React Training

> Day Five

Handling Async with React

Asynchronous

Asynchronous

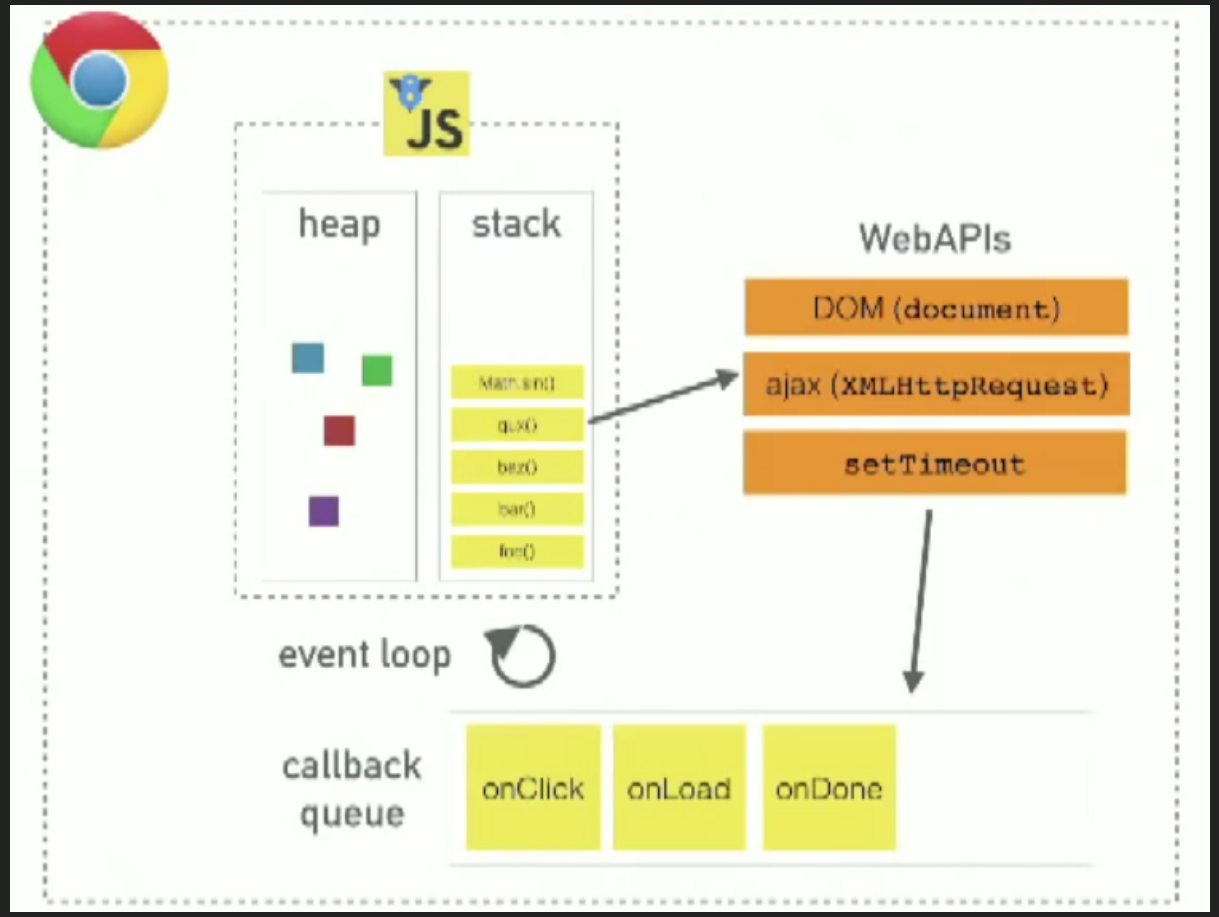
> A task be performed alongside the original task (or tasks), without stopping to wait for the task to complete. When the secondary task is completed, the original task is notified using an agreed-upon mechanism so that it knows the work is done, and that the result, if any, is available.

JavaScript is single threaded

> Even with multiple cores, you could only get it to run tasks on a single thread, called the **main thread**

setTimeout? setInterval?

Event Loop


JS
heap
stack
WebAPIs
DOM (document)
ajax (XMLHttpRequest)
setTimeout
event loop
callback queue
onClick
onLoad
onDone

Promise

Promise

> Assurance that a certain event will happen

Either Keep it or Break it

```
let promise = new Promise((resolve, reject) => {
  if(/* asynchronous code execution is successful */) {
    resolve(/* result */);
  } else {
    reject(/* error */);
  }
});
```

How to deal with fulfilled promise and heartbreaks?

```
promise.then((result) => {
  console.log(result);
},
(error) => {
  console.log(error);
});
// you could handle the errors by passing it in .catch instead of .then as well
promise.catch((error) => { console.log(error); });
```

How to keep multiple promises?

```
Promise.all(/* arrayOfPromises */).then((values) => {
  \\ values is array of resolved promise values from arrayOfPromises
},(error) => {
  \\ if any of the promises in arrayOfPromises fails we reach here
});
```

Async/Await

- ES2017
- Makes unpredictable asynchronous functions work sequentially (synchronously)

```
const someAsyncFn = async (param) => {
  const result = await someOtherAsyncFn(param);
  return result;
}
```

Async/Await

```
const someAsyncFn = async (param) => {
  const result = await someOtherAsyncFn(param);
  return result;
}
const mainFn = () => {
  someAsyncFn();
  console.log('Hello');
};
```

Network Requests are asynchronous

- Callback
- Promises
- Async/Await

More on promises

> https://medium.com/devnetwork/moving-to-promises-and-async-await-from-callback-a003e09fe91c