undenifined

Variabelen e blir deklarert med var og deklarasjonen blir flyttet til toppen av sitt scope, og e er allere deklarert når det logger, men den har ikke ennå fått tildelt sin verdi og derfor er den undenifed.