

Alternativa sätt att skriva koden

Kan vara farligt om man inte läser på mer!

Om vi ser till att `handleResponse` returnerar den nya promisen så kan vi skriva så här (vanligt sätt att skriva):

```
const request = new Request("http://localhost:4242/random");

const promiseResponse = fetch(request);
promiseResponse.then(handleResponse).then(handleResource);

function handleResponse (response) {
  const promiseResource = response.text();
  return promiseResource;
}

function handleResource (resource) {
  ... do something with the resource
}
```

Raden `promiseResponse.then(handleResponse).then(handleResource)` är en s.k. *promise-chain*. Det är en något speciell konstruktion som har särskilda regler.

Ni bör INTE använda detta sätt att skriva koden om ni inte är beredda att sätta er in i hur exakt en *promise-chain* fungerar. [Här](#) finns info om dem. Det finns säkert mer info på Youtube, eller ni kan fråga AI.

Alternativa sätt att skriva koden

Alla dessa sätt innehåller en promise chain. Läs på innan du använder dem!

Eller, med radbrytningar:

```
const request = new Request("http://localhost:4242/random");

const promiseResponse = fetch(request);
promiseResponse
  .then(handleResponse)
  .then(handleResource);

function handleResponse (response) {
  const promiseResource = response.text();
  return promiseResource;
}

function handleResource (resource) {
  ... do something with the resource
}
```

Alternativa sätt att skriva koden

Alla dessa sätt innehåller en promise chain. Läs på innan du använder dem!

Med anonyma funktioner:

```
const request = new Request("http://localhost:4242/random");

const promiseResponse = fetch(request);
promiseResponse
  .then(function (response) {
    const promiseResource = response.text();
    return promiseResource;
  })
  .then(function (resource) {
    ... do something with the resource
  })
  );
```

Alternativa sätt att skriva koden

Alla dessa sätt innehåller en promise chain. Läs på innan du använder dem!

Med pil-funktioner

(Kom ihåg: de returnerar automatiskt ett värde om de bara innehåller en instruktion)

```
const request = new Request("http://localhost:4242/random");

const promiseResponse = fetch(request);
promiseResponse
  .then(response => response.text())
  .then(resource => {
    ... do something with the resource
  })
  );
```

Alternativa sätt att skriva koden

Alla dessa sätt innehåller en promise chain. Läs på innan du använder dem!

Om vi skippar konstanten *promiseResponse*

```
const request = new Request("http://localhost:4242/random");

fetch(request)
  .then(response => response.text())
  .then(resource => {
    ... do something with the resource
  })
  );
```

Det går bra att använda vilket alternativ du vill

MEN du måste förstå hur koden fungerar, för annars blir det omöjligt för dig att ändra i den.

Tentan går inte ut på att kontrollera om du kan memorera raderna ovan, den går ut på att kontrollera om du har förstått hela processen!