

## OPPGAVER

# Oppgave 7: Block scope

## Oppgave 1

```
{  
  const i = 10;  
  console.log(`i = ${i}`);  
}
```

Utskriften blir: `i = 10`. Variabelen `i` er definert med `const` innenfor blokken, så den er bare tilgjengelig innenfor blokken.

## Oppgave 2

```
const x = 10;  
{  
  console.log(`i = ${x}`);  
}
```

Utskriften blir: `i = 10`. Variabelen `x` er definert med `const` utenfor blokken, så den er tilgjengelig innenfor blokken.

## Oppgave 3

```
{  
  const y = 10;  
}  
console.log(`v = ${v}`);
```

Dette vil føre til en feil. Variabelen `y` er definert med `const` innenfor blokken, og den er ikke tilgjengelig utenfor blokken.

## Oppgave 4

```
let z;
{
  z = 10;
  // utskrift a)
  console.log(`z = ${z}`);
}
// utskrift b)
console.log(`z = ${z}`);
```

Utskrift a) blir: `z = 10`. Utskrift b) blir: `z = 10`. Variabelen `z` er deklarert med `let` utenfor blokken, så den er tilgjengelig både innenfor og utenfor blokken.

## Oppgave 5

```
let a;
fubar();
console.log(`a = ${a}`);
function fubar() {
  a = 2;
}
```

Utskriften blir: `a = 2`. Funksjonen `fubar` blir kalt før `console.log`, og den endrer verdien til variabelen `a`.

## Oppgave 6

```
let b = 10;
add(b, 5);
```

```
console.log(b);  
function add(c, d) {  
  let result = c + d;  
  console.log(result);  
}
```

Denne koden gir to utskrifter:

Output oppgave 6

```
15  
10
```

Funksjonen `add` endrer ikke verdien til `b`, så `console.log(b)` gir 10.

## Oppgave 7

```
console.log(`e = ${e}`);  
var e = 2;
```

Utskriften blir: `e = undefined`. Dette er uvanlig atferd på grunn av "hoisting" i JavaScript. Variabelen `e` blir løftet til toppen av sitt scope, men initialiseres senere, derfor er verdien `undefined` ved første utskrift. Dette er annerledes enn `let` og `const`, som ikke blir initialisert før deklarasjonen, og derfor ikke er tilgjengelige før dette punktet.

[Previous](#)

Oppgavesett 6: Flytkontroll, loops

[Next](#)

Oppgave 8: Funksjoner

Last updated 1 year ago

---

