async | await

Nyckelordet async används före en funktionsdeklaration:

```
// Named functions
async function test () {}

// Anonyma funktioner
htmlElement.addEventListener("click", async () => {});

// Metoder (som ju är funktioner)
const objekt = {
  method () {},           // vanlig, synkron metod
  async a_method () {}    // asynkron metod
};
```

En noggrann genomgång av vad nyckelordet **async** gör när man skriver den före en funktionsdeklaration ligger utanför kursens ramar. Det finns info om det här, och på Youtube... och man kan alltid *fråga* AI.

async | await

En funktion som har deklarerats med nyckelordet async kallas – föga förvånande – en async-funktion. Inom ramen för denna kurs behöver ni veta detta om async-funktioner:

- En async-funktion returnerar ALLTID en instans av Promise.
- Async-funktioner, och endast dem, kan innehålla en await-instruktion.
- En async-funktion exekveras asynkront (mer om det när vi pratar om await).

async | await

```
function f1 () {
  return "WDU";
}

const x = f1();
console.log(x);
```

Vanlig funktion som returnerar strängen "WDU".

Inga konstigheter.

async | await

```
async function f1 () {
  return "WDU";
}

const x = f1();
console.log(x);
```

Async-funktioner returnerar ALLTID en *promise*.
Även om koden verkar säga att den returnerar en sträng.
Så när vi loggar x ser vi att den innehåller en *promise*.

(Notera dock att *promise* redan är uppfylld)