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.

**Oppgave 2**

**//Oppgave 2**

**const x = 10;**

**{**

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

**}**

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

**Oppgave 3**

//Oppgave 3

{

const y = 10;

}

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

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

**Oppgave 4**

//Oppgave 4

let z;

{

z = 10;

//utskrift a)

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

}

//utskrift b)

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

Hva blir utskrift a) ?

Hva blir utskrift b) ?

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

**//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 ?

**//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.

**~~//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?~~